

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
МЕЖДУНАРОДНАЯ АКАДЕМИЯ ИНФОРМАТИЗАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
НАУЧНО-ТЕХНИЧЕСКОЕ ПРЕДПРИЯТИЕ «КРИПТОСОФТ»
ПЕНЗЕНСКИЙ ФИЛИАЛ ОАО «РОСТЕЛЕКОМ»
ООО «ЦЕНТР АНАЛИЗА И РАЗВИТИЯ КЛАСТЕРНЫХ СИСТЕМ»
ТЕХНОПАРК ВЫСОКИХ ТЕХНОЛОГИЙ «РАМЕЕВ»
ЗАО «НИИФИ и ВТ»

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ

СБОРНИК НАУЧНЫХ СТАТЕЙ
XI Международной научно-технической
конференции

г. Пенза, 25–27 ноября 2014 г.

Proceedings of the Eleventh International Conference
of Science and Technology

NEW INFORMATION TECHNOLOGIES
AND SYSTEMS

Penza, Russia, November 25–27, 2014

Пенза
Издательство ПГУ
2014

Новые информационные технологии и системы : сб. науч. ст.
Н72 XI Междунар. науч.-техн. конф. (г. Пенза, 25–27 ноября 2014 г.). – Пенза :
Изд-во ПГУ, 2014. – 444 с.

ISBN 978-5-94170-916-8

Сборник статей подготовлен на основе докладов, сделанных на XI Международной научно-технической конференции «Новые информационные технологии и системы» («НИТиС-2014»), проведенной Министерством образования и науки РФ, Международной академией информатизации, Пензенским государственным университетом, НТП «Криптософт», Пензенским филиалом ОАО «РОСТЕЛЕКОМ», технопарком высоких технологий «Рамеев», ЗАО «НИИФИ и ВТ», ООО «Центр анализа и развития кластерных систем».

Доклады охватывают широкий спектр проблем в области новых информационных технологий в производстве, управлении, образовании; рассматриваются вопросы построения высокопроизводительных вычислительных комплексов, систем и сетей. Представлены современные технологии хранения и обработки данных, создания аппаратно-программных комплексов и информационно-вычислительных систем, интеллектуальных систем и систем управления. Обсуждаются вопросы моделирования информационно-вычислительных систем и применения математических методов в информатике. В тематику конференции включен раздел, посвященный вопросам электронного документооборота.

УДК 004.4

Редакционная коллегия:

**В. И. Волчихин, В. Б. Механов, Н. П. Вашкевич,
П. П. Макарычев, А. В. Кучин, Н. Н. Коннов**

*Сборник подготовлен и издан в рамках Федеральной целевой программы
«Исследования и разработки по приоритетным направлениям развития
разработки научно-технологического комплекса России на 2014–2020 годы»
(соглашение № 14.574.21.0045 от 06.19.2014)*

ISBN 978-5-94170-916-8

© Пензенский государственный
университет, 2014

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ И ПАРАЛЛЕЛЬНЫЕ СИСТЕМЫ

Н. П. Вашкевич, Е. И. Калиниченко

Россия, Пенза, Пензенский государственный университет

ФОРМАЛЬНЫЙ ЯЗЫК НА ОСНОВЕ КОНЦЕПЦИИ НЕДЕТЕРМИНИЗМА ДЛЯ ОПИСАНИЯ АЛГОРИТМОВ УПРАВЛЕНИЯ ПРОЦЕССАМИ И РЕСУРСАМИ В ПАРАЛЛЕЛЬНЫХ СИСТЕМАХ ОБРАБОТКИ ИНФОРМАЦИИ

Рассматривается математический аппарат для аналитического описания алгоритмов логического управления процессами и ресурсами, основанный на использовании модели недетерминированного автомата (НДА). Показана связь предлагаемого способа представления управляющих алгоритмов (УА) с описанием их на других широко известных языках, в том числе на языках: CTL, регулярных выражений алгебры событий, VHDL и др.

При разработке и создании систем логического управления (СЛУ) информационными процессами и ресурсами при реализации вычислительных задач или задач промышленной автоматики и управления объектами формальным методом описания управляющих алгоритмов (УА) придается большое значение. Это связано с тем, что такие формальные методы должны обеспечивать комплексное решение задач спецификации, разработки, верификации, реализации и анализа сложных систем управления, в том числе систем управления взаимодействующими параллельными процессами и ресурсами в вычислительных системах и сетях. Математический аппарат формальных методов описания УА должен обладать при этом высокой эффективностью, позволяющей решать, в том числе задачи синхронизации процессов и задачи, связанные с повышением производительности и надежности систем управления.

Под высокой эффективностью математического аппарата формального языка описания УА понимается выполнение следующих основных требований:

- простота, компактность и удобство использования для разработчика и заказчика;
- высокая степень выразительности при описании сложных вариантов алгоритмов, в том числе с отражением временного фактора, а также включающих как синхронные, так и асинхронные процессы с отсутствием тупиковых ситуаций, когда имеется зависимость реализуемых событий по данным и ресурсам;
- наличие методов оптимизационных равносильных преобразований, связанных с минимизацией, композицией, декомпозицией и верификацией УА на моделях и возможностью простой трансформации описания с одного языка на другой;
- простой переход от формального аналитического представления УА к его структуре в аппаратной, программной или аппаратно-программной реализации;
- математический аппарат должен быть ориентирован на описание параллельно-последовательных алгоритмов;
- должно быть обеспечено описание широкого круга применения по описанию управления процессами и объектами, начиная с торговых автоматов и кончая операционными системами с разделяемыми ресурсами.

Для описания систем параллельного действия авторы отдают предпочтение сетям Петри. Авторы предлагают для описания таких систем использовать модель НДА, учитывая ее достоинство, заключающее в явном параллелизме, который проявляется в том, что под действием одного и того же входного сигнала допускается переход от одного частного события к нескольким событиям, одновременное существование которых в соответствии с УА возможно. Т.е. описание переходов в УА при использовании моделей НДА выполняется не в терминах состояний ДА, а в терминах частных событий НДА, что и обеспечивает такую же простоту, компактность и удобство описания УА в параллельных системах, которые характерны для описания алгоритмов систем последовательного действия при использовании моделей ДА.

Формальный язык представления УА на основе логики НДА можно рассматривать и как вариант базового языка, который включает две составляющие: графическую форму для наглядного представления алгоритмов в виде граф-схем с параллельными ветвями (ГСАП) и аналитическую в виде системы рекуррентных канонических уравнений (СКУ), описывающих все реализуемые в алгоритме частные события. При этом под частными событиями, в зависимости от уровня детализации УА, понимается: для вычислительных алгоритмов – выполнение некоторых актов обработки информации, к которым могут относиться: выполнение микроопераций, макроопераций, функциональных операторов, подпрограмм и т.д., а для управляющих алгоритмов промышленной автоматики – выполнение отдельных технологических операций или групп операций. В дальнейшем аналитическую форму представления управляющего алгоритма в виде СКУ будем для краткости называть языком НД СКУ, который является аналитической интерпретацией языка ГСАП.

Предлагаются две канонические формы языка НД СКУ. Первая – простая каноническая форма языка НД СКУ. Для формального аналитического представления не сложных УА может быть использована простейшая модель конечного цифрового НДА Мура в виде стандартной системы рекуррентных бескванторных предикатных формул следующего вида, описывающих все реализуемые в алгоритме частные события:

$$S_j^{Ij}(t+1) = \bigvee_{i,j} X_{i,j}(t) \& S_i(t) \vee \bigvee_j X_{j,j}(t) \& S_j(t),$$

$$j = \overline{0, m}, S_0(0) = 1, \quad (1)$$

где $X_{i,j}$ – частный входной сигнал, представляющий собой сочетание (конъюнкцию) элементарных двоичных входных сигналов из структурного алфавита $[X]$, взятых с отрицанием или без отрицания, под действием которых осуществляется переход от события S_i к событию S_j ; $S_i(t)$ – сокращенное обозначение выражения, описывающего частное событие S_i , непосредственно предшествующее частному событию $S_j(t)$; каждая предикатная формула $S_i(t)$ должна быть либо тождественно истинной предикатной формулой, либо удовлетворять условию $S_i(t) \neq S_j(t)$ и определяться также выражением вида (1); Y_j – частный выходной сигнал, представляющий собой сочетание элементарных двоичных выходных сигналов из структурного алфавита Y , отмечающих событие S_j .

Вторая – обобщенная каноническая форма языка НД СКУ. Она позволяет расширить выразительные возможности языка. Расширение выразительных возможностей формального языка НД СКУ необходимо для учета связи событий не только по управлению и информации, но и по данным и ресурсам, а также для описания поведения управляющих систем с учетом временных факторов для асинхронного режима работы. Реализация таких возможностей языка НД СКУ достига-

ется за счет варьирования описаний условий зарождения и сохранения события S_j , которые могут зависеть не только от значений внешних частных входных сигналов $X_{i,j}$, но и от значений совокупности некоторых частных событий, имеющих место в моменты времени, предшествующие событию S_j . В связи с этим математическую модель НДА Мура можно представить в виде следующей обобщенной канонической формы:

$$S_j^{I,j}(t+1) = \bigvee_{i,j} S_{j,z} S_i \vee \bigvee_j S_{j,c} S_j, \quad (2)$$

где $S_{j,z}$ и $S_{j,c}$ – комбинационные события, определяющие условия зарождения и сохранения события S_j соответственно.

Следует обратить внимание на то, что при проверке систем описанных на НД СКУ, после формализации УА, в них отсутствуют недостижимые события, т.е. событий которые не имеют событий предшественников. Чтобы убедиться в этом, можно по исходной СКУ построить прямую таблицу переходов, которая строится таким образом, чтобы очередное событие можно было бы занести в таблицу в качестве исходного только в том случае, если оно имело место в одном из предшествующих шагов построения такой таблицы.

Отражение временного фактора в управляющих алгоритмах, определяющих поведение системы управления во времени. Один из возможных вариантов отражения асинхронности режима работы системы логического (СЛУ) управления может быть реализован за счет расширения входного алфавита $[X]$ путем ввода двоичных входных сигналов, которые сигнализируют о готовности входной информации (исходных данных) для ее обработки и готовности блоков системы для приема результатов обработки, а также сигналов о завершении обработки исходных данных.

Связь языка НД СКУ с другими языками обработки информации и с возможностью трансформации описания с одного языка на другой. Предлагаемая обобщенная каноническая форма представления УА в виде НД СКУ (2) позволяет естественно и просто переводить описание алгоритма на языки программирования, например, на язык VHDL, когда требуется выполнить моделирование алгоритма для его верификации или аппаратную реализацию, например, на ПЛИС. Действительно, при представлении исходного УА в виде программы на языке VHDL, правые части всех уравнений исходной СКУ НДА преобразуются в коды, которые имеют такой же вид, что и исходные уравнения, но только с другими обозначениями переменных и операций над ними, которые указываются в начале программы.

Например, для события типа S_k^i входа процесса в критический интервал имеем:

$$S_k^i(t+1) = S_{\text{ВП}}^i S_{\text{ВЗ}}^i S_{\text{ПР}}^i \vee S_k^i S_p^i :$$

$$sk \leq (svp \text{ and } svz \text{ and } spr) \text{ or } (sk \text{ and } (\text{not } sp)).$$

Закключение. Рассмотренный в статье формальный язык НД СКУ и его выразительные возможности позволяют формально описывать сложные алгоритмы ЛУ в том числе с параллельными взаимодействующими процессами в простой и наглядной форме как аналитически, так и графически. В статье показана также эффективность способов учета асинхронности и отражения временных факторов, в том числе для времени, заданного не в явном виде при представлении алгоритмов ЛУ, а также простота трансформации описания алгоритмов на известных начальных языках на язык НД СКУ и с этого языка на языки программирования.

Н. П. Вашкевич, Е. И. Калиниченко

Россия, Пенза, Пензенский государственный университет

ПРОЕКТИРОВАНИЕ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ В ЗАДАЧАХ ИДЕНТИФИКАЦИИ

Рассматриваются вопросы решения задач распознавания цепочек – образов с использованием теорий регулярных выражений алгебры событий и недетерминированных автоматов. Приводятся примеры решения таких задач. Предлагается методика проектирования параллельных алгоритмов. Описывается инструментальная система, позволяющая автоматизировать разработку алгоритмов с последующей их верификацией.

В типичной задаче идентификации цепочек-образов – задаются входная последовательность W (например, символов или пронумерованные фрагменты графических файлов и т.п.) и множество цепочек-образов $\{z_1, z_2, \dots, z_n\}$. Требуется найти либо хотя бы одно вхождение какой-то цепочки-образа, либо m из n ($1 \leq m < n$), либо вхождения всех n цепочек-образов в W . При этом множество цепочек-образов часто является регулярным множеством, которое можно задать регулярным выражением алгебры событий.

Необходимо отметить, что задачи распознавания образов не сводятся только к поиску одной единственной цепочки во входной последовательности. Приведем формулировку таких задач, которые встречаются на практике.

Поиск любой из нескольких цепочек, т.е. хотя бы одна из цепочек есть во входной последовательности. Вообще казалось бы эту задачу можно свести к очередному поиску каждой цепочки за несколько просмотров входной последовательности, но в этом случае время поиска возрастает пропорционально количеству цепочек и их длине, а в некоторых случаях, например «поиск на лету» (без возможности повторных просмотров входной последовательности) он в принципе становится невозможным.

Поиск наличия всех заданных цепочек, т.е. во входной последовательности должны быть все из искомых цепочек до того как на вход автомата поступит «bottom».

Поиск заданного количества из всех искомых цепочек, т.е. например ищутся 5 цепочек, а результат поиска может быть положительным, если будут найдены хотя бы 3 из них. Здесь дополнительная трудность при поиске состоит в том, что заранее невозможно определить порядок, в котором они будут находиться во входной последовательности. Если же искать их последовательно друг за другом то из-за этого возможны существенные потери времени.

Поиск нескольких цепочек, у которых префикс и суффикс одинаковы, а корни различны. Если префикс обозначить через A , суффикс через C , а корни через B_1, B_2, B_3 , то на языке РВАС алгоритм поиска запишется как: $r(y) = A \& (B_1 \vee B_2 \vee B_3) \& C$. Можно, конечно, выполнить преобразование, раскрыв скобки, но тогда суффиксы и префиксы уйдут в каждую из трех цепочек, что увеличит размерность алгоритма.

Поиск цепочек, когда во входной последовательности некоторые символы являются «прозрачными» (т.е. их как бы нет). Например, игнорирование знаков препинания в тексте или определенных байтовых комбинаций в двоичных файлах.

Когда поиск во входной последовательности начинается по сигналу «извне», а до того времени на каждом шаге осуществляется просто прием очередного символа входной последовательности. Например, когда поиском управляет внешняя среда или когда нужно найти цепочку после пропуска определенного количества символов входной последовательности.

Такие задачи эффективно решаются с использованием модели недетерминированного конечного автомата (НДКА).

Отметим, что начальное состояние s_0 НДКА, которое назовем **маркером начала поиска** имеет очень существенное значение. Маркер позволяет динамически управлять процессом поиска без изменения самого алгоритма поиска (программная или аппаратная реализация не меняется), что обеспечивает гибкость при реализации алгоритма. Например, не меняя алгоритма, а управляя только маркером можно получить решение следующих подзадач, которые могут являться составной частью, перечисленных выше, задач. А именно:

1) не начинается ли входная последовательность с искомым цепочек, т.е. самый первый символ последовательности совпадает с первым символом цепочки. Это может потребоваться если производится такой поиск во многих файлах или даже в том случае если это поиск в одном файле (который в принципе можно сделать вручную), но цепочка достаточно длинная (сотни и более байт) и тогда возможна ошибка из-за человеческого фактора. Такая задача решается установкой маркера начала поиска ($s_0 = 1$) перед приемом входной последовательности и сбросом его ($s_0 = 0$) после приема первого символа.

2) классическая задача поиска, когда входная последовательность может содержать цепочки в любом месте между «top» и «bottom». Эта задача решается установкой маркера ($s_0 = 1$) перед приемом входной последовательности, без последующего сброса (поиск искомым цепочек начинается с каждого очередного символа).

3) поиск во входной последовательности начинается после поступления в ней определенного символа или символов (точки, отмечающей конец предложения и т.п.). Эта задача решается установкой маркера после поступления этого символа, без последующего сброса.

Решение задачи типа «1». Алгоритм решения этой задачи (без учета поступления символа «bottom») можно непосредственно записать на языке РВАС. Например необходимо определить есть ли во входной последовательности любая из трех следующих цепочек:

$$1) z_3 z_2 z_1 \dots z_1 z_3; \quad 2) z_2 \dots z_2 z_1 z_3; \quad 3) z_4 \dots z_4 z_2 \dots z_2 z_3 z_1 \dots z_1,$$

где $z_i \dots z_i$ повторение символа z_i любое число раз, но не менее одного. При обнаружении любой цепочки должен быть выработан выходной сигнал y .

Тогда на языке РВАС алгоритм поиска запишется:

$$r(y) = s_0(z_3 z_2 z_1 \{z_1\} z_3 \vee z_2 \{z_2\} z_1 z_3 \vee z_4 \{z_4\} z_2 \{z_2\} z_3 z_1 \{z_1\}).$$

Для перехода от РВАС к СКУ и СВФ в соответствии с алгоритмом приведенным в 1.1, преобразуем выражение r в следующую систему из трех уравнений, где s_{k1} , s_{k2} , s_{k3} события, появляющиеся при обнаружении цепочек 1, 2, 3 соответственно:

$$s_{k1}(y) = s_0 z_3 z_2 z_1 \{z_1\} z_3$$

$$s_{k2}(y) = s_0 z_2 \{z_2\} z_1 z_3$$

$$s_{k3}(y) = s_0 z_4 \{z_4\} z_2 \{z_2\} z_3 z_1 \{z_1\}.$$

Дальнейшие преобразования дадут следующие СКУ и СВФ:

СКУ:

$$\begin{aligned}s_1 &= s_2 \& z_1 \vee s_1 \& z_1; & s_2 &= s_3 \& z_2; & s_3 &= s_0 \& z_3; & s_{k1} &= s_1 \& z_3; \\ s_4 &= s_5 \& z_1; & s_5 &= s_0 \& z_2 \vee s_5 \& z_2; & s_{k2} &= s_4 \& z_3; \\ s_6 &= s_7 \& z_3; & s_7 &= s_8 \& z_2 \vee s_7 \& z_2; & s_8 &= s_0 \& z_4 \vee s_8 \& z_4; \\ s_{k3} &= s_6 \& z_1 \vee s_{k3} \& z_1;\end{aligned}$$

СВФ:

$$y = s_{k1} \vee s_{k2} \vee s_{k3}$$

Решение задачи типа «2». Получить алгоритм решения этой задачи возможно следующим образом:

- на языке РВАС записывается алгоритм поиска любой цепочки, как и в предыдущей задаче;
- по РВАС строится СКУ и СВФ;
- СКУ изменяется (корректируется) так, что событие, возникающее при обнаружении какой-либо цепочки устанавливается и остается при каждом следующем поступлении входного символа (тактовом сигнале).
- поиск завершается успешно, если конъюнкция всех таких событий становится равной 1 (все цепочки найдены) и неуспешно если до этого поступит символ «bottom».

Используем тот же пример что и в задаче 1, только теперь все искомые цепочки должны присутствовать во входной последовательности.

СКУ:

$$\begin{aligned}s_0 &= 1; & s_1 &= s_2 \& z_1 \vee s_1 \& z_1; & s_2 &= s_3 \& z_2; & s_3 &= s_0 \& z_3; & s_{k1} &= s_1 \& z_3 \vee \underline{s_{k1}}; \\ s_4 &= s_5 \& z_1; & s_5 &= s_0 \& z_2 \vee s_5 \& z_2; & s_{k2} &= s_4 \& z_3 \vee \underline{s_{k2}}; \\ s_6 &= s_7 \& z_3; & s_7 &= s_8 \& z_2 \vee s_7 \& z_2; & s_8 &= s_0 \& z_4 \vee s_8 \& z_4; \\ s_{k3} &= s_6 \& z_1 \vee s_{k3} \& z_1 \vee \underline{s_{k3}}; \\ s_b &= s_0 \& \text{bottom} \vee s_b;\end{aligned}$$

СВФ:

$$y = s_{k1} \& s_{k2} \& s_{k3}; \quad y_b = s_b.$$

Решение задачи типа «3». При разработке алгоритма решения этой задачи наряду с замечаниями к задаче «2» нужно еще учесть, что нельзя заранее знать порядок, в котором появятся цепочки во входной последовательности. Используем тот же пример что и в задаче 1, только теперь необходимо найти хотя бы две цепочки из трех.

При переходе от РВАС к СКУ и СВФ проделаем тоже, что и в задаче 2, но по иному выполним коррекцию СВФ.

СКУ:

$$\begin{aligned}s_0 &= 1; \\ s_1 &= s_2 \& z_1 \vee s_1 \& z_1; & s_2 &= s_3 \& z_2; & s_3 &= s_0 \& z_3; & s_{k1} &= s_1 \& z_3 \vee s_{k1}; \\ s_4 &= s_5 \& z_1; & s_5 &= s_0 \& z_2 \vee s_5 \& z_2; & s_{k2} &= s_4 \& z_3 \vee s_{k2}; \\ s_6 &= s_7 \& z_3; & s_7 &= s_8 \& z_2 \vee s_7 \& z_2; & s_8 &= s_0 \& z_4 \vee s_8 \& z_4; \\ s_{k3} &= s_6 \& z_1 \vee s_{k3} \& z_1 \vee s_{k3}; \\ s_b &= s_0 \& \text{bottom} \vee s_b;\end{aligned}$$

СВФ:

$$y = \underline{s_{k1} \& s_{k2} \vee s_{k1} \& s_{k3} \vee s_{k2} \& s_{k3}};$$

$$y_b = s_b.$$

Следует отметить, что предложенные алгоритмы решения задач поиска разработаны для однопроцессорной системы (последовательная реализация алгоритма), но внутри они уже являются параллельными.

С. В. Шалагин

Россия, Казань, Казанский национальный исследовательский технический университет им. А. Н. Туполева

ОЦЕНКА СЛОЖНОСТИ РАСПРЕДЕЛЕННОГО ВЫЧИСЛЕНИЯ НЕЛИНЕЙНОЙ ПОЛИНОМИАЛЬНОЙ ФУНКЦИИ НАД ПОЛЕМ $GF(2^k)$ НА МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ

Задача оценки сложности распределенного вычисления заданной нелинейной полиномиальной функции от m переменных над полем Галуа вида $GF(2^k)$ на многопроцессорной вычислительной системе сведена к задаче анализа ярусно-параллельного графа.

При реализации широкого класса систем обработки цифровой информации в реальном масштабе времени применимы распределенные вычисления: параллельное и/или конвейерное выполнения одних и тех же операций над потоком данных, которые реализуемы на многопроцессорных вычислительных системах (МВС) [1]. Вычислительными узлами МВС являются микропроцессоры (МП) как общего, так и специального назначения, реализуемые на СБИС, в частности, на программируемых логических интегральных схемах [2]. В РФ производятся МВС с программируемой архитектурой, элементами которых служат ПЛИС класса FPGA (например, «Рысь», «Медведь», «Орион» и др.) [3, 4], применение которых возможно в качестве бортовых систем и/или встроенных систем, реализующих различные устройства с высокой степенью надежности в разные периоды времени. В данном случае в качестве МП могут выступать аппаратные IP-ядра внутри ПЛИС/FPGA: генераторы булевых функций от заданного количества переменных внутри конфигурируемых логических блоков [2, 5, 6].

Задача реализации на МВС с программируемой архитектурой широкого класса устройств для генерирования и обработки цифровых сигналов сводится к задаче синтеза однотипных вычислительных устройств, реализующих нелинейную полиномиальную функцию от m переменных над полем Галуа вида $GF(2^k)$ (НПФ(m)) [7]. В [8] показано, что НПФ(m), определенная над $GF(2^2)$, синтезируема на ПЛИС семейства Virtex-4 на основе однотипных IP-ядер, имеющих высокую степень соответствия архитектуре ПЛИС указанного семейства. Высокая степень соответствия достигается за счет возможности конфигурирования внутри ПЛИС/Virtex-4 генераторов произвольных булевых функций от четырех переменных – ГФ(4). Два ГФ(4) позволяют реализовать либо операцию умножения двух элементов, либо операцию поразрядной суммы по модулю два четырех элементов $GF(2^2)$. В общем случае, однотипные IP-ядра (программные или схемотехнические), реализующие операции над $GF(2^k)$, имеют высокую степень соответствия архитектуре МВС, включающей однотипные элементы (аппаратные IP-ядра), реализующие произвольную булеву функцию от $2 \cdot k$ переменных – ГФ($2 \cdot k$). Получены оценки временной и аппаратной сложности распределенного вычисления НПФ(m), определенной над $GF(2^k)$ вида:

$$f(x_1, \dots, x_m) = \sum_{i_1=0}^w \dots \sum_{i_m=0}^w a_{i_1 \dots i_m} x_1^{i_1} \dots x_m^{i_m}, \quad (1)$$

где $a_{i_1 \dots i_m}, x_1^{i_1}, \dots, x_m^{i_m} \in GF(2^k)$ – элементарный полином (ЭП), $i_j = \overline{0, 2^k - 1}$, $j = \overline{1, m}$, $w = 2^k - 1$, символом $\sum$ обозначена операция вычисления поразрядной суммы по

модулю два. Выражения $a_{i_1 \dots i_m} x_1^{i_1} \dots x_m^{i_m}$ в (1) определены как элементарные полиномы (ЭП). Наличие коэффициентов $a_{i_1 \dots i_m} = 0$ позволяет не производить вычисление значений соответствующих ЭП. Рассмотрим процесс вычисления НПФ(1) вида (1) на основе ярусно-параллельного графа (ЯПГ), описывающего заданный процесс вычисления и описываемого определенными числовыми характеристиками [9]. Пусть МВС указанного класса включают в свой состав МП, реализующие $\Gamma\Phi(2 \cdot k)$, а также параллельные регистры на k разрядов – $RG(k)$, $k = 2, 3, \dots$. Обозначим $t_{\hat{A}\hat{O}(2k)}$, $t_{RG(k)}$ и t_{IC} времена задержки функционирования $\Gamma\Phi(2 \cdot k)$, $RG(k)$ и межсоединений (МС) внутри МВС, соответственно, а t_{IOB}^{out} и t_{IOB}^{in} – времена задержек внутри модулей МВС, запрограммированных (или сконфигурированных) как выходы и входы, соответственно. Имеют место.

Определение 1. Процесс вычисления НПФ вида (1) описывается ЯПГ G с вершинами $v_{01}, v_{02}, \dots, v_{0w}, v_{10}, v_{11}, v_2$ и v_D , которым соответствуют следующие операции над элементами $GF(2^k) - c \cdot x, x^2, \dots, x^w, x_1 \cdot x_2, c \cdot x_1 \cdot x_2$, сумма по модулю два от $2 \cdot k$ операндов и сохранение элемента $GF(2^k)$ в параллельном регистре, соответственно, $w = 2^k - 1$.

Множество вершин ЯПГ G $v_{01}, v_{02}, \dots, v_{0w}, v_{10}, v_{11}$ и v_2 обозначим как V . Дугам G соответствуют МС МВС, требуемые для вычисления НПФ вида (1). Для выполнения операций, которым соответствуют вершины G из V , требуется по k $\Gamma\Phi(2 \cdot k)$ на каждую операцию, а которым соответствуют $v_D - RG(k)$. Справедливо

Определение 2. При конвейерном вычислении НПФ вида (1) с сохранением промежуточных результатов операций, описываемых на каждом ярусе G , верхняя оценка времени задержки вычислений на одной ступени конвейера составляет:

$$t_{\hat{A}\hat{O}(2k)} + t_{RG(k)} + T, \quad (2)$$

где $T = \max(t_{IC}, t_{IOB}^{out}, t_{IOB}^{in})$.

Вычисление ЭП $a_{i_1 \dots i_m} x_1^{i_1} \dots x_m^{i_m}$ отображено при использовании множества подграфов $G - \text{ЯПГ } G(m, P)$, где множество $P = \{p_0, p_1, \dots, p_w\}$, $p_t \geq 0$ – количество x_j^t , $t = \overline{0, w}$, $j = \overline{1, m}$, $\sum_{t=0}^w p_t = m$, $w = 2^k - 1$. Имеет место.

Определение 3. Количество ярусов подграфа $G(m, P)$, при условии, что $p_0 \in [0, (m-1)]$, равно $h(G(m, P)) = 1 + \lceil \log_2(\lceil p_1 / 2 \rceil + p_2 + \dots + p_w) \rceil$, причем $h = \max(h(G(m, P))) = 1 + \lceil \log_2 m \rceil$. Если для $G(m, P)$ $p_1 = 0$, то множество вершин первого яруса $G(m, P)$ – пустое.

Определение 4. Процесс вычисления операций, соответствующих v_2 , описывается графом G_2 , который является подграфом ЯПГ G . Справедливы

Утверждение 1. Количество вершин $G(m, P)$ на l -м ярусе – $N_l(G(m, P)) = N_l^{(V)}(G(m, P)) + N_l^{(D)}(G(m, P))$, где $N_l^{(V)}(G(m, P))$ – количество вершин из V , $N_l^{(D)}(G(m, P))$ – количество вершин вида v_D , $p_0 \in [0, (m-1)]$.

Утверждение 2. Количество вершин G_2 на l -м ярусе – $N_l(G_2) = N_l^{(v)}(G_2) + N_l^{(D)}(G_2)$, где $N_l^{(v)}(G_2)$ и $N_l^{(D)}(G_2)$ – количество вершин вида v_2 и v_D , соответственно, причем и $N_0(G_2) = N_1(G_2) = 0$.

Обозначим q_l – количество ЭП НПФ вида (1), вычисленных на $(l-1)$ -м ярусе, $l = \overline{2, h}$, где $h \leq L$, h введено в определении 5, L – номер максимального яруса ЯПГ G . На основе утверждения 2 предложен алгоритм вычисления значений $N_l^{(v)}(G_2)$ и $N_l^{(D)}(G_2)$, включающий два этапа. На этапе инициализации: $X := 0$; $l := 2$. Этап 1 для $l = \overline{2, h}$: $X := X + q_l$, если $X > 2 \cdot k$, то $\{N_l^{(v)}(G_2) = \lfloor X/(2 \cdot k) \rfloor$, $N_l^{(D)}(G_2) = \lfloor X/(2 \cdot k) \rfloor + X \bmod(2 \cdot k)$ и $X := \lfloor X/(2 \cdot k) \rfloor + X \bmod(2 \cdot k)$; иначе $\{N_l^{(D)}(G_2) = 1$, если $X > 1$, то $N_l^{(v)}(G_2) = 1$, иначе $N_l^{(v)}(G_2) = 0\}$. Этап 2 для $l = \overline{h+1, L}$: если $X \leq 2 \cdot k$, то $\{N_l^{(D)}(G_2) = 1$, если $X > 1$, то $N_l^{(v)}(G_2) = 1$, иначе $N_l^{(v)}(G_2) = 0$; **выводится значение L** ; иначе $\{N_l^{(v)}(G_2) = \lfloor X/(2 \cdot k) \rfloor$, $N_l^{(D)}(G_2) = \lfloor X/(2 \cdot k) \rfloor + X \bmod(2 \cdot k)$ и $X := \lfloor X/(2 \cdot k) \rfloor + X \bmod(2 \cdot k)$. Оценка аппаратной сложности вычисления заданной НПФ вида (1) сводится к оценке количества вершин ЯПГ G . Возможна оценка количества вершин G на каждом из его ярусов, а также подсчет количества ярусов в зависимости от НПФ вида (1), задаваемой значениями $a_{i_1 \dots i_m} \in GF(2^k)$, $i_j = \overline{0, w}$, $j = \overline{1, m}$, $w = 2^k - 1$. Обозначим $Q(G(m, P))$ – множество ЭП внутри НПФ вида (1), соответствующих $G(m, P)$, подграфу ЯПГ G . Справедливо

Утверждение 3. $|Q(G(m, P))| = q - z$, где $q = \frac{m!}{(m - \sum_{t=1}^w p_t)! \cdot p_1! \dots p_w!}$, z –

количество нулевых коэффициентов во множестве элементарных полиномов, соответствующих $G(m, P)$, $w = 2^k - 1$.

На основе заданных значений коэффициентов в (1) вычисляемы множества P , а также $|Q(G(m, P))|$. На нулевом ярусе требуется m вершин G вида v_D , отображающих сохранение x_j , $j = \overline{1, m}$, – $N_0^{(D)}(G) = m$. Отображение вычисления x_j^2 , x_j^3 , ..., x_j^w , $j = \overline{1, m}$, $w = 2^k - 1$, на ярусе № 1 G требует по m вершин v_{02} , v_{03} , ..., v_{0w} , соответственно, а отображение сохранения x_j , $j = \overline{3, m}$, и x_j^2 , x_j^3 , ..., x_j^w , $j = \overline{1, m}$, – $\overline{N}_1^{(D)} = w \cdot m - 2$ вершин v_D . Отображение сохранения значений x_j^2 , x_j^3 , ..., x_j^w , $j = \overline{2, m}$, на ярусе № 2 G требует $\overline{N}_2^{(D)} = (w-1)(m-1)$ вершин v_D . Значения $\overline{N}_l^{(D)} = 0$, $l = \overline{3, h}$.

В результате, $N_1^{(v)} = m \cdot (w-1) + \sum_{\forall P} (|Q(G(m, P))| \cdot N_1^{(v)}(G(m, P)))$, $N_1^{(D)} = w \cdot m - 2 + \sum_{\forall P} (|Q(G(m, P))| \cdot N_1^{(D)}(G(m, P)))$.

На ярусе l ЯПГ G , $l = \overline{2, h}$, $h = 1 + \lceil \log_2 m \rceil$ (см. определение 3),
 $N_l^{(V)} = N_l^{(V)}(G_2) + \sum_{P: h(G(m, P)) \geq l} (|Q(G(m, P))| \cdot N_l^{(V)}(G(m, P)))$, $N_l^{(D)} = N_l^{(D)}(G_2) + \overline{N}_l^{(D)} +$
 $+ \sum_{P: h(G(m, P)) \geq l} (|Q(G(m, P))| \cdot N_l^{(D)}(G(m, P)))$. На ярусе l , $l = \overline{(h+1), L}$, $N_l^{(V)} = N_l^{(V)}(G_2)$,
 $N_l^{(D)} = N_l^{(D)}(G_2)$.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 15-01-03842.

Список литературы

1. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. – СПб. : БХВ-Петербург, 2002. – 600 с.
2. Кузелин, М. О. Современные семейства ПЛИС фирмы Xilinx : справ. пособие / М. О. Кузелин, Д. А. Кнышев, В. Ю. Зотов. – М. : Горячая линия – Телеком, 2004. – 440 с.
3. Высокопроизводительные реконфигурируемые вычислительные системы / А. И. Дордопуло, И. А. Каляев, И. И. Левин и др. // Суперкомпьютеры. – 2010. – № 3 (3). – С. 44–48.
4. Каляев, И. А. Реконфигурируемые мультимикровейерные вычислительные системы для решения потоковых задач обработки информации и управления / А. В. Каляев, И. И. Левин // Параллельные вычисления и задачи управления (РАСО'10) : пленарные докл. 5-й Междунар. конф., 26–28 окт. 2010. – М. : ИПУ РАН, 2010. – С. 23–37.
5. Virtex-4 Family Overview / Xilinx Inc. Cop. 2011. – URL: http://www.xilinx.com/support/documentation/data_sheets/ds112.pdf.
6. Virtex-7 FPGAs Data Sheet: DC and Switching Characteristics. DS183 (v1.0) March 1, 2011/ Xilinx Inc. Cop. 2011. – URL: http://www.xilinx.com/support/documentation/data_sheets/ds183_Virtex_7_Data_Sheet.pdf.
7. Захаров, В. М. Вычисление нелинейных полиномиальных функций на многопроцессорной системе с программируемой архитектурой / В. М. Захаров, С. В. Шалагин // Информационные технологии. – 2012. – № 5. – С. 6–11.
8. Шалагин, С. В. Сложность вычисления нелинейных полиномиальных функций над полем $GF((2^2)^k)$ на ПЛИС/FPGA / С. В. Шалагин // АКТО-2014 : сб. докл/ междунар. науч.-практ. конф. – Казань, 2014 – Т. II. – С. 661–664.
9. Пospelов, Д. А. Введение в теорию вычислительных систем / Д. А. Пospelов. – М. : Сов. радио, 1972. – 280 с.

Р. Н. Федюнин, А. С. Войнов, И. В. Сенокосов

Россия, Пенза, Пензенский государственный университет

ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ АЛГОРИТМОВ УМНОЖЕНИЯ БОЛЬШИХ ДАННЫХ С ПОМОЩЬЮ МАРКОВСКИХ ПРОЦЕССОВ

Для выбора оптимального варианта организации вычислений, необходима методика формальной оценки, позволяющая единообразно сопоставить различные решения с точки зрения сложности их реализации и скорости вычисления, которая была бы применима для оценки временной и аппаратной сложности вычислительных структур. В статье представлена методика оценки временной сложности алгоритмов умножения больших чисел.

Для исследования эффективности организации вычислений, с точки зрения временных затрат на уровне АЛУ, достаточно эффективен аппарат теории марковских процессов в следующей интерпретации [1]. В качестве структурной модели принимается модель АЛУ на рис. 1 (на основе организации представленной в [2]).

Вычислительный процесс в АЛУ представляется следующими состояниями марковского графа (рис. 1):

S_n – состояние, в котором находится вычислитель, до запуска вычислительного процесса;

S_1 – состояние характеризующее работу сортировщика входного потока даны;

$S_2 – S_7$ – состояния вычислительного процесса, характеризующие буферизацию входных данных с последующей ее обработкой и сохранением в регистрах временного хранения;

S_8 – состояние, в котором арбитр принимает решение направить данные на выход или обратно на обработку в ИУ;

S_k – финальное состояние, соответствующее полному завершению вычислительного процесса и его останов.

При этом:

P_{01} – вероятность перехода из состояния S_n в S_1 ;

$P_{12}, P_{13}, \dots, P_{17}$ – вероятности перехода из состояния S_1 (сортировщика данных) в состояния $P_2 – P_7$, соответственно ИУ1 – ИУ7 АЛУ;

$P_{28} – P_{78}$ – вероятности переходов из состояний $P_2 – P_7$, соответственно ИУ1 – ИУ7 АЛУ в состояние S_8 (арбитраж обработанных данных);

P_{81} – вероятность того что данные из арбитра S_8 снова уйдут на обработку S_1 (данная вероятность определяется спецификой реализации того или иного алгоритма вычислений);

P_{8k} – вероятность того что данные из арбитра S_8 обработаны и процесс их обработки окончен.

При условии, что данные поступают на вход по закону равномерного распределения [2].

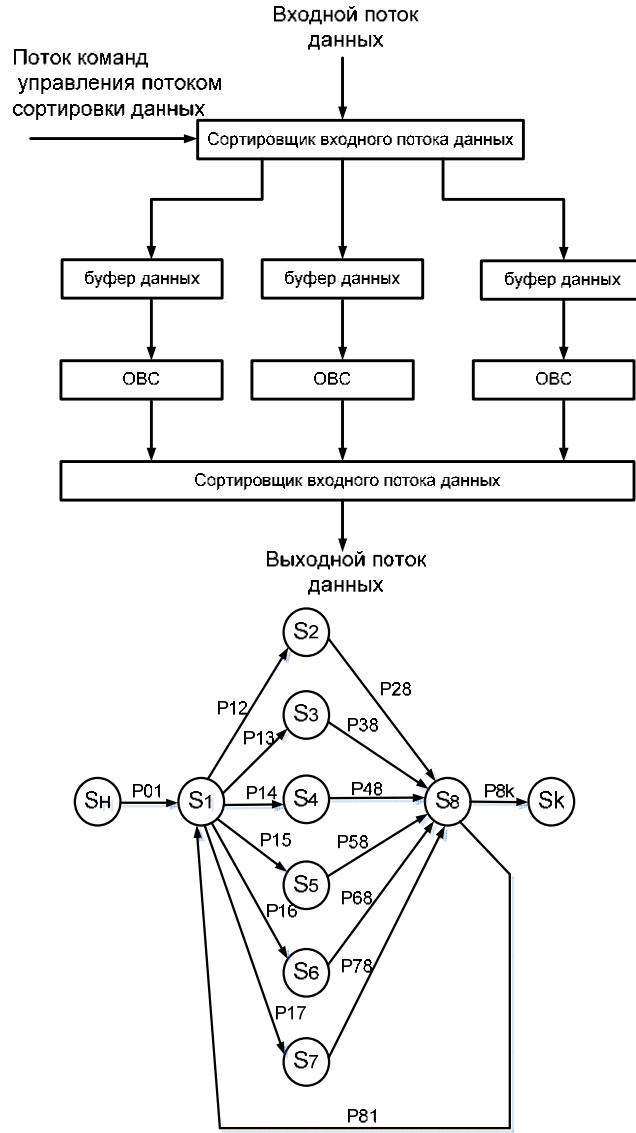


Рис. 1. Граф-схема вычислительного процесса в ИУ АЛУ

При однократной и результативной реализации процесса среднее число попаданий процесса d_i^{cp} в состояния $S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_k$ очевидно определить следующим образом (рис. 1):

$$d_1^{cp} = d_0 \times P_{0,1} + d_8 \times P_{8,1} \quad (1.5)$$

$$d_2^{cp} = d_1^{cp} \times P_{1,2} \quad (1.6)$$

$$d_3^{cp} = d_1^{cp} \times P_{1,3} \quad (1.7)$$

$$d_4^{cp} = d_1^{cp} \times P_{1,4} \quad (1.8)$$

$$d_5^{cp} = d_1^{cp} \times P_{1,5} \quad (1.9)$$

$$d_6^{cp} = d_1^{cp} \times P_{1,6} \quad (1.10)$$

$$d_7^{cp} = d_1^{cp} \times P_{1,7} \quad (1.11)$$

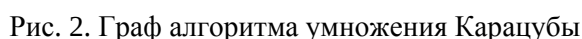
$$d_8^{cp} = d_2^{cp} \times P_{2,8} + d_3^{cp} \times P_{3,8} + d_4^{cp} \times P_{4,8} + d_5^{cp} \times P_{5,8} + d_6^{cp} \times P_{6,8} + d_7^{cp} \times P_{7,8} \quad (1.12)$$

$$d_k = d_8^{cp} \times P_{8,k} \quad (1.13)$$

Рассмотренная методика применима только для однородных неветвящихся процессов. В действительности же, алгоритмы арифметических операций ИУ АЛУ — процессы с ветвлениями и для оценки их временной сложности достаточно решение следующей системы уравнений:

$$\begin{aligned} & (p_{1,1}-1)\cdot d_1 + p_{2,1}\cdot d_2+...+p_{k-1,1}\cdot d_{k-1}= -1 \\ & p_{1,2}\cdot d_1+(p_{2,2}-1)\cdot d_2+...+p_{k-1,2}\cdot d_{k-1}=0 \tag{1.15}\\&\dots\dots\dots\\ &p_{1\ k-1}\cdot d_1+p_{2\ k-1}\cdot d_2+...(p_{k-1\ k-1}-1)\cdot d_{k-1}=0 \end{aligned}$$

На основании [3] – граф алгоритма умножения Карацубы имеет вид представленный на рис. 2.



15

Таблица 1

Система уравнений определения временной сложности алгоритма умножения Карацубы Q	Временная сложность алгоритма умножения Карацубы при $N = 8$	Временная сложность алгоритма умножения Карацубы при $N = 16$	Временная сложность алгоритма умножения Карацубы при $N = 32$	Временная сложность алгоритма умножения Карацубы при $N = 64$	Временная сложность алгоритма умножения Карацубы при $N = 128$
$-1 \cdot d_0 = -1$	1	1	1	1	1
$P_{0,1} \cdot d_0 - d_1 = 0$	1	1	1	1	1
$P_{1,2} \cdot d_1 + P_{3,2} \cdot d_3 - d_2 = 0$	4	8	16	32	64
$P_{2,3} \cdot d_2 - d_3 = 0$	3	7	15	31	63
$P_{2,4} \cdot d_2 - d_4 = 0$	1	1	1	1	1
$P_{4,5} \cdot d_4 + P_{6,5} \cdot d_6 - d_5 = 0$	4	8	16	32	64
$P_{5,6} \cdot d_5 - d_6 = 0$	3	7	15	31	63
$P_{5,7} \cdot d_5 - d_7 = 0$	1	1	1	1	1
$P_{7,8} \cdot d_7 + P_{9,8} \cdot d_9 - d_8 = 0$	4	8	16	32	64
$P_{8,9} \cdot d_8 - d_9 = 0$	3	7	15	31	63
$P_{8,10} \cdot d_8 - d_{10} = 0$	40	174	622	2064	6582
$P_{10,11} \cdot d_{10} - d_{11} = 0$	1	1	1	1	1
$P_{11,12} \cdot d_{11} + P_{13,12} \cdot d_{13} - d_{12} = 0$	4	8	16	32	64
$P_{12,13} \cdot d_{12} - d_{13} = 0$	3	7	15	31	63
$P_{12,14} \cdot d_{12} - d_{14} = 0$	1	1	1	1	1
$P_{14,15} \cdot d_{14} + P_{16,15} \cdot d_{16} - d_{15} = 0$	4	8	16	32	64
$P_{15,16} \cdot d_{15} - d_{16} = 0$	3	7	15	31	63
$P_{15,17} \cdot d_{15} - d_{17} = 0$	40	174	622	2064	6582
$P_{17,18} \cdot d_{17} - d_{18} = 0$	40	174	622	2064	6582
$P_{18,19} \cdot d_{18} - d_{19} = 0$	1	1	1	1	1
$P_{19,20} \cdot d_{19} - d_{20} = 0$	1	1	1	1	1
$P_{20,21} \cdot d_{20} - d_{21} = 0$	1	1	1	1	1
$P_{21,22} \cdot d_{21} + P_{23,22} \cdot d_{23} - d_{22} = 0$	4	8	16	32	64
$P_{22,23} \cdot d_{22} - d_{23} = 0$	3	7	15	31	63
$P_{22,24} \cdot d_{22} - d_{24} = 0$	1	1	1	1	1
$P_{24,25} \cdot d_{24} - d_{25} = 0$	1	1	1	1	1
$P_{25,26} \cdot d_{25} - d_{26} = 0$	1	1	1	1	1
Временная сложность алгоритма умножения Карацубы $Q = \sum_{k=0}^{k-1} d_k$, в тактах машинного времени	174	622	2064	6582	20520

В табл. 2 показаны временные сложности алгоритмов умножения больших чисел на примере обработки больших данных.

Таблица 2

Система уравнений определения временной сложности алгоритма умножения Карацубы Q	Временная сложность алгоритма умножения Карацубы при $N = 4096$	Временная сложность алгоритма умножения Карацубы при $N = 16384$	Временная сложность алгоритма умножения Карацубы при $N = 32768$	Временная сложность алгоритма умножения Карацубы при $N = 65536$	Временная сложность алгоритма умножения Карацубы при $N = 131072$
Временная сложность алгоритма умножения Карацубы $Q = \sum_{k=0}^{k-1} d_k$, в тактах машинного времени	5311182	48046422	144335880	433400862	1300989024
Временная сложность алгоритма умножения Тоома-Кука $Q = \sum_{k=0}^{k-1} d_k$, в тактах машинного времени	3509310	31785780	95519073	286881035	861290487
Временная сложность алгоритма умножения Штрассена $Q = \sum_{k=0}^{k-1} d_k$, в тактах машинного времени	5594451	46191507	135262371	399169491	1184279907

Графически это будет выглядеть следующим образом – рис. 2.



Рис. 2. Временная сложность алгоритмов умножения больших чисел

Следующим этапом исследования авторы планируют разработать матричный процессор на базе изученных алгоритмов с последующим макерированием данного устройства на ПЛИС.

Список литературы

1. Федюнин, Р. Н. Оценка пространственно-временной сложности и способы повышения скорости двоичных арифметических операций / Р. Н. Федюнин // Научное обозрение. – 2006. – № 3. – С. 100–111.
2. Федюнин, Р. Н. Способ реализации конвейерного сумматора / Р. Н. Федюнин // Новые информационные технологии : сб. тр. X Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2012. – С. 48–52.
3. Федюнин, Р. Н. Способ реализации аппаратного слоя VLIW архитектуры на базе систолических структур / Р. Н. Федюнин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2013. – № 2. – С. 15–22.
4. Свидетельство о государственной регистрации программы для ЭВМ № 2014619242 / Федюнин Р. Н., Медведик М. Ю., Москалева М. А., Войнов А. С., Сенокосов И. В. ; заявитель и патентообладатель ФГБОУ ВПО «ПГУ», г. Пенза. – Зарег. в Реестре программ для ЭВМ 11.09.2014.

Р. Н. Федюнин

Россия, Пенза, Пензенский государственный университет

СИСТОЛИЧЕСКИЕ СТРУКТУРЫ СРАВНИТЕЛЬНОГО АНАЛИЗА МАССИВА И ЭТАЛОНА

Процессоры арифметико-логической обработки данных на базе систолических вычислительных пространств наиболее эффективны при выполнении вычислений массивного класса в соответствующих прикладных областях. Соединяя алгебраическую гибкость и низкую сложность вычислений за счет возможности перенастройки аппаратной логики процессорных модулей данный класс устройств также имеет низкую стоимость реализации на основе современных технологий СБИС.

При выполнении арифметической обработки данных часто используются операции сравнения значений данных некоторого массива с аргументом-эталонном [1]. Наиболее типовыми массивными операциями данного класса являются операции выделения из массива подмножеств [2, 3], находящихся в отношениях равенства, меньше, больше некоторого аргумента-эталона, элементов с максимальными и минимальными значениями, ближайшего большего или меньшего значения данных относительно эталона, принадлежности к некоторому интервалу значений или пересечению интервалов.

Перечисленные типы операций эффективно реализуемы в матричных вычислительных пространствах с клеточной топологией, показанной на рис. 1.

Функционально процессорный модуль данного типа обеспечивает реализацию следующей системы элементарных логических функций:

1. $z_1^* = z_1 x_H \vee z_1 \bar{a} \vee z_1 w;$
2. $z_2^* = z_2 a x_B \vee z_2 \bar{a} \bar{x}_B;$
3. $w^* = w \vee x_1 \bar{a} x_H \vee z_2 a \bar{x}_B;$
4. $y_1^* = y_1 \vee z_1 a \bar{w};$
5. $y_2^* = y_2 \vee z_2 \bar{a};$
6. $x_H^* = \bar{z}_1 \bar{z}_2 w a \vee x_H (z_1 \vee z_2 \vee \bar{w});$
7. $x_B^* = \bar{z}_1 \bar{z}_2 w \bar{a} \vee x_B (z_1 \vee z_2 \vee \bar{w});$
8. $q^* = q;$
9. $a_1(t+1) = q a_1(t) \vee q (z_1 z_2 w y_1 \vee z_1 x_H \vee z_2 x_H \vee w x_H);$
10. $a_0(t+1) = \bar{q} a(t) \vee q (\bar{z}_1 \bar{z}_2 w y_1 \vee z_1 x_B \vee z_2 x_B \vee \bar{w} x_B);$

Соответственно переменные логических функций обозначают: a – значение состояния триггера памяти процессорного модуля; x_H, x_B – значения разрядов данного – эталона; z_1, z_2, w, y_1, y_2 – внешние и локальные управляющие переменные.

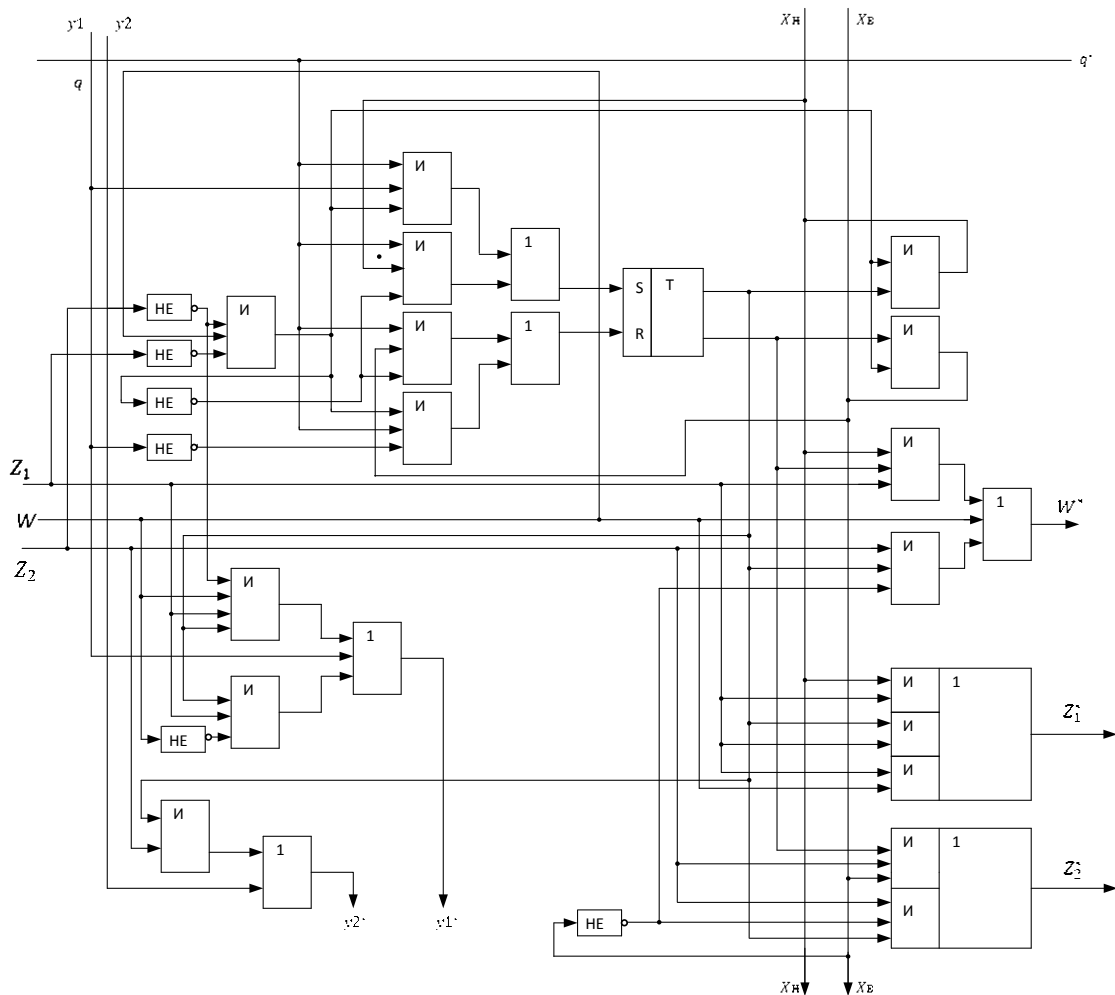


Рис. 1. Процессорный модуль систолической структуры

Вычислительное пространство формируется из процессорных модулей данного типа путем коммутации одноименных входных и выходных информационных и

управляющих каналов. При построении вычислений считается, что данные представлены в процессорной среде в виде двоичных чисел с горизонтальным размещением разрядов. Значения разрядов данных обрабатываемого массива размещены в триггерах внутренней памяти соответствующих модулей ПЭ. Внешними каналами управления и каналами данных являются соответствующие информационные и управляющие каналы правого крайнего столбца и крайней верхней строки вычислительной среды. При такой схеме организации вычислительного пространства, перечисленные операции выполняются следующим образом (рис. 2).

Приведем пример реализации операции в структуре рис. 2, на примере разбиения массива на подмассивы.

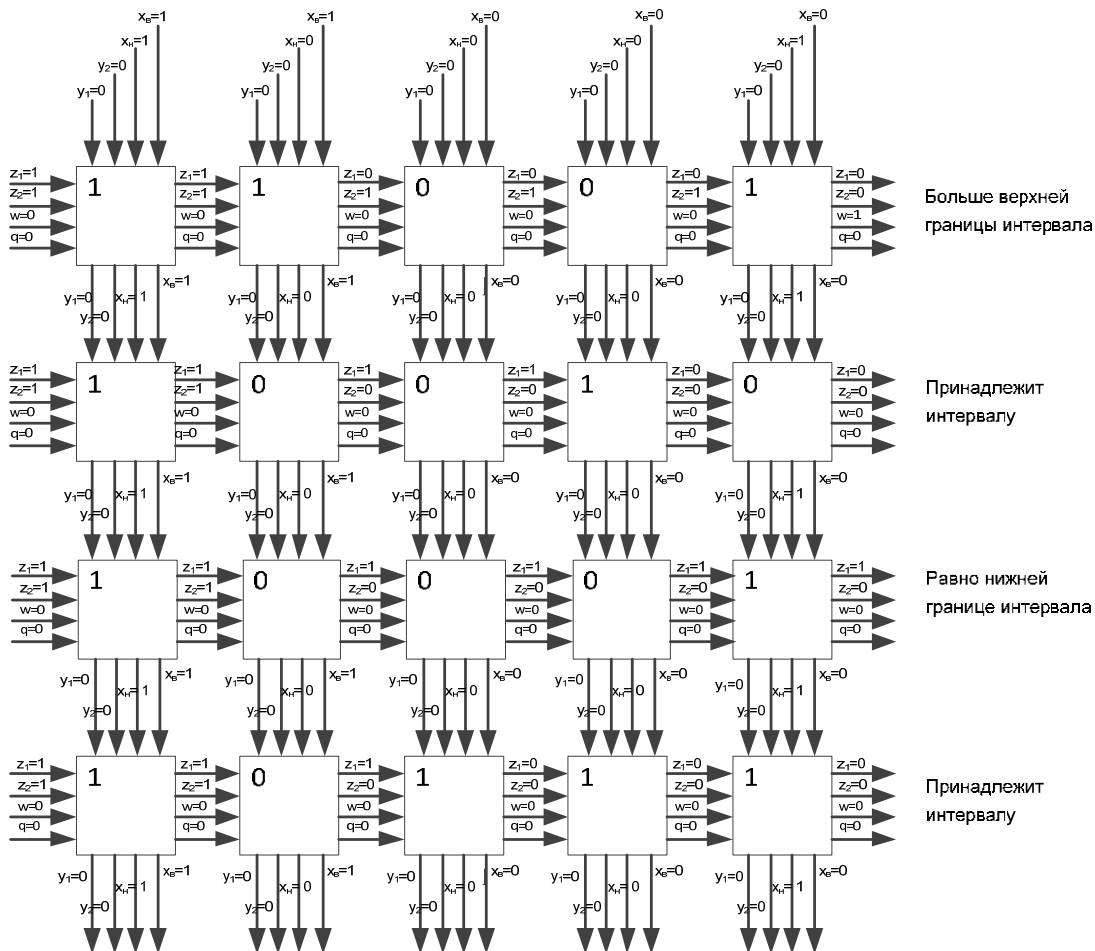


Рис. 2. Организация вычислительного пространства

Разбиение массива из m двоичных слов A на подмассивы по критериям «больше», «меньше», «равно», «соответствует интервалу значений» обеспечивается при установке на входах модулей ПЭ крайнего левого столбца сигналов управления $z_1 = 1, w = 0, z_2 = 1$, на входы модулей ПЭ верхней строки вычислительной среды X_H и X_B соответственно устанавливаются значения нижней и верхней границы интервала, а на входы q_i всех сток – $q_i = 0$. В этом случае в каждом модуле ПЭ реализуются функции:

$$\begin{aligned} z_1^* &= x_H \vee \bar{a}; z_2^* = ax_B \vee \bar{a} \bar{x}_B; w^* = \bar{a}x_H \vee ax_B; \\ y_2^* &= y_2 \vee \bar{a}; x_H^* = x_H; x_B^* = x_B; q^* = q; \end{aligned}$$

$$y_1^* = y_1 \vee a; a_1(t+1) = \bar{q}a_1(t) \vee qx_H; a_0(t+1) = \bar{q}a(t) \vee qx_B;$$

В результате вычислений в процессорных модулях крайнего правого столбца вычислительной среды на выходах $z_1^* z_2^*$ и w^* будут сформированы трехразрядные признаки-указатели значения отношения между соответствующим данной строке двоичным словом и двоичными значениями границ интервала (рис. 2).

Значения векторов-указателей z_1^*, z_2^*, w^* соответствуют следующим отношениям между двоичным словом A строки вычислительной среды и значениями границ интервала X_H и X_B :

1. $z_1^*, z_2^*, w^* = 001 \leftrightarrow A > X_B;$
2. $z_1^*, z_2^*, w^* = 101 \leftrightarrow A < X_H;$
3. $z_1^*, z_2^*, w^* = 000 \leftrightarrow X_H \leq A \leq X_B;$
4. $z_1^*, z_2^*, w^* = 100 \leftrightarrow A = X_H;$
5. $z_1^*, z_2^*, w^* = 010 \leftrightarrow A = X_B.$

В рамках данной статьи проведено исследование способа архитектурной и структурно-логической организации АЛУ для арифметико-логических вычислений, преобразования структур данных и арифметической обработки данных на основе массивных вычислений.

Список литературы

1. Федюнин, Р. Н. Способ реализации конвейерного сумматора / Р. Н. Федюнин // Новые информационные технологии : сб. тр. X Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2012. – С. 48–52.
2. Федюнин, Р. Н. Способ реализации аппаратного слоя VLIW архитектуры на базе систолических структур / Р. Н. Федюнин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2013. – № 2. – С. 15–22.
3. Свидетельство о государственной регистрации программы для ЭВМ № 2014619242 / Федюнин Р. Н., Медведик М. Ю., Москалева М. А., Войнов А. С., Сенокосов И. В. ; заявитель и патентообладатель ФГБОУ ВПО «ПГУ», г. Пенза. – Зарег. в Реестре программ для ЭВМ 11.09.2014.

АППАРАТНО-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СИСТЕМ УПРАВЛЕНИЯ

В. А. Чулков, С. А. Нестеренко

Россия, Пенза, Пензенский государственный технологический университет

ЛОКАЛЬНАЯ ФАЗОВАЯ ИНТЕРПОЛЯЦИЯ В ТЕХНИКЕ ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ

Рассмотрены технические принципы локальной фазовой интерполяции в системах цифрового преобразования и обработки информационных сигналов. Представлен вариант осуществления элемента ФИ с регулируемой осью интерполяции на основе сопряженно управляемых КМОП элементов задержки.

Локальная фазовая интерполяция (ФИ) применяется для сокращения шага дискретизации времени при оцифровывании информации путем создания ряда промежуточных точек отсчета на интервале между двумя событиями, представленными фронтами опорных колебаний [1]. Локальную ФИ следует рассматривать как средство достижения субвентильного разрешения, дополняющее процесс непрерывной ФИ.

Задача локальной интерполяции, которая решается элементами ФИ, состоит в получении колебания с фазой, средней между двумя опорными колебаниями. Процесс локальной ФИ можно представить взвешенным сложением двух опорных колебаний x и y с единичной амплитудой:

$$z(t) = ax(t) + (1-a)y(t),$$

где $a \in 0...1$ – управляющее воздействие. Положим для примера, что $x(t)$ и $y(t)$ – квадратурные гармонические колебания $\varphi_x = 0, \varphi_y = \pi/2$. Тогда

$$Z_m = \sqrt{1-2a(1-a)},$$
$$\varphi_z = \arctg \frac{Y_m}{X_m} = \arctg \left(\frac{1-a}{a} \right).$$

Элемент локальной ФИ подобен фазовому модулятору, который при изменении воздействия « a » от 0 до 1 меняет фазу выходного колебания в пределах от 0 до $\pi/2$. При $a = 1/2$ фаза результирующего колебания z будет $\varphi_z = \pi/4$.

Получить интерполирующий сигнал можно и путем сложения негармонических колебаний. Диаграммы, показанные на рис. 1, иллюстрируют процесс сложения кусочно-линейных колебаний $x(t)$ и $y(t)$, в результате которого получается кусочно-линейное колебание $z(t)$. Фаза $z(t)$ определяется управляющим воздействием a :

$$\varphi_z = (1-a)\pi/2.$$

Если опорные колебания имеют трапецеидальную форму с фронтами t_r , то путем сложения можно получить «среднюю» фазу только в случае, когда их фронты хотя бы частично перекрываются во времени. В противном случае ($|\varphi_x - \varphi_y| > \omega t_r$) момент пересечения нуля выходным сигналом оказывается неопределенным. Диапа-

зон регулирования позиции результирующего колебания здесь ограничен длительностями фронтов опорных сигналов.

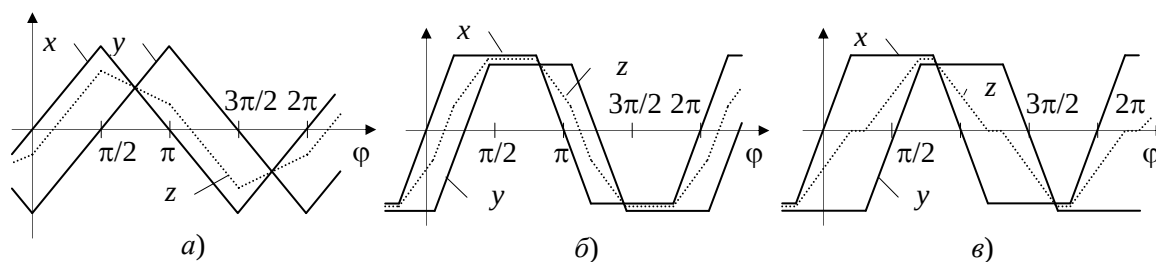


Рис. 1. Сложение опорных колебаний формы:
 а – треугольной; б – трапецидальной с перекрывающимися фронтами;
 в – трапецидальной с неперекрывающимися фронтами

Технически сложение колебаний можно выполнить разными способами, суммируя либо напряжения, либо токи опорных источников. В КМОП базисе это достигается простым объединением выходов каскадов задержки опорных колебаний, поскольку такое объединение, в отличие от биполярных аналогов, не вызывает опасных сквозных токов. На рис. 2 в качестве примера управляемого ЭЗ представлена схема КМОП инвертора с ограничением тока. Задержка определяется процессом перезаряда емкости нагрузки C_H выходным током инвертора на транзисторах T_{p1} , T_{n1} , который задается дополнительными комплементарными транзисторами T_{p2} , T_{n2} , образующими вместе с транзисторами T'_{p2} , T'_{n2} схему токового зеркала. Ток в T_{p2} при любом значении управляющего напряжения U_y равен току, протекающему в T_{n2} , что обеспечивает равенство втекающего и вытекающего токов инвертора и, следовательно, симметрию задержки входного сигнала U_L , значения токов регулируется управляющим напряжением. Исходя из комплементарности транзисторов с противоположными типами каналов $|K_n| = |K_p|$, $|U_{tn}| = |U_{tp}|$, можно найти связь времени задержки ЭЗ с управляющим напряжением [2]:

$$t_{PLH} = t_{PHL} = \frac{C_H}{K_n (U_{tn} - U_y)^2}.$$

Схема элемента ФИ с управляемой осью интерполяции, показанная на рис. 2,б, включает пару описанных ЭЗ с общим сопряженным управлением напряжением U_y и его инверсией $-U_y$, отсчитываемыми от уровня опорного напряжения U_{ref} . Для того, чтобы интерполирующий сигнал Z располагался между опорными колебаниями X и Y , последние также задерживаются своими ЭЗ, настроенными на среднюю задержку (в схеме не показаны). Результаты моделирования схемы в среде *LTSpice* (рис. 2,в) свидетельствуют о значительном диапазоне регулирования оси интерполяции.

В ряде применений предпочтительными оказываются неинвертирующие дифференциальные элементы ФИ, которые хотя и не обладают полным размахом напряжения, характеризуются лучшей помехоустойчивостью. Каскадирование элементов ФИ позволяет сократить шаг дискретизации времени до значений, ограничиваемых лишь шумовыми факторами.

Применение локальной ФИ обеспечивает сокращение кванта времени до субвентильных значений, что позволяет существенно повысить точность преобразователей информации. На элементах ФИ несложно построить генератор с электронной

перестройкой частоты, прецизионный цифровой преобразователь интервалов времени, частотный синтезатор, устройство фазовой синхронизации.

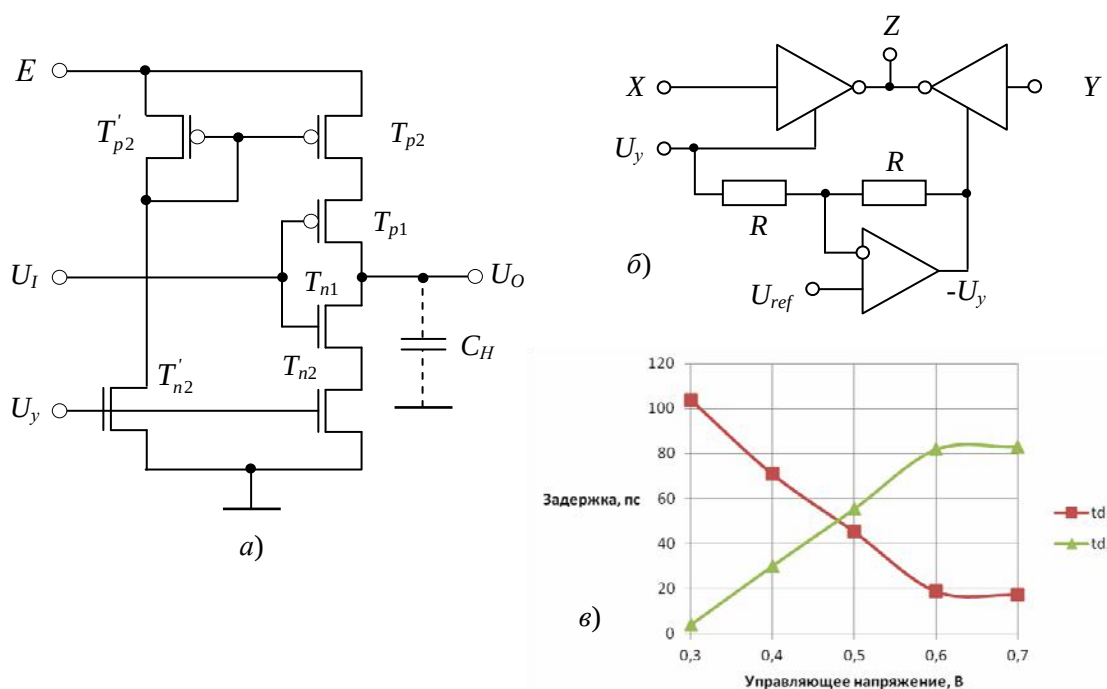


Рис. 2. Управляемый элемент симметричной задержки (а), построенная на его базе схема элемента ФИ (б) и регулировочная характеристика элемента ФИ

Список литературы

1. Чулков, В. А. Интерполирующие устройства синхронизации и преобразователи информации / В. А. Чулков. – М. : Физматлит, 2010. – 328 с.
2. Бутаев, М. М. Управляемые КМОП элементы задержки для интерполирующих преобразователей информации / М. М. Бутаев, В. А. Чулков // Вопросы радиоэлектроники. Сер. ЭВТ. – 2010. – Вып. 5. – С. 130–140.

В. А. Чулков, С. А. Нестеренко

Россия, Пенза, Пензенский государственный технологический университет

НЕПРЕРЫВНАЯ ФАЗОВАЯ ИНТЕРПОЛЯЦИЯ В ТЕХНИКЕ ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ

Рассмотрены технические принципы непрерывной фазовой интерполяции в системах цифрового преобразования и обработки информационных сигналов. Представлены базовые структуры интерполирующих устройств на основе микросхемной цифровой линии задержки.

Фазовая интерполяция (ФИ) применяется для повышения точности цифрового представления информации путем сокращения шага дискретизации времени с созданием ряда промежуточных точек отсчета на интервалах между двумя событиями. В отличие от интерполяции в строгом математическом смысле ФИ в технических системах представляет собой именно «фазовую» процедуру: если изменяется ин-

терполируемый интервал, то изменяется и протяженность субквантов интерполяции, а их число на интервале остается неизменным [1]. В соответствии с решаемой задачей можно говорить о непрерывной (круговой) интерполяции тактового периода и о локальной интерполяции с целью создания оси интерполяции между двумя опорными колебаниями.

Непрерывная ФИ осуществляется с помощью цифровых линий задержки (ЦЛЗ) и мультифазных кольцевых генераторов (МКГ), перекрывающих тактовый период совокупностью импульсов своих каскадов – буферов задержки. Стабилизация генерируемого субкванта времени достигается введением обратных связей, преобразующих ЦЛЗ в систему автоподстройки задержки *DLL (Delay Lock Loop)*, а МКГ – в систему фазовой синхронизации *PLL (Phase Lock Loop)*.

В процессе оцифровывания интервала времени $t_1 \dots t_2$ между двумя событиями выполняется по существу процедура дискретизации прямоугольного импульса с единичной амплитудой, результатом которой оказывается пачка δ -импульсов

$$s_d(t) = [\sigma(t_1) - \sigma(t_2)] \sum_{k=-\infty}^{\infty} \delta(t - kT_O). \quad (1)$$

Поскольку сигнал за пределами отрезка $t_1 \dots t_2$ равен нулю, то (1) можно переписать также в виде ряда с ограниченным числом членов

$$s_d(t) = \sum_{k=\text{ent}\{t_1/T_O\}+1}^{\text{ent}\{t_2/T_O\}} \delta(t - kT_O), \quad (2)$$

где $\text{ent}\{x\}$ – операция взятия целой части аргумента.

В отличие от традиционной импульсной системы с единственным отсчетом за период дискретизации (рис.1а), при ФИ в каждом опорном периоде генерируется N δ -импульсов (рис. 1,б). Позиции импульсов ввиду аккумуляирования джиттера могут быть искажены с нарастающей в пределах периода степенью искажения (рис. 1,в). Множество N отсчетов образуют N частных решетчатых «образов» сигнала:

$$\left. \begin{aligned} s_{d0}(t) &= [\sigma(t_1) - \sigma(t_2)] \sum_{k=-\infty}^{\infty} \delta(t - kT_O); \\ s_{d1}(t) &= [\sigma(t_1) - \sigma(t_2)] \sum_{k=-\infty}^{\infty} \delta(t - kT_O - T_O/N); \\ s_{d2}(t) &= [\sigma(t_1) - \sigma(t_2)] \sum_{k=-\infty}^{\infty} \delta(t - kT_O - 2T_O/N); \\ &\dots \\ s_{d(N-1)}(t) &= [\sigma(t_1) - \sigma(t_2)] \sum_{k=-\infty}^{\infty} \delta(t - kT_O - (N-1)T_O/N). \end{aligned} \right\} \quad (3)$$

Полный «образ» сигнала получается сложением частных решетчатых функций:

$$s_d(t) = [\sigma(t_1) - \sigma(t_2)] \sum_{i=0}^{N-1} \sum_{k=-\infty}^{\infty} \delta\left(t - \left(k + \frac{i}{N}\right)T_O\right). \quad (4)$$

Формула (4) в общем виде описывает процедуру непрерывной ФИ в процессе преобразования интервала времени в цифровой код. Поскольку по сравнению с (2)

здесь предусматривается в N раз большее количество отсчетов сигнала, то применение ФИ эквивалентно повышению частоты отсчетов в N раз.

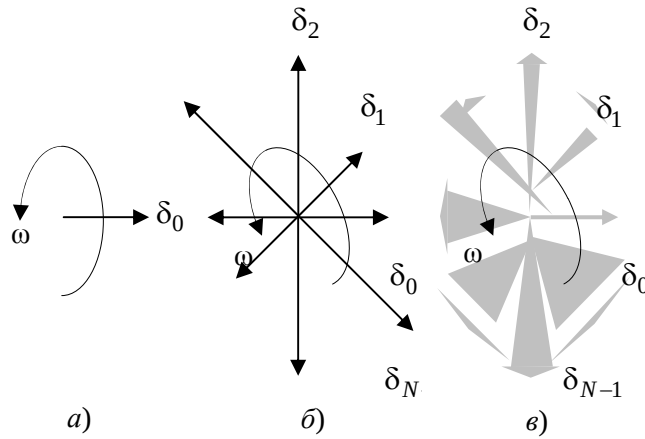


Рис. 1. Фазовые диаграммы дискретизации времени:
 a – традиционная; b – с интерполяцией; $в$ – с «шумящей» интерполяцией

Устройства DLL и PLL строятся на базе микроселекционных ЦЛЗ, звеньями которых служат управляемые элементы задержки (ЭЗ). Благодаря развитию микроэлектроники, давшему свет низковольтным МОП приборам и снявшему ограничения на сложность схем ЭЗ, стало возможным применение для задержки цифровых сигналов собственных задержек логических вентилей, снабженных цепями электронной перестройки [2].

Основное различие между устройствами DLL и PLL состоит в том, что в первом нет местного генератора, его выходные импульсы являются задержанными копиями входного сигнала, в то время как устройство PLL является автогенератором с обратной связью по фазе.

В структуре устройства PLL (рис. 2, a) обратная связь генератора выполняется с единственного его выхода, совокупность импульсов на промежуточных выходах МКГ интерполирует их период. В интегральных схемах PLL фазовая ошибка представляется преимущественно шириной импульсов, управляющих накачкой заряда интегрирующего конденсатора (в зарубежной литературе *Charge Pump PLL* [3]). Обозначив общий коэффициент передачи не включающих реактивностей ФК и БНЗ через $K_{\Phi K}$ с размерностью А/рад и коэффициент передачи МКГ как K_I/s с размерностью рад/(В·с) (оператор s отражает интегральную связь выхода с входом), можно найти передаточную функцию замкнутой системы PLL в виде

$$W(s) = \frac{K(s)}{1 + \beta K(s)} = \frac{K_{\Phi K} K_I (1 + sRC)}{C(s^2 + s \cdot \beta K_{\Phi K} K_I R + \beta K_{\Phi K} K_I / C)}. \quad (5)$$

Основные свойства PLL определяются характеристическим уравнением $\Delta(s) = s^2 + 2\zeta\omega_n s + \omega_n^2$, где $\omega_n = \sqrt{\beta K_{\Phi K} K_I / C}$ – собственная частота системы, $\zeta = \beta K_{\Phi K} K_I R / 2\omega_n$ – отношение демпфирования. Полюса передаточной функции

$$P_1 = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2} = -\alpha + j\omega, \quad P_2 = -\zeta\omega_n - j\omega_n\sqrt{1-\zeta^2} = -\alpha - j\omega,$$

($\alpha = \zeta\omega_n$ – коэффициент демпфирования, $\omega = \omega_n\sqrt{1-\zeta^2}$ – демпфированная частота) определяют динамику переходного процесса при скачке фазы входного сигнала: если $\alpha < 1$ – колебательный, если $\alpha > 1$ – аperiodический. Отметим, что устройство *PLL* даже с простейшим фильтром является системой как минимум второго порядка.

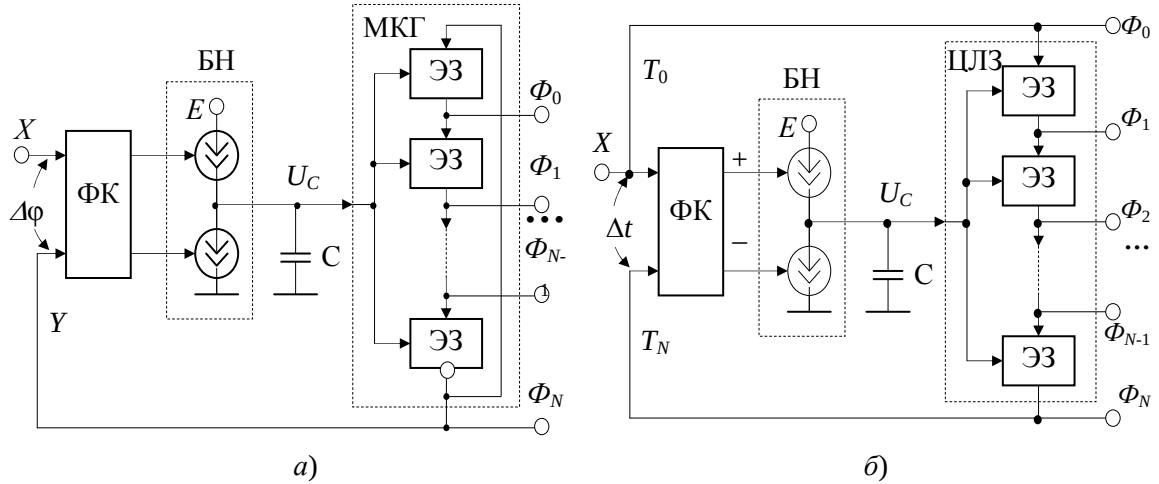


Рис. 2. Структуры систем *PLL* (а) и *DLL* (б) с блоком накачки заряда

Устройства автоподстройки задержки *DLL* получили распространение в высокоскоростных системах обработки данных как альтернатива устройствам *PLL* благодаря меньшей чувствительности к технологическим отклонениям, питающему напряжению и температуре. Достоинством *DLL* является также ослабление эффекта накопления джиттера, поскольку такое накопление возможно только в пределах опорного периода. Структура *DLL* (рис.2б) включает управляемую ЦЛЗ, каждое из N звеньев которой задерживает входной сигнал X на время t_D , а также петлю автоподстройки задержки в составе ФК и БНЗ. Фазовая разность между опорным сигналом и выходным сигналом цепи ЭЗ с помощью петли обратной связи сводится к нулю. Подстройка задержки осуществляется путем регулирования времени задержки t_D всех ЭЗ по результату Δt фазового сравнения в ФК:

$$\Delta t = \sum_{i=1}^N t_{Di} - T_0,$$

где t_{Di} – время задержки i -го ЭЗ, а T_0 – опорный период. В режиме синхронизации общее время задержки ЦЛЗ точно равно периоду ОГ, и если ЭЗ идентичны, то $\sum_{i=1}^N t_{Di} = Nt_D = T_0$. Идентичность ЭЗ означает не только равенство их задержек, но и одинаковые обязательно монотонные регулировочные характеристики $t_D(U_C)$.

Передаточная функция ЦЛЗ в отличие от передаточной функции генератора в устройстве *PLL* не содержит интегрирующего оператора и для относительно малых фазовых ошибок представляет собой постоянный коэффициент $K_D = Nk_D$ (с/В), где k_D – крутизна регулировочной характеристики ЭЗ. БНЗ преобразует импульс напряжения ФК в заряд $\Delta q = I_q \cdot \Delta t$, где I_q – амплитуда тока БНЗ, а Δt – временная разность между сигналами ФК. Коэффициент передачи БНЗ равен, следовательно,

I_q (А). Последовательность импульсов БНЗ преобразуется далее в управляющее напряжение U_C с помощью интегрирующего фильтра с передаточной функцией $K_F(s) = K_F/s$, где K_F – коэффициент усиления фильтра по постоянному току. Таким образом, передаточная функция малосигнальной модели замкнутой системы *DLL*

$$W(s) = T_N(s)/T_O(s) = K(s)/[1 + K(s)] = I_q K_F K_D / (I_q K_F K_D + s). \quad (6)$$

соответствует системе первого порядка, обладающей абсолютной устойчивостью и полосой захвата, равной полосе удержания.

Точность *DLL* определяется степенью согласования времен задержки звеньев ЦЛЗ. Можно показать, что дисперсия технологического отклонения задержки изменяется с номером m отвода ЦЛЗ по закону

$$\sigma_{tm}^2 = \frac{m(N-m)}{N} \sigma_T^2,$$

достигая максимума в середине ЦЛЗ ($m = N/2$).

Список литературы

1. Чулков, В. А. Интерполирующие устройства синхронизации и преобразователи информации / В. А. Чулков. – М. : Физматлит, 2010. – 328 с.
2. Бутаев, М. М. Управляемые КМОП элементы задержки для интерполирующих преобразователей информации / М. М. Бутаев, В. А. Чулков // Вопросы радиоэлектроники. Сер. ЭВТ. – 2010. – Вып. 5. – С. 130–140.
3. Gardner, F. M. Charge-Pump Phase-Locked Loops / F. M. Gardner // IEEE Transactions on Communications. – November 1980. – Vol. COM-28, no. 11. – P. 1849–1858.

Д. А. Уваров

Россия, Пенза, Пензенский государственный университет

СТАНДАРТ ИЕС 61499, ФУНКЦИОНАЛЬНЫЕ БЛОКИ ИЕС 61499, РАСПРЕДЕЛЕННАЯ МОДЕЛЬ СИСТЕМЫ

Рассматривается стандарт ИЕС 61499, предназначенный для унификации правил создания распределенных приложений и применения функциональных блоков, а также описание распределенной модели системы с использованием стандарта ИЕС 61499 вместе с распределенными функциональными блоками из нескольких устройств.

Введение

В настоящее время происходит процесс перехода к системе для распределенной системы управления, т.к. при глобализации рынка нужно подстраивать промышленное производство под постоянно меняющиеся условия торговли. Современное производство и продукция должны быстро меняться, оперативно реагировать на быстро меняющиеся требования рынка и изменение производства и технологий. Для этого используется стандарт ИЕС 61499, предназначенный для раз-

работки гибких распределенных приложений в промышленных процессах, системах измерения и контроля. Функциональный блок является абстрактной моделью его и устройств, приложений и ресурсов, являющихся ключевыми элементами распределенного приложения под IEC 61499.

Новый международный стандарт IEC 61499 позволяет внедрить модель для реализации распределенной системы управления с учетом данных особенностей.

Стандарт IEC 61499

Стандарт IEC 61499 определяет распределенную, управляемую событиями архитектуру и требования к программному инструментарию для инкапсуляции, встраивания, развертывания и интеграции программного обеспечения в интеллектуальных устройствах, машинах и системах. Он обозначает распределенную модель как совокупность функциональных блоков, на которые разбиваются различные части промышленного процесса автоматизации и сложной системы управления. Эти функциональные блоки могут распределяться и взаимодействовать по коммуникационной сети через множество контроллеров. Приложение становится распределенным путем размещения экземпляров функциональных блоков на различных ресурсах в одном или более устройствах. Функциональные блоки являются атомарными элементами распределения. Приложение со многими функциональными блоками отображается как один элемент, хотя экземпляры функциональных блоков распределяются по ресурсам и устройствам.

В стандарте модели IEC 61499 распределение приложений строится путем соединения экземпляров многообразных функциональных типов блоков с соответствующими связями событий и данных. При использовании IEC 61499-совместимых программных средств, эти функциональные блоки могут быть распределены с физическими устройствами по сети.

Функциональные блоки IEC 61499

Функциональный блок, ФБ (*Function block*) – программная функциональная единица, которая является наименьшим элементом в распределенной системе управления. Функциональный блок использует машину состояний (*state machine*) с диаграммой управления выполнением (*execution control chart – ECC*) для задания алгоритма работы ФБ. На рис. 1 приведен дизайн функционального блока IEC 61499.

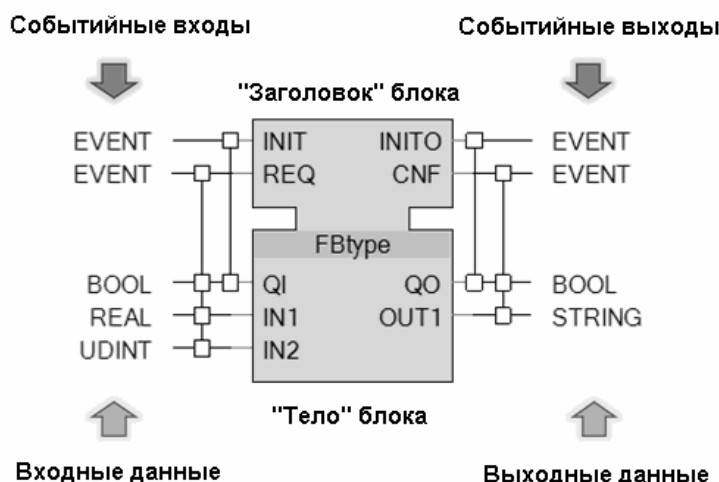


Рис. 1. Дизайн функционального блока IEC 61499

Различают базовые и композиционные функциональные блоки IEC 61499. Композиционные функциональные блоки – это набор базовых функциональных блоков IEC 61499. Для определения базового блока надо задать входные и выходные переменные, входные и выходные события, диаграмму управления выполнением и алгоритм блока. Диаграмма управления выполнением – это описание реакций на внешние воздействия, в котором задается, что именно нужно сделать, если произошло конкретное событие. Входные события инициируют и управляют работой функционального блока IEC 61499. Помимо внешних (входных) событий указывается, какие события будут генерироваться при окончании выполнения функций-обработчиков. Наиболее удобным языком для создания диаграмм управления является SFC (язык последовательных функциональных схем).

В стандарте IEC 61499 определены 18 типов функциональных блоков: расщепление события, слияние, рандеву двух событий, разрешающее распространение события, выбор между двумя событиями, переключение событий, задержка событий и другие.

Ключевой особенностью функциональных блоков IEC 61499 является управление самими блоками с помощью внешних событий.

Приложение может содержать один или более циклов управления, т.е., ввод данных выполняется на одном устройстве, управление – на втором, а преобразование выходных данных – на третьем устройстве.

Распределенная модель системы в IEC 61499

Понятие распределения приносит изменения в ускорении автоматизации на производственных и перерабатывающих отраслях промышленности. Стандарт IEC 61499 позволяет проектировать и разрабатывать гибкие распределенные системы управления/приложения путем слияния особенностей функциональных блоков и абстрактной модели функционального блока. Он определяет общую модель и методологию для описания функциональных блоков в формате, который не зависит от конкретной реализации приложений. Эта модель используется дизайнерами для разработки распределенных приложений управления промышленностью. В основном, применение на основе этого стандарта позволяет системам объединять вместе функциональные блоки, разделенные между различными ресурсами. Этот стандарт подчеркнул формальные модели и методы, основанные на объектно-ориентированной концепции и единого языка моделирования.

На рис. 2 показана распределенная модель системы в соответствии со стандартом IEC 61499.

Устройство, применение и ресурсы являются основными элементами распределенной системы управления технологическими процессами на основе модельной структуры IEC 61499. Устройство является основным элементом управления, которое состоит из процессора, входного/выходного взаимодействия и коммуникационного интерфейса. Приложение является группой функциональных блоков, которые сообщаются друг с другом для выполнения задач управления. Ресурс рассматривается как под-устройство, которое может независимо контролировать его работу и часть распределенного приложения работать на ресурсы.

Стандарт IEC 61499 облегчает международную торговлю путем ликвидации централизованных барьеров, обеспечивая мобильность, возможности настройки, реконфигурации и способность к взаимодействию.

Алгоритм функционального блока может поддерживать функциональные блоки, лестничные диаграммы, текстовые сообщения и Java. В данном смысле про-

исходит обеспечение реконфигурации на абстрактном уровне функционального блока IEC стандарта 61499.

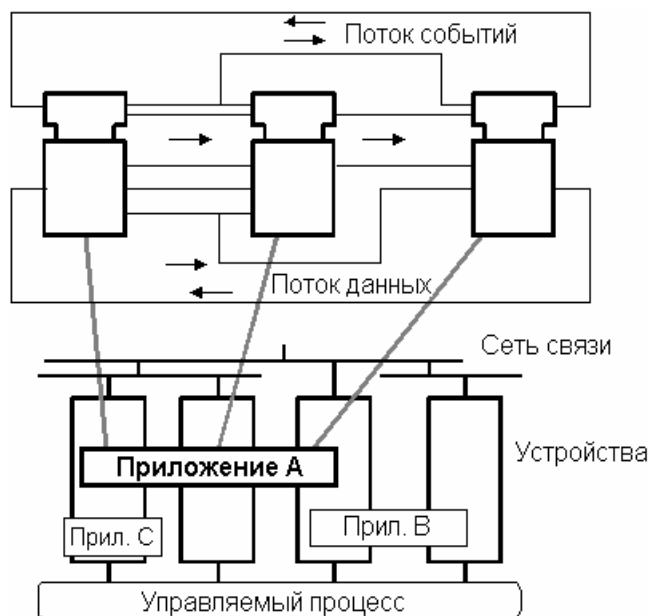


Рис. 2. Распределенная модель системы в стандарте IEC 61499.

Инструмент разработки FBDK

Существуют различные программы для проектирования и разработки распределенных приложения стандарта IEC 61499. Одним из них является средство FBDK (function block development kit). Оно было разработано компаниями Rockwell Automation и Holobloc inc. в США.

FBDK обеспечивает графическую платформу, через которую может быть запрограммировано и смоделировано IEC 61499-приложение. Создание функциональных блоков производится с помощью комплекта Java Development Kit (JDK). Приложение позволяет также разрабатывать сетевые функциональные блоки и дает пользователю определить функциональные блоки в проектировании распределенных приложений управления.

Управляющее программное обеспечение может поддерживать стандартные библиотеки функциональных блоков. Можно также проектировать отдельные устройства, блоки интерфейса стандарта обслуживания, стандартные блоки для создания связей между устройствами.

Ключевыми особенностями программного инструмента FBDK являются:

1) Просмотр – предусматриваются три вида просмотров в окне редактирования: в первом окне находится дерево приложений, состоящее из приложений, устройств и ресурсов. Второе окно рабочего пространства из FBDK редактора позволяет определить и разработать различные функциональные блоки сети с помощью соответствующих соединительных событий и данных. Третье окно располагается снизу, и оно имеет текстовый вид, показывающий либо соответствующий исходный код, либо эквивалентный исходный текст XML.

2) Графическая и текстовая разработка – текстовый вид позволяет просматривать и изменять XML исходного текста или исходного кода. Любые изменения, внесенные в текст, отражаются в графическом представлении.

3) Взаимодействие пользователя – существует только один способ изменить свойства функционального блока. Для этого нажимается правой кнопкой мыши в графической среде и выбирается определенное действие из контекстного меню.

4) Механизм библиотек – при разработке составного ФБ необходимо создавать и изменять его сетевую составляющую. Новые функциональные блоки вставляются в сеть с помощью контекстного меню (NEW FB). Затем появляется мастер вставки, где пользователь может выбрать файл, который хранит еще функциональный блок и дать ему название.

С. В. Тумасов, Е. К. Таранцев, А. С. Бычков

Россия, Пенза, Пензенский государственный университет

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ SDR-РАДИОСРЕДСТВОМ

Разработанное программное обеспечение позволяет через сеть Ethernet управлять SDR-радиосредством, рассчитывать и отображать спектр принимаемого сигнала.

Разработчики и производители беспроводного оборудования уделяют много внимания технологии программно-конфигурируемого радио (Software Defined Radio – SDR), позволяющей изменять эксплуатационные параметры радиооборудования на уровне ПО(программное обеспечение) без изменения электрической схемы [1].

Ниже на рис. 1 представлена обобщенная структура SDR радиооборудования.

SDR включает аппаратную и программную части.

Аппаратная состоит из трех функциональных блоков:

- RF-секция;
- IF-секция;
- секция базовой станции.

– RF-секция (называемая также RF front-end) включает только аналоговые аппаратные модули (в то время как две другие целиком содержат цифровую аппаратуру) и отвечает за передачу/прием радиосигнала. IF-секция отвечает за цифро-аналоговое преобразование и модуляцию/демодуляцию сигнала.

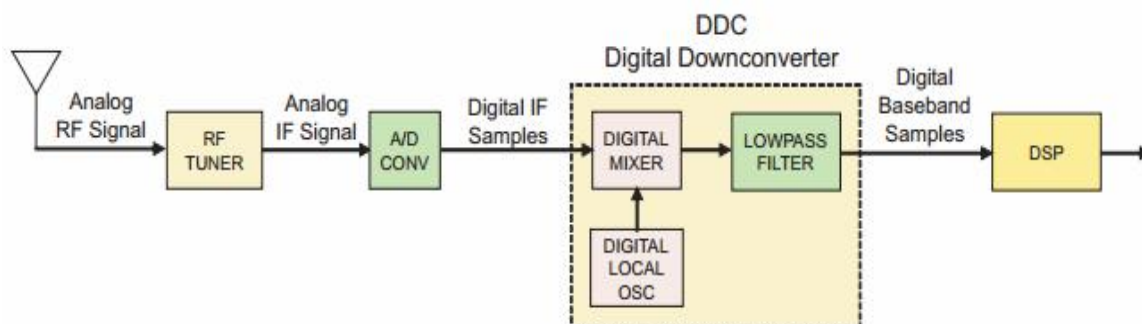


Рис. 1. Обобщенная структура SDR-радиооборудования

Секция базовой станции, содержащая протокол второго уровня (Layer 2, согласно семиуровневой модели OSI), выполняет базовые операции связи: установку сеанса связи, уравнивание, хоппинг, восстановление тайминга, корреляцию и т.д. В программно-управляемом радио протокол второго уровня и операции модуляции/демодуляции реализованы программно.

Такая концепция выполнения радиооборудования является альтернативой традиционным многодиапазонным и многомодовым устройствам, в которых для каждого диапазона, стандарта или технологии используются отдельные специализированные компоненты или даже тракты блока радиочастот (РЧ).

При разработке SDR-радиосредства одной из важных проблем является обеспечение его управления. Переконфигурация радиосредства может проводиться как на низком уровне (прошивка ПЛИС), так и на высоком уровне (например, C++).

Также важной проблемой является обеспечение анализа работы радиосредства во время его разработки и тестирования. Анализ может быть осуществлен на низком (ChipScore), так и на высоком уровне (на высокоуровневом языке программирования).

В данной статье представлен вариант решения изложенных проблем, связанный с привлечением системного языка программирования C++, на примере разработки конкретного комплекса программ. При этом программная часть SDR-радиосредства реализована на ПЛИС.

В начале процесса разработки ПО нужно выбрать средства разработки. При этом необходимо учитывать требования к операционной системе, для которой будет создаваться ПО, наличие классов, обеспечивающих требуемый функционал.

В рассматриваемом варианте решения ПО должно обладать кроссплатформенностью и обеспечивать обмен данными между ПК и ПЛИС по сети Ethernet с помощью протокола TCP/IP. Основываясь на этих требованиях, можно в качестве средства разработки выбрать инструмент Qt. Использование библиотек Qt позволяет запускать написанное с его помощью ПО в большинстве современных операционных систем путем простой компиляции программы для каждой ОС без изменения исходного кода. Библиотеки Qt упрощают работу с TCP/IP протоколами, поскольку она имеет огромное количество довольно абстрактных классов для работы с протоколами передачи данных, например, QUdpSocket [2].

Одним из основных вариантов анализа работы радиосредства является оценка спектра его выходного сигнала. Чтобы обеспечить этот метод анализа, необходимо реализовать быстрый расчет спектра сигнала и его отображение. Наиболее подходящим для этого средством как раз является библиотека OpenGL, поскольку трансформации и растеризация в ней выполняются графическим конвейером, который позволяет обновлять кадры с частотой, в разы превышающей частоту обновления кадров, если бы мы пользовались стандартными средствами Qt [3].

Ниже на рис. 2 представлена схема, отражающая на абстрактном уровне вариант работы ПО по управлению радиосредством и анализа его работы.

Таким образом, пользователь данного ПО управляет работой ПЛИС, задавая значения несущей частоты, и получает на выходе графическое отображение сигнала, который будет истинным на данный момент времени.

Передача данных осуществляется UDP-пакетами (датаграммами). Как известно, UDP является протоколом транспортного уровня семейства TCP/IP, который в свою очередь определяет, для какого именно приложения предназначены передаваемые данные. Он считается протоколом «ненадежной» передачи, в смысле невозможности удостовериться в доставке сообщения адресату, а также возможного

перемешивания пакетов. UDP обеспечивает многоканальную передачу (с помощью номеров портов) и проверку целостности (с помощью контрольных сумм) заголовка и существенных данных. Надежная передача в случае необходимости должна реализовываться пользовательским приложением. Ниже в табл. 1 представлена информация о структуре UDP-пакета.



Рис. 2. Схема работы ПО

Таблица 1

Структура UDP-пакета

Биты	0 - 15	16 - 31
0-31	Порт отправителя (Source port)	Порт получателя (Destination port)
32-63	Длина датаграммы (Length)	Контрольная сумма (Checksum)
64-...	Данные (Data)	

Если рассматривать все уровни стека TCP/IP, то можно сказать, что в представленном в данной статье ПО затронуты: протокол сетевого уровня IP, который, в свою очередь, был разработан для передачи данных из одной (под) сети в другую, а также протокол канального уровня Ethernet, который описывает, каким образом будут передаваться пакеты данных через физический уровень, включая кодирование, среду передачи данных, физические характеристики такой среды и принцип передачи данных, содержит указание того, какой машине или машинам в сети предназначен этот пакет.

На рис. 3 представлен внешний вид разработанного программного обеспечения. С помощью программы управления установлена несущая частота радиосредства. Программа отображает спектр сигнала типового OFDM модема, принимаемый экспериментальной платформой.

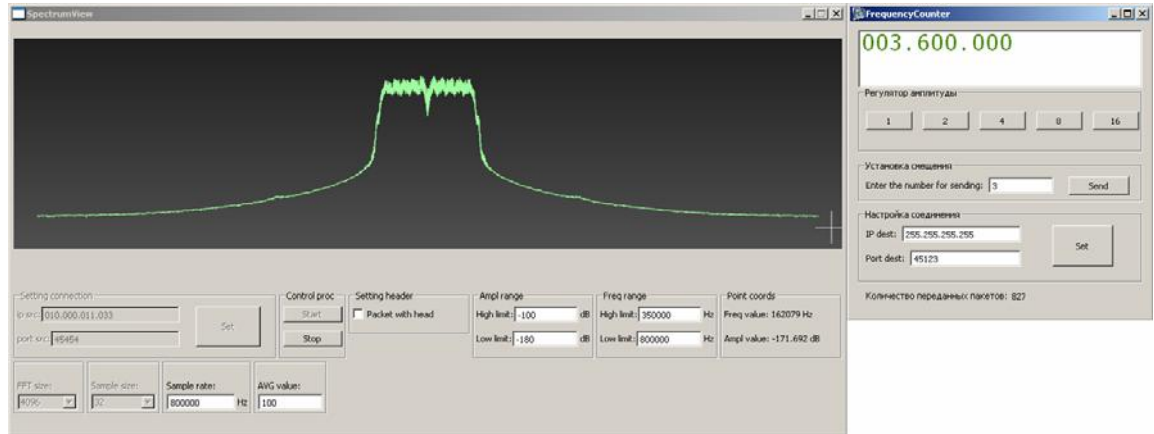


Рис. 3. Внешний вид программного обеспечения

Разработанное авторами программное обеспечение позволяет дистанционно управлять SDR-радиосредством и отображать спектр принимаемого сигнала в реальном времени. Созданное программное обеспечение широко используется магистрами и аспирантами кафедры «Вычислительная техника» Пензенского Государственного Университета при проведении экспериментов в области беспроводных сетей передачи данных и технологии программно-конфигурируемого радио.

Список литературы

1. Sofie, Pollin. Software Defined Radios from Smart(er) to Cognitive / Pollin Sofie, Timmers Michael, Van der Perre Liesbet. – New York : Springer Science+Business Media, 2011. – 140 с.
2. Макс Шлее Qt 4.8. Профессиональное программирование на C++. – СПб. : БХВ-Петербург, 2012. – 912 с.
3. Graham, Sellers, OpenGL SuperBible: Comprehensive Tutorial and Reference / Sellers Graham, S. WrightRichard, Haemel Nicholas. – 6th ed. – New York : Addison-Wesley Professional, 2013. – 848 с.

М. С. Смагин

**Россия, Москва, ЗАО «МНИТИ», Московский государственный
технический университет «МИРЭА»**

ПЕЧАТНАЯ ЭЛЕКТРОНИКА КАК ПЕРСПЕКТИВНОЕ АППАРАТНО-ТЕХНОЛОГИЧЕСКОЕ НАПРАВЛЕНИЕ

Дается базовый обзор современного состояния печатной электроники и пластиковой логики, как перспективных аппаратно-технологических направлений. Дается описание существующих и перспективных применений указанных технологий, приведены краткие описание различных способов производства устройств печатной электроники.

Печатная электроника и пластиковая логика, пока еще, не получили большого распространения в России, и многим специалистам в области традиционной твердотельной электроники и вычислительной техники представляются мало знакомой экзотикой. Однако в настоящее время они вызывают все больший интерес, поэтому основная задача данной статьи, – создать представление о данных научных направлениях, представить современный уровень развития, а также обрисовать основные проблемы и перспективы дальнейшего развития.

Прежде чем перейти к основной части статьи, следует разобраться с объектами рассмотрения, – что есть печатная электроника и что есть пластиковая логика. В настоящее время эти термины настолько неразрывно связаны друг с другом, что часто считаются синонимами, хотя это не совсем верно.

Под пластиковой логикой понимают электронные компоненты или устройства, основой для создания которых являются органические полимеры, играющие роль проводников, полупроводников и подложек. Печатная электроника же представляет собой совокупность печатных методов, используемых для создания электронных приборов. Их неразрывная связь обусловлена тем, что электронные устройства на основе пластиковой логики чаще всего выполняются методами печатной электроники.

Следует отметить, что понятия «пластиковая электроника» и «печатная электроника» близки, но не совпадают по смыслу. Помимо органических полупроводников в печатной электронике также применяются неорганические материалы. Они используются для создания проводников (красители на основе серебра, меди и алюминия) и диэлектриков.

Каковы же основные достоинства устройств печатной электроники? Их два – дешевизна и конструктивная гибкость. Дешевизна обусловлена, в первую очередь, технологической простотой, поскольку производство пластиковых электронных устройств требует гораздо меньшего, по сравнению с кремниевой электроникой, числа технологических операций, выполняется на сравнительно более простом технологическом оборудовании и требует меньше сырья.

Конструктивная гибкость достигается, в первую очередь, благодаря эластичности материалов печатной электроники после нанесения, их устойчивости к изгибам. Это позволяет печатать электронные схемы на эластичных подложках, таких как гибкая пластиковая пленка, бумага, или даже ткань.

Данные особенности обуславливают и основные направления практического использования печатной электроники. Это сравнительно простые и дешевые электронные устройства, изготавливаемые большими сериями и эксплуатирующиеся в условиях постоянных нагрузок на изгиб. Если расположить основные направления применения печатной электроники в порядке массовости и популярности, то он примет следующий вид:

- 1) OLED-устройства;
- 2) Гибкие солнечные батареи;
- 3) Устройства отображения информации;
- 4) Датчики различного назначения;
- 5) Гибкие элементы питания;
- 6) Вычислительные устройства.

OLED-устройства являются наиболее популярным направлением применения печатной электроники. Этому способствует их невысокая цена и высокие потребительские качества. Основными группами OLED-устройств являются устройства отображения информации и устройства освещения. OLED-устройства отображения информации наиболее известны среди неспециалистов в настоящее время, поскольку они уже нашли массовое применение в мобильных устройствах. В данной области наиболее активны такие гиганты рынка, как LG, Panasonic и Samsung.

OLED-устройства освещения вызывают высокий интерес благодаря своей гибкости и высокой энергетической эффективности. Подобные устройства имеют очень широкий спектр возможных применений, например, в качестве светоизлучающих элементов одежды, светоизлучающих покрытий и т.д. Наибольшую активность в этой области проявляют фирмы Du Pont и Siemens.

Гибкие солнечные батареи, изготавливаемые печатным способом, обладают в настоящее время очень низкой энергетической эффективностью, порядка 5–7 %, однако, несмотря на это пользуются большим вниманием. В первую очередь, как и в предыдущем случае, благодаря своей гибкости.

Следующий вид устройств, изготавливаемых методами печатной электроники, это датчики различного назначения. В настоящее время многие виды этих датчиков уже выпускаются серийно и используются в электронных устройствах массового потребления.

Следующий вид электронных устройств, производимых печатными методами, это гибкие элементы питания. Иногда их еще именуют энергетической бумагой

(power paper). Это плоские гибкие устройства, позволяющие сохранять электрический заряд длительное время. В зависимости от особенностей конструкции они могут быть однократного применения или перезаряжаемыми. Основными локомотивами данного направления являются фирмы BlueSpark и Enfucell.

Ну и наконец, последнее, но, пожалуй, наиболее интересное направление применения достижений печатной электроники и пластиковой логики. Основным препятствием в данном направлении является ограничение существующих печатных методов по созданию схем, сопоставимых по уровню миниатюризации и рабочим частотам с кремниевыми микросхемами, изготавливаемыми традиционным способом. Данные ограничения задаются как особенностями используемых методов печати, так и характеристиками красителей, в первую очередь их вязкостью. Согласно данным компании CSEM (Швейцария), наиболее передовые на настоящий момент методы печати позволяют создавать схемы с шириной канала транзистора до 500 нанометров, работающие на тактовых частотах до 500 килогерц с прогнозируемым на ближайшие годы выходом на рубеж в 1 мегагерц. Наибольшую активность в данном направлении предпринимает знаменитый исследовательский центр компании Хегох в Пало Альто, который в настоящее время реорганизован в самостоятельную компанию PARC (Palo Alto Research Company).

Помимо исполнительных вычислительных устройств, активная работа ведется в области создания запоминающих устройств, изготавливаемых печатным способом. Данное направление активно развивается компанией ThinFilm. В настоящее время компания может производить запоминающие устройства емкостью до 100 бит, изготавливаемые печатным способом на практически любом носителе, включая плотную бумагу.

В современной полиграфии насчитывается порядка 20 различных методов печати [1]. Однако далеко не все из них используются для производства печатной электроники. Наиболее широко в настоящее время применяется 5 из них, – трафаретная печать, флексографическая печать, офсетная (литографическая) печать, ротогравюрная печать, струйная печать.

Если свести числовые характеристики рассмотренных технологий в таблицу, то она будет иметь следующий вид [1].

Таблица 1

Метод печати	Толщина печатного слоя (мкм)	Минимальный размер печати (мкм)	Приводка* (мкм)	Скорость печати (м ² /с)
Трафаретный	5–100	20–100	> 25	2–3
Флексографический	0,04–2,5	80	< 200	3–30
Офсетный [3]	0,5–2	10–50	> 10	3–30
Ротогравюрный	0,1–8	75	>20	3–60
Струйный	0,05–20	20–50	5–20	0,01–0,5

*Приводка – совмещение в процессе печатания цветоделенных изображений на оттиске при многокрасочной печати.

Ярким примером применения технологий печатной электроники является оригинальная разработка компании PARC-Research, выполнявшейся совместно с фирмой ThinFilm по заказу Министерства Обороны США и Агентства по перспективным разработкам (DARPA). Основной задачей разработки было создание multifunctional сенсора для бойцов пехоты. Согласно требованиям заказчика, сенсор должен регистрировать параметры близких взрывов (яркость, громкость хлопка, температуру, параметры ударной волны и т.д.), сбор которых необхо-

дим для последующего лечения причиненной взрывом контузии. Предполагается, что сенсор должен состоять из гибкой батареи, изготавливаемой печатным способом, а также набора датчиков различного типа, также изготавливаемых методами печатной электроники. В задачи сенсора входит сбор, хранение (до 7 дней) и выдача собранных данных при считывании.[2]

Другим проектом, выполняемым также в интересах военных, является разработка радиоантенн, которые могут выполняться печатным образом на касках или боевой одежде солдат [5].

Еще одним интересным и перспективным направлением исследований является разработка преобразователей электромагнитного излучения, позволяющих преобразовывать инфракрасное излучение в свет видимого диапазона, и выполненных на основе материалов органической электроники. Подобные разработки, в частности, ведутся в Мичиганском университете, причем их конечной целью является создание приборов ночного видения, выполненных в виде контактных линз [4]

В заключение приведем некоторые экономико-статистические показатели отрасли печатной электроники. Суммарный объем рынка печатной электроники составляет, по данным исследовательской компании IDTechEX, 16 миллиардов долларов [3]. Из них порядка 10 миллиардов приходится на OLED-устройства и еще около 3 – на печатные солнечные батареи. Согласно долгосрочным прогнозам IDTechEX, в 2023 суммарный объем рынка печатной электроники составит около 77 миллиардов долларов, которые будут приблизительно распределены следующим образом: OLED-устройства – 65 миллиардов, солнечные батареи – 5 миллиардов, датчики – 1,15 миллиардов, средства отображения данных – 500 миллионов.

Список литературы

1. Handbook of Print media: Technologies and Production Methods / ed. Helmut Kipphan. – Berlin : Springer-Verlag Berlin Heidelberg, 2001.
2. Printed Electronics: Technologies, Challenges and Applications / Jurgen Daniel // International Workshop on Flexible and Printed Electronics (IWFPE 10).
3. Printing Technologies for Electronic Applications // Printed Electronics Europe and Energy Harvesting & Storage Europe, 2013.
4. New tech could lead to night vision contact lenses / Kate McAlpine. – 2014. – Mar. 16. – URL: <http://www.ns.umich.edu/new/releases/22042-new-tech-could-lead-to-night-vision-contact-lenses>
5. 3D Electronic Printing Holds Promise of Various Applications for Soldiers / Audra Calloway // Picatinny. – 2013. – Dec. 2. – URL: http://www.army.mil/article/116189/3D_electronic_printing_holds_promise_of_various_applications_for_Soldiers

Р. Г. Скопинцев

Россия, Пенза, Пензенский государственный университет

АППАРАТНАЯ РЕАЛИЗАЦИЯ НА ПЛИС АССОЦИАТИВНОЙ ПАМЯТИ С ПОИСКОМ БЛИЖАЙШИХ ПОИСКОВЫХ СЛОВ

Рассматривается вариант реализации ассоциативной памяти для встраиваемых систем на основе ПЛИС, которая позволяет, в случае отсутствия искомого элемента находить ближайшие по значению поисковые слова и значения по этим поисковым словам.

Ассоциативная память (АП) или Ассоциативное запоминающее устройство (АЗУ) является особым видом машинной памяти, используемой в приложениях очень быстрого поиска. Известна также как память, адресуемая по содержимому, ассоциативное запоминающее устройство, контентно-адресуемая память или ассоциативный массив [1]. Данный тип памяти известен давно и широко применяется в Центральных процессорах, сетевых маршрутизаторах, системах ориентированных на работу с базами данных и т.д.

Наиболее широко распространены два типа АП [2]:

- двоичная АП;
- троичная АП.

Двоичная АП – наиболее простой вариант АП, в котором поиск осуществляется по полному совпадению поискового слова, состоящего из единиц и нулей, с словом поискового запроса.

Троичная АП отличается наличием третьего состояния бита («Х» или «Не важно») в процедуре сравнения. В троичной АП может быть сохранено слово «101XX», которое будет совпадать с любым из четырех слов поиска «10100», «10101», «10110», или «10111». Добавление такой возможности усложняет АП за счет использования регистра бит «важности», но делает АП более гибкой.

Предлагаемый вариант ассоциативной памяти с поиском ближайших поисковых слов содержит в своей основе АП двоичного типа с расширенными возможностями.

Основные возможности АП с поиском ближайших поисковых слов:

- получение значения по поисковому слову;
- получение значения и поискового слова ближайшего меньшего от требуемого поискового слова;
- получение значения и поискового слова ближайшего большего от требуемого поискового слова.

Упрощенная структура ячейки АП с поиском ближайших поисковых слов представлена на рис. 1.

Необходимым условием для функционирования АП с поиском ближайших поисковых слов является его упорядоченность по поисковым словам. Данное требование достигается за счет специального аппаратного режима сортировки. Добавление нового элемента в АП происходит следующим способом: на шину поискового слова (CBus) подается значение поискового слова нового элемента, на шину данных (WData) подается значение данных нового элемента, после подается сигнал записи (WR). Если новое поисковое слово больше остальных, то новый элемент добавляется в конец, занимая свободную ячейку. Если новое поисковое слово меньше всех других, то элементы сдвигаются в соседние ячейки (через шины IData, IContent, OData, OContent), освобождая начальную ячейку, в которую производится

запись нового элемента. Если новое поисковое слово меньше только части ячеек, то ячейки с большим поисковым словом сдвигаются, а в освобожденное место записывается новая ячейка.

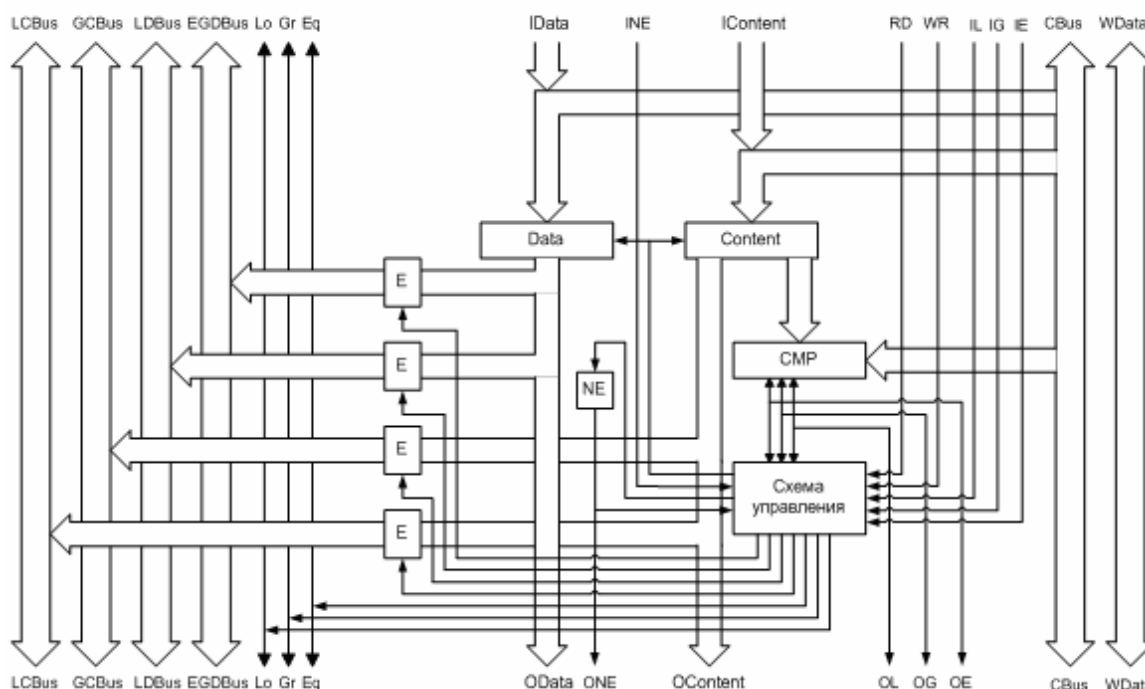


Рис. 1. Упрощенная структура ячейки АП с поиском ближайших поисковых слов

Основной алгоритм работы АП с поиском ближайших поисковых слов начинается с подачи на шину поискового слова (CBus) значения необходимого поискового слова. Каждая ячейка АП выполняет сравнение значения поискового слова, полученного с CBus, со значением поискового слова, хранимом в ячейке. После сравнения формируются признаки: больше (OG), меньше (OL) или равно (OE). Полученные признаки передаются соседним ячейкам (OG предыдущей, OL следующей) для функционирования схемы управления (входы IL, IG, IE). Признак равенства является наиболее приоритетным признаком. Если признак равенства сформирован, то поиск ближайших поисковых слов не требуется, ячейка АП выставляет хранимое значение на совмещенную шину равного или большего значения (EGDBus) и формирует сигнал о готовности значения (Eq). Если нет эквивалентного значения поискового слова в ячейках, то выполняется поиск ближайших. Меньшая и большая ячейка определяется следующими выражениями:

- ячейка является соседней меньшей если: $RD \& NE \& IG \& OL$;
- ячейка является соседней большей если: $RD \& NE \& IL \& OG$.

В рассмотренном выше выражении бит RD является признаком операции чтения из памяти, бит NE является признаком заполненной ячейки. Для первой ячейки IL всегда 1, для последней ячейки IG всегда 1.

Если ячейка является соседней меньшей, то выполняются следующие действия: содержимое регистра Data передается на шину LDBus, содержимое регистра Content передается на шину LCBus, признак готовности передается на линию Lo.

Если ячейка является соседней большей, то выполняются следующие действия: содержимое регистра Data передается на шину EGDBus, содержимое регистра Content передается на шину GCBus, признак готовности передается на линию Gr.

На языке VHDL описание выходов ячейки будет выглядеть следующим образом (для не описанных аналогично):

```

if (NE = '1' and Cont < Content) then
    OL <= '1';
else OL <= '0';
end if;
if ((NE = '1' and RD = '1' and Cont = Content) or
    (NE = '1' and RD = '1' and Cont > Content and IL = '1')) then
    GCContent <= Cont;
else GCContent <= (others=>'Z');
end if;
if(NE = '1' and RD = '1' and Cont = Content) then
    CReady <= '1';
else CReady <= 'Z';
end if;
if (NE = '1' and RD = '1' and Cont > Content and IL = '1') then
    GReady <= '1';
else GReady <= 'Z';
end if;
if (NE = '1' and RD = '1' and Cont < Content and IG = '1') then
    LReady <= '1';
else LReady <= 'Z';
end if;

```

Временные диаграммы работы АП представлены на рис. 2.

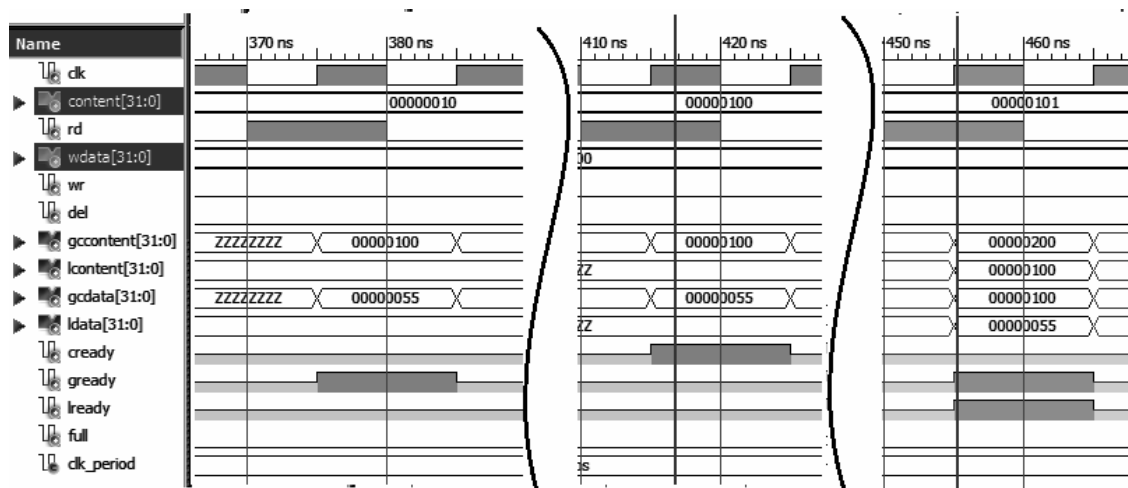


Рис. 2. Временные диаграммы работы АП

На первом фрагменте временной диаграммы показана попытка чтения (сигнал RD) значения по поисковому слову 0x010. В результате выполнения операции найдена большая соседняя ячейка (сигнал GReady) с поисковым словом 0x100 и значением 0x055.

На втором фрагменте временной диаграммы показана попытка чтения значения по поисковому слову 0x100. В результате выполнения операции найдена ячейка с указанным поисковым словом (сигнал CReady) и значением 0x055.

На третьем фрагменте временной диаграммы показана попытка чтения значения по поисковому слову 0x101. В результате выполнения операции найдена меньшая соседняя ячейка (сигнал LReady) с поисковым словом 0x100 и значением 0x055 и большая соседняя ячейка (сигнал GReady) с поисковым словом 0x200 и значением 0x100.

Рассмотренный тип АП может найти применение в математических приложениях, например для интерполяции, реализуемой аппаратно. Данный тип памяти сможет повысить производительность за счет упразднения алгоритмов поиска в памяти.

Список литературы

1. Patent 6,823,434 U.S. System and Method for Resetting and Initializing a Fully Associative Array to a Known State at Power on or through Machine Specific State / Hannum et al. (2004)
2. Pagiamtzis, K. Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey / Kostas Pagiamtzis // IEEE Journal of Solid-state Circuits. – 2006. – March. – Vol. 41, no 3.

А. В. Рыбаков

Россия, Пенза, Пензенский государственный университет

УЛУЧШЕНИЕ ОЦЕНКИ СЛОЖНОСТИ АСИМПТОТИЧЕСКИ ОПТИМАЛЬНЫХ ПО НАДЕЖНОСТИ КЛЕТОЧНЫХ СХЕМ

Предложено улучшение ранее известной оценки сложности для асимптотически оптимальных по надежности клеточных схем.

Введение и постановка задачи

Впервые задачу синтеза надежных схем, реализующих булевы функции и состоящих из ненадежных функциональных элементов (ФЭ), рассматривал Дж. фон Нейман [1]. Он предполагал, что все элементы схемы независимо друг от друга с вероятностью ε ($\varepsilon \in (0; 1/2)$) подвержены инверсным неисправностям на выходах. Эти неисправности характеризуются тем, что в исправном состоянии функциональный элемент реализует приписанную ему булеву функцию ϕ , а в неисправном – функцию $\bar{\phi}$. С помощью итерационного метода Дж. фон Нейман установил, что в произвольном полном базисе при $\varepsilon \in (0; 1/6)$ любую булеву функцию можно реализовать схемой, вероятность ошибки, на выходе которой при любом входном наборе значений переменных не превосходит $c_1\varepsilon$ (c_1 – некоторая константа, зависящая от базиса). Затем схемы с инверсными неисправностями на выходах элементов исследовались в работах С. И. Ортюкова [2], Д. Улига [3] и некоторых других авторов, причем главное внимание уделялось сложности надежных схем. В [2, 3] были построены асимптотически оптимальные по сложности схемы, функционирующие с некоторым уровнем надежности (задача синтеза оптимальных по надежности схем до появления работ М.А. Алехиной не ставилась).

Асимптотически оптимальные по надежности схемы, реализующие булевы функции, в базисе $\{x \& y, x \vee y, \bar{x}\}$ при инверсных неисправностях на выходах

элементов построены А.В. Васиным [4], а в работе М. А. Алехиной и С. И. Аксенова [5] доказано, что сложность таких схем превышает сложность схем, построенных из абсолютно надежных элементов, асимптотически не более чем в 3 раза.

В этой статье рассматривается реализация булевых функций клеточными схемами (КС) (еще их называют плоскими схемами), содержащими как надежные, так и ненадежные элементы, и оценивается их ненадежность и сложность. Определение КС можно найти, например, в работе С. С. Кравцова [6]. В работе А. Ю. Улесовой [7] получены оценки сложности КС в предположении, что все элементы схемы абсолютно надежны. Введем необходимые понятия и определения для случая, когда некоторые из базисных элементов ненадежны.

Как и в [6] предполагается, что базис содержит два типа элементов: функциональные (рис. 1) и коммутационные (рис. 2). Каждый из этих элементов может быть повернут на плоскости на угол $\frac{k\pi}{2}$ ($k = 0, 1, 2, 3$).

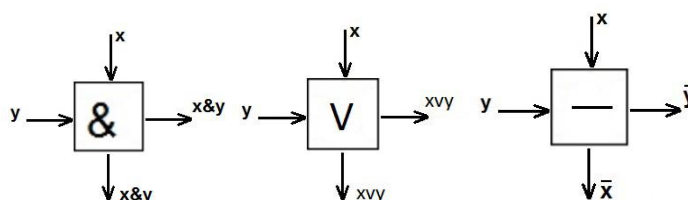


Рис. 1. Функциональные элементы

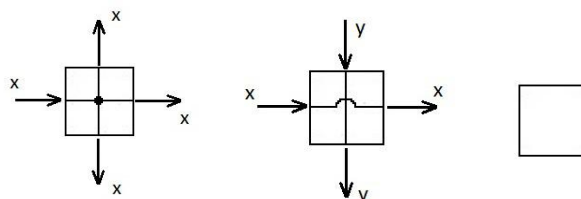


Рис. 2. Коммутационные элементы

Предполагается, что коммутационные элементы абсолютно надежны, а на любом из двух выходов каждого из функциональных элементов с вероятностью ε ($\varepsilon \in (0; 1/2)$) независимым образом появляются инверсные неисправности.

Считаем, что КС, содержащая ненадежные элементы, реализует булеву функцию $f(\tilde{x}^n)$ ($\tilde{x}^n = (x_1, \dots, x_n)$), если она реализует $f(\tilde{x}^n)$ при отсутствии неисправностей.

Пусть КС S реализует функцию $f(\tilde{x}^n)$. Обозначим через $P_{f(\tilde{a}^n)}(S, \tilde{a}^n)$ вероятность появления ошибки на входном наборе $\tilde{a}^n$ схемы S . *Ненадежность* $P(S)$ клеточной схемы S определяется как максимальная вероятность ошибки на выходе схемы при всевозможных входных наборах схемы (т.е. так же, как и для схемы из функциональных элементов). *Надежность* схемы S равна $1 - P(S)$.

Пусть $P_\varepsilon(f) = \inf_S P(S)$, где инфимум берется по всем схемам S , реализующим функцию f . Клеточная схема A , реализующая функцию f , называется *асимптотически оптимальной по надежности*, если $P(A) \sim P_\varepsilon(f)$ при $\varepsilon \rightarrow 0$.

Сложность КС, которая, напомним, имеет вид прямоугольника, определяется как площадь этого прямоугольника. Определим две функции Шеннона для площа-

ди схем этой модели. Обозначим высоту клеточной схемы S через $h(S)$, длину через $\lambda(S)$ и будем считать, что $h(S) \leq \lambda(S)$.

Пусть f – произвольная булева функция. Обозначим через $L(f)$ наименьшую из площадей $L(S) = h(S)\lambda(S)$, где минимум берется по всем КС S , реализующим функцию f , а через $L(n)$ – функцию Шеннона, которая равна $L(n) = \max L(f)$, где максимум берется по всем функциям f от n переменных.

Аналогично определяется функция Шеннона $L^h(n)$ для случая, когда высота h клеточных схем фиксирована.

Для клеточных схем из абсолютно надежных элементов в работе [7] получена асимптотика функции Шеннона (в произвольном базисе при $h \geq 4$, а в рассматриваемом базисе при $h \geq 3$), которая имеет вид $L^h(n) \sim \frac{h2^n}{\log n}$.

В работе [8] доказана оценка сложности для асимптотически оптимальных по надежности КС, которая сформулирована в Теореме 1.

Теорема 1 [8]. Любую булеву функцию $f(x_1, \dots, x_n)$ можно реализовать такой клеточной схемой S , что $P(S) \leq 3\varepsilon + 240\varepsilon^2$, $L(S) \leq 3,1n6^n$ при всех $\varepsilon \in (0, 1/1000]$.

В данной работе оценку сложности удалось улучшить. Сформулируем полученные результаты.

Теорема 2. Любую булеву функцию $f(x_1, \dots, x_n)$ можно реализовать такой клеточной схемой S , что $P(S) \leq 3\varepsilon + 416\varepsilon^2$, $L(S) \leq 12 \cdot 4^n$ при всех $\varepsilon \in (0, 1/1000]$.

Теорема 3. Любую булеву функцию $f(x_1, \dots, x_n)$ можно реализовать такой клеточной схемой S , что $P(S) \leq 3\varepsilon + 48\varepsilon^2$, $L(S) \leq 36 \cdot 4^n$ при всех $\varepsilon \in (0, 1/1000]$.

Вывод: Полученная оценка сложности для асимптотически оптимальных по надежности схем была улучшена, однако она по-прежнему уступает оценке сложности для клеточных схем, построенных из абсолютно надежных элементов.

Исследование выполнено при финансовой поддержке РФФИ, номер проекта 14-01-00273.

Список литературы

1. Neuman, J. von Probabilistic logics and the Synthesis of reliable organisms from unreliable components // Automata studies / ed. by Shannon C., Mc. Carthy J. – Princeton University Press, 1956 (Русский перевод: Автоматы. – М. : ИЛ, 1956. – С. 68–139.)
2. Ортюков, С. И. Об избыточности реализации булевых функций схемами из ненадежных элементов / С. И. Ортюков // Труды семинара по дискретной математике и ее приложениям (Москва, 27 – 29 января 1987 г.). – М. : Изд-во Моск. ун-та, 1989. С. 166–168.
3. Uhlig, D. Reliable networks from unreliable gates with almost minimal complexity / D. Uhlig // Fundamentals of Computation Theory. Intern. conf. FCT'87 (Kazan, June 1987). – Berlin : Springer-Verl., 1987. – P. 462–469 (Lecture Notes in Comput. Sci.; V. 278) (Русский перевод: Автоматы. – М. : ИЛ, 1956. – С. 68–139).
4. Васин, А. В. Об асимптотически оптимальных схемах в базисе $\{x \& y, x \vee y, \bar{x}\}$ при инверсных неисправностях на выходах элементов / А. В. Васин // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2008. – № 4. – С. 3–17.
5. Алехина, М. А. О сложности надежных схем при инверсных неисправностях / М. А. Алехина, С. И. Аксенов // Материалы IX Международного семинара «Дискретная

математика и ее приложения», посвященного 75-летию со дня рождения О. Б. Лупанова (Москва, 18–23 июня 2007 г.). – М. : Изд-во мех.-мат. фак-та МГУ, 2007. – С. 56–59.

6. Кравцов, С. С. О реализации функций алгебры логики в одном классе схем из функциональных и коммутационных элементов / С. С. Кравцов // Проблемы кибернетики. – М. : Наука, 1967. – Вып. 19. – С. 285–292.

7. Улесова, А. Ю. Сложность реализации булевых функций в некоторых моделях клеточных схем : дипломная работа / А. Ю. Улесова. – М. : МГУ им. Ломоносова, факультет ВМиК, кафедра математической кибернетики, 2010. – 25 с.

8. Алехина, М. А. Синтез и сложность асимптотически оптимальных по надежности клеточных схем / М. А. Алехина, А. В. Рыбаков // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2014. – № 4.

Н. И. Пикулева

Россия, г. Казань, Казанский национальный исследовательский технический университет им. А. Н. Туполева

ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В CASE-СРЕДСТВЕ SILVERRUN

Рассматриваются вопросы моделирования информационно-вычислительных систем с помощью case-средства Silverrun. Это средство реализует технологию проектирования программного обеспечения вычислительных систем и используется для построения моделей бизнес процессов и структур данных с последующей генерацией программного кода моделируемой системы и базы данных.

Для создания сложного программного продукта проектировщикам требуется средство для анализа и формирования требований, разработки, тестирования и документирования будущей вычислительной системы, а также для обеспечения качества выполнения проекта и управления проектом. При проектировании сложных вычислительных систем используются два разных подхода к проектированию: объектно-ориентированный и структурный. В случае структурного подхода будущая система представляется в виде одной большой функции и далее проводится декомпозиция этой функции на подфункции до тех пор, пока простые функции не смогут быть представлены простым алгоритмом, который записывается в 20–30 строк программного кода. Одновременно с разработкой функциональности системы проводится моделирование информационной модели данных, используемой системой. Современный рынок case-средств предлагает для моделирования структурным методом большое количество программных средств, наиболее популярными из которых являются Erwin, BPwin, Silverrun.

Для автоматизации моделирования программного обеспечения хорошо подходит case-средство Silverrun. Эта система ориентирована на спиральную модель ЖЦ и может применяться для поддержки любой методологии, которая основана на разделении моделирования функциональной и информационной модели. Информационная модель представляет статическую структуру системы и используется в дальнейшем для построения базы данных, а функциональная модель отражает поведение будущей системы. В Case-средство Silverrun присутствует возможность настройки работы со многими известными методологиями, но наиболее широко ис-

пользуемой является методология Datarun. Любая методология обеспечивает концептуальную основу для проектирования и предлагает технологию руководства проектом. При использовании case-средства Silverrun будущие программы представляются схематическими формами, которые проще и понятнее в использовании, а именно в виде наглядных диаграмм различных типов. Прежде всего это диаграммы функционального проектирования (DFD – диаграммы потоков данных) и диаграммы моделирования данных (различные модификации диаграммы «сущность-связь»). Построенные диаграммы позволяют выявить все требования к будущей системе, проанализировать функции программного обеспечения и обеспечить стандартное представление структуры программы и используемых данных. Case-средство Silverrun состоит из трех модулей, которые могут использоваться как совместно, так и отдельно друг от друга.

Первый модуль case-средства Silverrun – это модуль BPM (Business Process Modeler), который используется для построения диаграмм потоков данных (DFD). Он позволяет поэтапно создавать модели бизнес процесса, начиная с контекстной диаграммы создания внешних сущностей, основных процессов и потоков данных, которыми обмениваются процессы и внешние сущности между собой. Далее, каждый процесс детализируется на диаграммах нижнего уровня, при этом должно обязательно соблюдаться правило балансировки, это означает, что при детализации процесса детализирующая диаграмма в качестве входных или выходных потоков данных может иметь только те потоки данных, с которыми имеет информационную связь детализируемый процесс на родительской диаграмме [1, с. 250]. Одновременно с моделированием процессов, в диаграммах необходимо начинать разрабатывать структуры данных, уделяя, однако, основное внимание бизнес процессам. При построении иерархии диаграмм бизнес модели переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных. Содержание потоков и накопителей данных описывается с помощью структур данных. Для каждого потока данных формируется список всех его элементов, затем элементы объединяются в структуры данных, которые соответствуют более крупным объектам данных. Каждый объект должен состоять из элементов, которые являются его атрибутами.

При построении законченной модели системы ее необходимо верифицировать, т.е. проверить на полноту и согласованность. В согласованной модели для всех потоков данных и накопителей данных должно выполняться правило сохранения информации: все поступающие в систему данные должны быть считаны, а все считанные данные должны быть записаны.

Модуль Silverrun BPM может использоваться как автономный инструмент моделирования бизнес процессов, так и в связке с другими модулями Silverrun.

Второй модуль Silverrun – это модуль построения концептуальной модели данных будущей системы, модуль ERX. Он позволяет построить модель данных, которая пока не привязана к конкретной реализации. Модель типа «сущность-связь» может создаваться в модуле с «нуля», однако возможна передача структуры данных, разработанной ранее в модуле BPM.

Цель моделирования данных состоит в обеспечении разработчика ПО концептуальной схемой базы данных в виде одной или нескольких локальных моделей. С помощью диаграммы DFD, созданной в модуле BPM разрабатываются накопители данных, а с помощью модуля ERD осуществляется детализация накопителей данных, а также документируются информационные части бизнес модели. В информационную часть входит полное описание объектов, важных для разрабатываемой системы (сущностей), свойств этих объектов (атрибутов) и их связей с другими объектами (отношений).

В модели ERX имеется встроенная экспертная система, которая помогает построить целостную и непротиворечивую нормализованную модель данных. Построенная модель «сущность-связь» может быть передана в третий модуль Silverrun.

Третий модуль Silverrun – это модуль реляционного моделирования RDM (Relational Data Modeler). Этот модуль при моделировании программного обеспечения может использоваться самостоятельно для разработки будущей базы данных реляционного типа. Однако рекомендуется детализировать в нем ранее созданные модели в модуле ERX и переданные через общий для case-средства Silverrun репозиторий в модуль RDM. На данном этапе создаются детализированные ERD модели, при этом из модели исключаются связи типа «многие-ко-многим», создаются первичные и внешние ключи, индексы, домены, триггеры ссылочной целостности и хранимые процедуры.

В результате разработки получают реляционную модель данных, из которой генерируется код базы данных. Генерация схемы базы данных происходит путем создания файла с SQL-операторами, в которые переводятся конструкции модели данных. Этот файл используется для создания физической базы данных на сервере.

Для того, чтобы обеспечить интеграцию всех модулей Silverrun в единую среду проектирования используется общий в case-средстве Silverrun словарь или репозиторий, который является основой case-средства.

В репозитории хранятся версии проекта и его отдельных компонентов, синхронизируется поступление информации от различных разработчиков, контролируются данные на полноту и непротиворечивость. Содержимое репозитория включает все элементы модели, объекты разных типов, отношения между элементами модели, правила использования или обработки этих элементов. В репозитории хранится подробное описание каждого объекта: идентификатор объекта, имена синонимы, тип, свойства, текстовое описание, область значений. Кроме этого, хранятся все отношения с другими объектами, правила формирования и редактирования объекта, времени его последнего обновления и т. д. Каждый модуль Silverrun использует один и тот же репозиторий, благодаря этому разработанные в одном модуле модели можно использовать в другом модуле. Репозиторий является базой для стандартизации документации по проекту. Все отчеты строятся автоматически по содержанию репозитория. Важные функции по управлению проектом и контролю за разработкой проекта тоже реализуются на основе репозитория, с помощью репозитория можно осуществлять контроль безопасности (ограничение доступа, привилегии доступа) и т.д. [2, с.74].

Основные функции поддержки репозитория – хранение, доступ, обновление, анализ и визуализация всей информации по проектируемой модели программного обеспечения вычислительных систем.

В заключении хотелось бы отметить эффективность использования case-средства Silverrun для моделирования программного обеспечения и баз данных, удобство изучения, использования и понятный интерфейс case-средства Silverrun.

Список литературы

1. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем / А. М. Вендров. – М. : Финансы и статистика, 2006. – 544 с.
2. Орлов, С. А. Технологии разработки программного обеспечения / С. А. Орлов. – СПб. : Питер, 2012. – 608 с.

ДАТЧИКИ СИСТЕМ КОНТРОЛЯ И УПРАВЛЕНИЯ

Рассмотрены основные виды выходных сигналов датчиков и проблемы организации баз данных по датчикам при автоматизации проектирования систем управления.

Условием успешной разработки систем контроля, управления и информационно-измерительных систем является наличие баз данных о технических средствах этих систем. Основные технические средства, входящие в состав систем управления представлены на рис. 1.

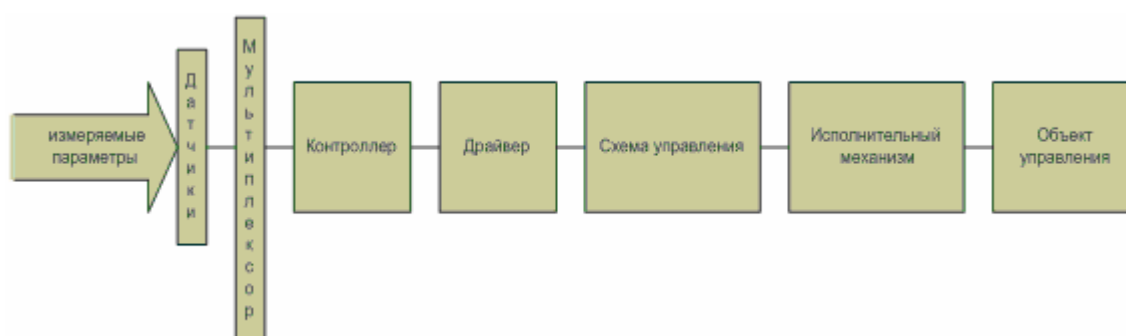


Рис. 1. Структура системы управления

Для управления объектом необходимо создать систему, способную приводить в действие регулирующие органы, которые перемещаются исполнительными механизмами. Исполнительные механизмы являются электромеханическими устройствами для управления которыми служат схемы управления. Эти технические средства входят в силовую часть системы и должны управляться мощными сигналами, которые создаются драйверами, обеспечивающими помимо усиления информационных сигналов по мощности еще гальваническую развязку силовой части системы от маломощной, информационной части. Контроллер выполняет функции сравнения сигналов, поступающих с датчиков с заданными уставками, и реализации необходимого закона управления, формируя информационные сигналы для управления силовой частью системы. Датчики в системе управления служат для измерения тех параметров объекта управления, исполнительного механизма или схемы управления, по которым осуществляется управление системой. Для ввода сигналов датчиков в контроллер используется аналоговый или цифровой мультиплексор.

В системах управления сложными технологическими процессами или даже большим предприятием действует множество локальных систем управления, объединенных в единую систему управления. Все технические средства, входящие в такую систему управления делятся на несколько уровней. Датчики и исполнительные механизмы относятся к нижнему уровню, следующий уровень составляют контроллеры и удаленные станции, образуя сеть для мониторинга и управления в реальном времени, более высокий уровень образован системами управления процессом - супервизорное управление.

. Заметим, что из всей номенклатуры технических средств автоматизации и управления датчики различных физических величин составляют около 30 %. Датчики в системе управления выполняют важную роль элементов обратной связи,

причем такая важная характеристика системы управления, как точность во многом определяется тем, насколько точно определяется информация о состоянии объекта.

Как и любое техническое средство датчик имеет своего производителя и потребителя. Для разработчика и производителя датчика пожалуй основным его признаком является физический эффект, используемый для его создания, а для потребителя самым значимым признаком выступает его назначение. Это нашло свое отражение в названии датчиков. Например, тензометрический датчик давления, магнитоупругий датчик силы, индуктивный датчик толщины покрытия. Однако, множество измеряемых физических величин, используемых для измерения физических эффектов, способов проведения измерений, конструктивных особенностей привело к тому, что существуют такие наименования датчиков как термопара, энкодер, буйковый уровнемер, турбинный расходомер и т.д.

В связи с этим определяющим признаком для отнесения датчика к той или иной группе все-таки является количественная характеристика, такая как диапазон измеряемых величин, а не наименование датчика.

Статистика применения датчиков по виду контролируемых параметров выглядит следующим образом:

- температура 50 %;
- объемный и массовый расход 15 %;
- давление 10 %;
- уровень жидкости 5 %;
- масса, объем 5 %;
- время 4 %;
- электрические и магнитные величины < 5 %.

Выходным сигналом датчика, работающего в системе, является электрический сигнал, генераторные датчики способны сразу выдавать такой сигнал, а параметрические датчики преобразуют измеряемый параметр в изменение первичных параметров электрической цепи: сопротивлений, емкостей, индуктивностей. Первичные параметры являются элементами измерительной схемы, которая нуждается в подведении питающего напряжения.

Для работы в системе управления или измерения выходной сигнал датчика нормируется. Такую функцию выполняет нормирующий преобразователь. Унифицируются сигналы датчиков по виду и уровню. Основными видами унифицированных сигналов датчиков являются ток, напряжение и частота (рис. 2).

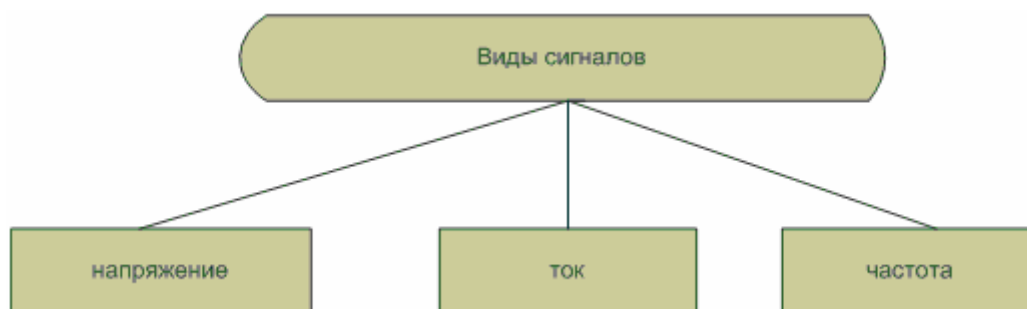


Рис. 2. Виды выходных сигналов датчиков

При этом уровни сигналов составляют

- по току 0...5, 0...20, 4...20 мА;
- по напряжению 0...10, 0...20 мВ, 0...1, 0...10 В;
- по частоте 2...4, 4...8 кГц.

Конструктивно датчики могут быть объединены с нормирующими преобразователями, но иногда по условиям применения датчик не может быть объединен с нормирующим устройством, например, когда электроника не может работать в агрессивной среде или при очень низких или высоких температурах.

Разнообразие видов сигналов датчиков обусловлено тем, что датчики с выходным напряжением удобны в использовании, когда они располагаются территориально вблизи с устройствами более высокого уровня и линия связи невелика, например в автомобиле. Если датчики территориально удалены, то используются два других вида сигналов, ток и частота. Нормирующие преобразователи с выходным сигналом в виде тока используются на расстояниях до 10 км, в этом случае не влияют на точность измерения токи утечки между проводами линии связи. Переход на другой вид выходного сигнала – частоту может быть продиктован большим количеством датчиков даже в рамках одного большого предприятия, когда количество проводов становится недопустимо большим и приходится переходить на беспроводные технологии с использованием передатчиков и приемников. Частотные сигналы используют также для передачи по имеющимся линиям связи или линиям электропередач.

В современных системах с цифровыми контроллерами возникает необходимость преобразования аналогового сигнала датчика в цифровой. Напряжение преобразуется в цифру с помощью аналого-цифрового преобразователя (АЦП). Точковый сигнал, прошедший по линии связи легко преобразуется в стандартное входное напряжение АЦП с помощью нагрузочного резистора, а частота преобразуется в цифровой код путем подсчета количества импульсов частотомером. АЦП может входить в состав нормирующего преобразователя, или выполняться в виде отдельного узла. Если в контроллере имеются аналоговые входы, то предварительного преобразования в цифру не требуется.

В любом случае информация с датчиков в контроллер вводится через цифровой или аналоговый мультиплексор.

Рассмотрев особенности включения датчиков в систему управления, основные виды стандартных выходных сигналов датчиков и, отметив большое разнообразие измеряемых физических величин и датчиков для их измерения, можно констатировать, что создание баз данных по датчикам явится важным шагом в автоматизации проектирования систем.

Работа выполнена в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития разработки научно-технологического комплекса России на 2014–2020» (соглашение № 14.574.21.0045 от 06.19.2014).

Список литературы

1. Осадчий, Е. П. К вопросу организации периферийных средств сбора информации / Е. П. Осадчий, В. В. Пашенко // Приборы и системы управления. – 1995. – № 1.
2. Беспроводные технологии и их применение в промышленности / Г. Гайкович, П. Чен, Х. Каган, П. Фур // Единая скоростная среда для систем промышленной автоматизации. Электронные компоненты. – 2012. – № 2.
3. Пашенко, В. В. Цифровое фазовое управление асинхронным электроприводом / В. В. Пашенко, Э. К. Конюкаев // Проблемы автоматизации и управления в технических системах : тр. Междунар. науч.-техн. конф. – Пенза : Информационно-издательский центр ПГУ, 2009.

Д. А. Трокоз, К. С. Максимова, Е. А. Бальзанникова

Россия, Пенза, Пензенский государственный университет

СПОСОБ ТРАНСЛЯЦИИ ПОСЛЕДОВАТЕЛЬНЫХ ФУНКЦИОНАЛЬНЫХ СХЕМ В ДИАГРАММЫ ФУНКЦИОНАЛЬНЫХ БЛОКОВ

Рассматриваются особенности многоуровневой трансляции программ, написанных на графических языках программирования логических контроллеров. Выдвигается способ преобразования последовательных функциональных схем в диаграммы функциональных блоков.

В настоящее время все больше и больше управляющих устройств используют микроконтроллеры. Широкий выбор вариантов их конфигураций дает разработчикам возможность проектировать готовые системные решения для любых приложений. На сегодняшний день пользуются популярностью 8-битные микроконтроллеры PIC и AVR, 16-битные MSP430, а также 32-битные микроконтроллеры архитектуры ARM. Следует отметить развитие отечественного производства в данной сфере. «ПКК Миландр» – российская компания-разработчик и производитель микроэлектронной элементной базы, ориентированной на применение в изделиях с повышенными требованиями к надежности (авиакосмическая техника, спецтехника). Использует микропроцессорное ядро ARM, на базе которого был разработан российский микроконтроллер, предназначенный для систем учета электроэнергии, энергосберегающих систем и систем с автономным электропитанием, микроконтроллер, ориентированный на применение в специальной технике и системах с повышенной надежностью, а также более 200 типономиналов изделий. Таким образом, возникает необходимость в поддержке развития отечественного производителя.

Программирование микроконтроллеров обычно осуществляется на языках Assembler, C, Forth, Basic. В то же время существует признанный стандарт языков программирования МЭК 61131-3, предоставляющий технологу возможность максимально удобно, качественно и в короткие сроки описывать работу систем автоматического управления. Системы программирования, основанные на данном стандарте, характеризуются надежностью создаваемого программного обеспечения, возможностью простой модификации программы, переносимостью проекта с одного ПЛК на другой, простотой языка и ограничением количества его элементов. В линейке языков МЭК 61131-3 особое место заняли языки графической разработки программ – SFC и FBD.

Язык функциональных блоков (FBD) является наиболее распространенным языком программирования ПЛК благодаря своей наглядности и удобству, возможности повторного использования функциональных диаграмм, наличию библиотек функциональных блоков, а также простой внутренней структуре команд, которая транслируется в быстрый и надежный код. Вместе с тем, нельзя не отметить тот факт, что на языке функциональных блоков неудобным и неочевидным образом реализуются те участки программы, которые было бы удобно представить в виде конечного автомата. Одним из способов решения данной проблемы является совместное использование языков FBD и SFC, при котором последовательность выполнения действий системой управления описывается на языке SFC, в то время как жесткая логика каждого из существующих действий представляется с помощью функциональных блоков.

Целью нашего исследования является разработка кроссплатформенного транслятора программ, написанных на языках SFC и FBD, в параллельно выполняющиеся программы на языке C.

Трансляция включает в себя следующие этапы:

1) преобразование программы, описанной с помощью совместного использования языков SFC и FBD, в аналогичную программу на языке функциональных блоков. Выбор данного языка был сделан на основе имеющихся преимуществ, описанных выше, а так же благодаря возможности максимально точной дальнейшей конвертации;

2) на данном этапе продолжается преобразование полученной программы в язык «Т», в ходе которого происходит распараллеливание вычислений, представленных в формате FBD;

3) заключительный этап трансляции, результатом которой является комплекс параллельно работающих программ, написанных на языке С.

В данной статье более подробно рассматривается первый этап разработки, а именно формирование единого кода программы на языке функциональных блоков.

Язык FBD – графический язык, предназначенный для описания процессов прохождения сигналов через различные объекты. Является наиболее простым и удобным языком для технологов, не имеющих опыта программирования, поскольку структура языка имеет сходство с электрическими схемами.

Язык SFC – графический язык описания последовательного управления процессами, предназначен для описания систем на самом верхнем уровне абстракции. Структуру языка составляют такие элементы, как: состояния, переходы, действия, альтернативные и параллельные ветви, остановка процесса, переход к заданному состоянию. Действия могут быть реализованы с помощью вложенных последовательных функциональных схем или на отличном от SFC языке МЭК.

Таким образом, разрабатываемый алгоритм трансляции должен предоставлять возможность преобразования всех основных элементов языка SFC в диаграммы функциональных блоков, обеспечивать сохранение текущего состояния при отсутствии дальнейших переходов, а так же использовать элементы, не имеющие внутренней памяти.

Предложенная схема перехода системы в состояние State в общем случае показана на рис. 1, где State – переменная состояния, Condition1...ConditionN – условия выхода из интересующего состояния, PreviousState – имя предшествующего состояния, PreviousCondition – условие, по которому система переходит в необходимое состояние.

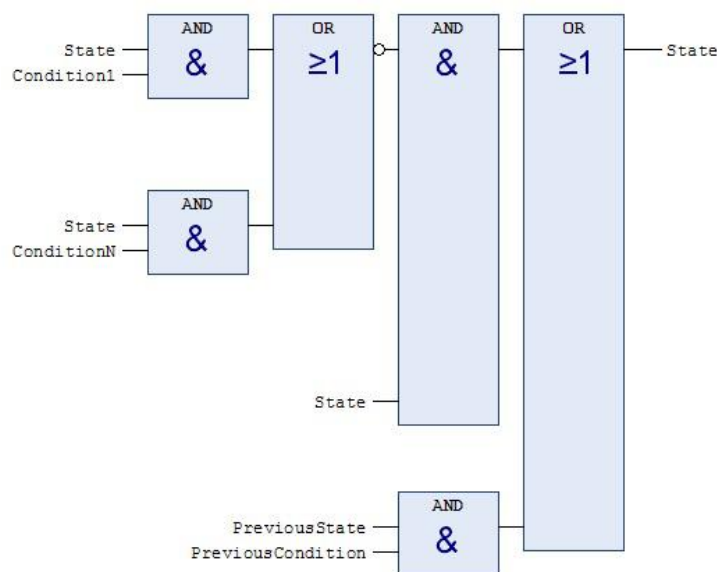


Рис. 1. Схема перехода в состояние State

В полученной схеме реализован алгоритм стандартного SFC перехода на языке FBD. Использованный фрагмент программы на языке SFC представлен на рис. 2.

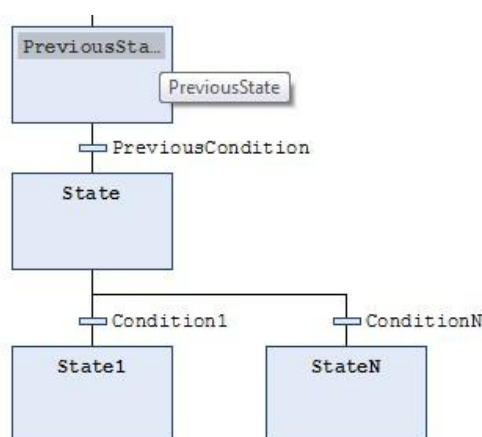


Рис. 2. Фрагмент программы на языке SFC

Работа реализованной подобным образом системы описывается следующими правилами:

4) переход в необходимое состояние выполняется в том случае, если было выполнено условие перехода из предшествующего состояния. Эта часть реализована конъюнкцией переменных PreviousState и PreviousCondition;

5) в том случае, если переход возможен из нескольких различных состояний, либо по нескольким условиям, справедливо выражение

$$(\text{PreviousState } 1 \wedge \text{PreviousCondition } 1) \vee (\text{PreviousState } 2 \wedge$$

$$\text{PreviousCondition } 2) \vee \dots \vee (\text{PreviousState } N \wedge \text{PreviousCondition } N),$$

которое может быть также реализовано на стандартных элементах FBD;

6) для случая, когда система уже находится в состоянии State, и условия дальнейших переходов не выполняются, предусмотрено сохранение состояния. Проверка выполнения выходных условий реализована с помощью дизъюнкции конъюнкций значений переменных условий и переменной, отображающей наличие системы в состоянии State. Инверсия данного значения подается на вход конъюнктора, назначением которого является проверка двух условий: нахождения системы в состоянии State и отсутствие всех возможных выходных переходов. Сохранение состояния происходит при условии получения положительных результатов обеих проверок;

7) выполнение действий, предписанных тем или иным состоянием в языке SFC, осуществляется с помощью блока тернарной операции, на условный вход которого подается переменная необходимого состояния;

8) преобразование как альтернативных, так и параллельных ветвей SFC осуществляется аналогично представленному на рис. 1 примеру, с учетом имеющихся особенностей представления.

Таким образом, предложенный алгоритм преобразования соответствует заданным требованиям и предоставляет возможность преобразования всех основных элементов языка SFC в диаграммы функциональных блоков, а также обеспечивает сохранение текущего состояния при отсутствии дальнейших переходов и позволяет

использовать элементы, не имеющие внутренней памяти. На следующем этапе исследования на его основе планируется реализация кроссплатформенного транслятора, который использует в качестве входных данных и генерирует в результате своей работы код программы стандарта XML формата PLCOpen.

Работа выполнена в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития разработки научно-технологического комплекса России на 2014-2020» (соглашение № 14.574.21.0045 от 06.19.2014).

Список литературы

1. Деменков, Н. П. Языки программирования промышленных контроллеров : учеб. пособие / Н. П. Деменков ; под ред. К. А. Пупкова. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2004.
2. Пащенко, Д. В. Аппаратно-программный комплекс для анализа результатов документирования радиолокационной информации / Д. В. Пащенко, Д. А. Трокоз, М. П. Синев // Вопросы радиоэлектроники. – 2012. – Т. 3, № 1. – С. 144–152.
3. Пащенко, Д. В. Стратегия параллельной обработки массивов данных в системе объективного контроля радиотехнического комплекса радиолокационного дозора и наведения / Д. В. Пащенко, В. А. Мачалин, Д. А. Трокоз, М. П. Синев, А. Н. Токарев // Радиотехника. – 2010. – № 8. – С. 51–55.
4. Петров, И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования / И. В. Петров ; под ред. проф. В. П. Дьяконова. – М. : СОЛОН-Пресс, 2004.

Е. А. Бальзанникова, Д. А. Трокоз, Д. В. Пащенко

Россия, Пенза, Пензенский государственный университет

АЛГОРИТМ ТРАНСЛЯЦИИ ИЗ ФУНКЦИОНАЛЬНОГО ЯЗЫКА «Т» В ПРОГРАММНЫЕ МОДУЛИ НА ЯЗЫКЕ «С»

Рассмотрены основные принципы преобразования алгоритма для программируемых логических контроллеров из языка функциональных блоков в программные модули на языке «С» посредством промежуточного языка «Т».

Автоматизированные системы управления на данный момент широко используются для управления различными процессами в рамках технологического процесса, производства, предприятия. Однако по мере развития промышленного производства, роста разнообразия производимой продукции, усложнения технологических процессов и процедур управления ресурсами одного программируемого логического контроллера (ПЛК) оказывается недостаточно. В связи с этим возникает необходимость проектирования распределенной системы управления, которая позволяет, спроектировав общий алгоритм управления, разделить его на несколько связанных между собой ПЛК, повысив тем самым эффективность системы.

Современные языки программирования ПЛК не позволяют разделить программу на несколько параллельно функционирующих блоков, поэтому было принято решение о разработке собственной системы проектирования, основанной на су-

ществующих языках программирования SFC и FBD с введением определенных ограничений.

Программа, разработанная на вышеуказанных языках, не подходит для непосредственной трансляции в язык программирования микроконтроллеров. Поэтому целесообразно ввести промежуточный язык, который удовлетворяет требованиям, позволяющим разбить алгоритм на несколько независимых подпрограмм:

- отсутствие обратных связей;
- отсутствие циклов, как частного случая обратной связи;
- отсутствие переменных как ячеек памяти.

Назовем данный язык – «язык Т». Язык Т является развитием языка FBD, но содержащий указанные ранее ограничения. Основной структурной единицей данного языка является функциональный блок, имеющий множество входов, один выход и выполняющий определенную функцию. Функции разделяются на базовые и производные (пользовательские). К базовым функциям относятся арифметические и логические операции, операции сравнения, тернарный оператор выбора (который в зависимости от результата логического выражения возвращает один из аргументов) и вызов функции, который позволяет организовать рекурсию или повысить уровень абстракции.

Отсутствие циклов в данном языке компенсируется наличием оператора вызова функции, поскольку любой цикл можно представить в виде рекурсии. Пользовательские функции задаются как совокупность функциональных блоков, в том числе, определенных в основной схеме. Пример использования рекурсии в качестве альтернативы циклам показан на рис. 1 на примере итерационной суммы вида

$$Sum = c + \sum_{k=0}^n (a * k).$$

Кроме того к основным сущностям языка Т можно отнести типы и константы. Помимо базовых типов (числа с фиксированной или плавающей запятой, символы) допустимы и сложные типы, такие как статические массивы и структуры, полями которых могут быть в свою очередь как базовые, так и производные типы

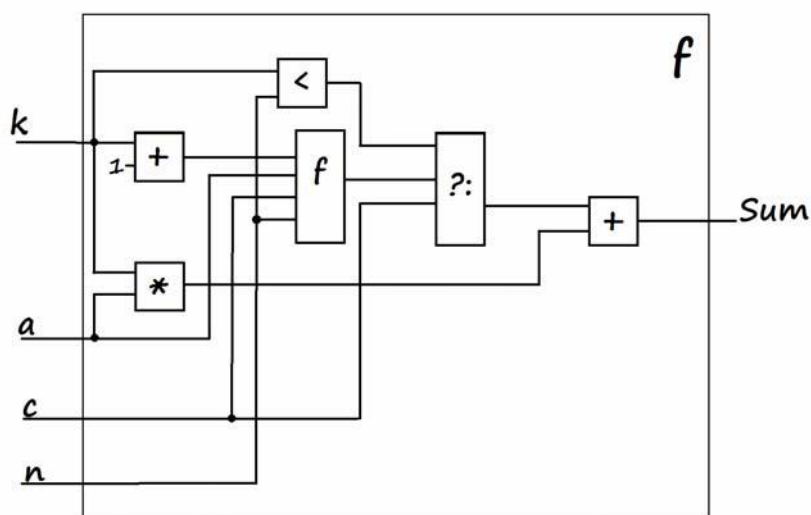


Рис. 1. Организации цикла при помощи рекурсии на примере итерационной суммы

В случае если необходимо использовать какие-либо значения от одного цикла выполнения программы до другого, могут понадобиться элементы памяти, но поскольку в данном языке отсутствуют переменные, то вводятся пара виртуальных входов и выходов, и при этом выход подается на вход. Поскольку данные входы и

выходы фактически не относятся к схеме и являются виртуальными, то ограничение на использование циклов не нарушается. Для мнимого элемента памяти так же допускается применение функциональных блоков, как и для реальных входов, обеспечив тем самым возможность изменения ее значения. Так как такой виртуальный вход может использоваться в качестве входа для множества других блоков, то необходимо предусмотреть задержку формирования выходного значения элемента памяти на этапе трансляции. Пример использования элемента памяти представлен на рис. 2.

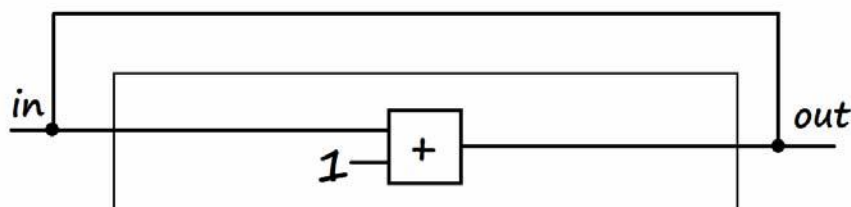


Рис. 2. Пример использования элемента памяти

Основной целью введения данного языка является возможность разбиения сложного алгоритма на несколько параллельно работающих программ, которые будут выполняться на отдельных ПЛК. Благодаря введенным ограничениям в язык Т, имеется такая возможность. Разбиение начинается с выбора очередного выхода схемы и последовательно выделения подчиненных блоков, т.е. тех, которые необходимы для вычисления выходного выражения. Таким образом, если разбиение удалось, то каждый из выделенных алгоритмов будет представлен в виде дерева. Однако, у данного подхода есть недостаток: избыточность. При разбиении программы возможна ситуация, когда выражение в одном и зависимых от него блоках будет вычислено несколько раз, на разных ПЛК.

Несмотря на данный недостаток, в общем случае, распараллеливание способствует увеличению эффективности программы за счет распределения ресурсов между ПЛК и сокращения общего времени обработки информации.

Схема, которую визуальнo можно изобразить как множество связанных блоков, описывается в формате XML. Реализуемый алгоритм представлен в виде двух файлов: файла, непосредственно описывающего схему и базы знаний, содержащей перечисление всех базовых типов и функций. Данный подход имеет ряд преимуществ:

- сокращается размер файла, определяющего сам алгоритм;
- возможность использования одного библиотечного файла для нескольких схем.

В процессе анализа и трансляции схемы можно выделить несколько этапов:

1. Используя файлы схемы и базовых функций, сформировать список блоков, определив их входы, выходы и выполняемые функции;
2. Преобразовать список блоков в граф;
3. Проверить корректность схемы, используя алгоритмы теории графов;
4. Разбить граф на множество деревьев, вершинами которых являются выходы схемы;
5. Преобразовать каждый выделенный алгоритм, представленный в виде дерева, в программный код на языке С.

Первым этапом трансляции алгоритма является предварительная обработка файлов схемы и базовых функций путем сопоставления и занесение данной информации в ОЗУ в виде списков.

Поскольку файл схемы содержит описание всех блоков, входов, выходов и функций, которые связаны только при помощи идентификаторов, дальнейшая обработка предполагает обеспечить связь между ними, например посредством указателей.

За данным этапом следует верификация схемы на корректность соответствие введенным ограничениям. В частности алгоритмы теории графов позволяют определить наличие обратных связей при помощи проверки графа на ацикличность. На данном этапе проводится проверка на полноту и непротиворечивость связей.

Следующий этап позволяет разделить один сложный алгоритм на несколько параллельно выполняющихся алгоритмов. Он заключается в построении деревьев, корнями которых являются каждый из выходов схемы, из общего графа схемы. Дерево формируется путем определения смежных вершин с очередной вершины, начиная с корня. Данный процесс завершается, когда все листья будут представлять собой входы или константы.

После того, как были получены множество деревьев, представляющих собой отдельную часть общего алгоритма, возможна трансляция каждого из них в код на языке С.

Таким образом, предложенный вспомогательный функциональный язык Т и схема его трансляции из языка FBD позволяют преобразовать большой, требовательный к ресурсам алгоритм в множество программных модулей, параллельно выполняющихся на разных ПЛК. Данный подход позволяет повысить эффективность автоматизированных систем управления за счет ускорения процесса обработки информации путем распределения выполняемой программы между множеством ПЛК.

Работа выполнена в рамках Федеральной целевой программы "Исследования и разработки по приоритетным направлениям развития разработки научно-технологического комплекса России на 2014-2020" (соглашение № 14.574.21.0045 от 06.19.2014)

Список литературы

1. Алгоритмы. Построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – М. : Вильямс, 2013.
2. Баби́чев, А. Распознавание и спецификация структур данных / А. Баби́чев. – М. : Ленанд, 2009.

К. О. Барабаш, П. К. Чечель, Д. В. Пащенко

Россия, Пенза, Пензенский государственный университет

КЛАССИФИКАЦИЯ ДАТЧИКОВ ДЛЯ ЭЛЕКТРОННОГО КАТАЛОГА

Рассматриваются различные способы классификации датчиков для электронного каталога.

В ходе работы по направлению Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития разработки научно-технологического комплекса России на 2014–2020» (соглашение № 14.574.21.0045 от 06.19.2014), одной из задач на ранних этапах проекта явилось создание универсальной и актуальной классификации датчиков.

Человек живет в аналоговом мире, где большинство устройств не являются цифровыми. Сигналы окружающего мира почти никогда не бывают электрическими. Для того, чтобы цифровые системы могли воспринимать информацию из внешнего мира, необходимы специальные интерфейсные устройства, преобразующие разнообразные физические величины в определенные электрические сигналы. Такими интерфейсными устройствами являются датчики.

В общем случае, датчик – это устройство, воспринимающее внешние воздействия и реагирующее на них изменением электрических сигналов. Выходными сигналами датчиков могут быть напряжение, ток или заряд, описываемые такими характеристиками как амплитуда, частота, фаза или цифровым кодом. Каждый датчик характеризуется набором входных параметров (любой физической природы) и набором выходных параметров.

Большинство датчиков являются электрическими. Это обусловлено следующими достоинствами электрических измерений:

- электрические величины удобно передавать на расстояние, причем передача осуществляется с высокой скоростью;
- электрические величины универсальны в том смысле, что любые другие величины могут быть преобразованы в электрические и наоборот;
- они точно преобразуются в цифровой код и позволяют достигнуть высокой точности, чувствительности и быстродействия средств измерений.

Системы классификации датчиков могут быть самыми разнообразными в зависимости от цели проведения классификации. Например, одна из наиболее распространенных – датчики могут быть разделены на активные и пассивные. Классификацию датчиков можно проводить и по другим признакам. По их характеристикам, материалам, из которых они изготовлены, по средствам детектирования, механизмам преобразования энергии воздействия в электрический сигнал, областям применения, видам внешних воздействий и т. д.

В настоящее время существует приблизительно следующее распределение доли измерений различных физических величин: температура – 50 %, расход (массовый и объемный) – 15 %, давление – 10 %, уровень – 5 %, количество (масса, объем) – 5 %, время – 4 %, электрические и магнитные величины – менее 4 %.

Для начала необходимо рассмотреть наиболее распространенные классификации датчиков.

Аналогово-цифровая классификация. Согласно этой классификации различают три класса датчиков:

1. аналоговые датчики, т.е. датчики, вырабатывающие аналоговый сигнал, пропорционально изменению входной величины;
2. цифровые датчики, генерирующие последовательность импульсов или двоичное слово;
3. бинарные (двоичные) датчики, которые вырабатывают сигнал только двух уровней: «включено/выключено» (иначе говоря, 0 или 1). Датчики такого типа получили широкое распространение благодаря своей простоте.

Классификация по виду измеряемой величины. В зависимости от вида входной величины различают: датчики механических перемещений (линейных и угловых), пневматические, электрические, расходомеры, датчики скорости, ускорения, усилия, температуры, давления и др.

А по виду выходной величины, в которую преобразуется входная величина, различают неэлектрические и электрические: датчики постоянного тока (ЭДС или

напряжения), датчики амплитуды переменного тока (ЭДС или напряжения), датчики частоты переменного тока (ЭДС или напряжения), датчики сопротивления (активного, индуктивного или емкостного) и другие.

Классификация датчиков по принципу действия. Данная классификация предусматривает деление датчиков на два класса: генераторные и параметрические (датчики-модуляторы).

Генераторные датчики осуществляют непосредственное преобразование входной величины в электрический сигнал.

Параметрические датчики входную величину преобразуют в изменение какого-либо электрического параметра датчика.

Проанализировав вышеописанные классификации, была создана своя уникальная классификация датчиков. Созданная классификация наиболее точно подходит для дальнейшего ее использования в базе данных, содержащей описание и классификацию датчиков. Данная классификация не имеет более двух уровней вложения, а термины используемые в описании классификации позволяют наиболее быстро и точно однозначно определить принадлежность датчика к той или иной категории. Правила классификации основывались на уже существующих принципах группировки датчиков по измеряемому параметру.

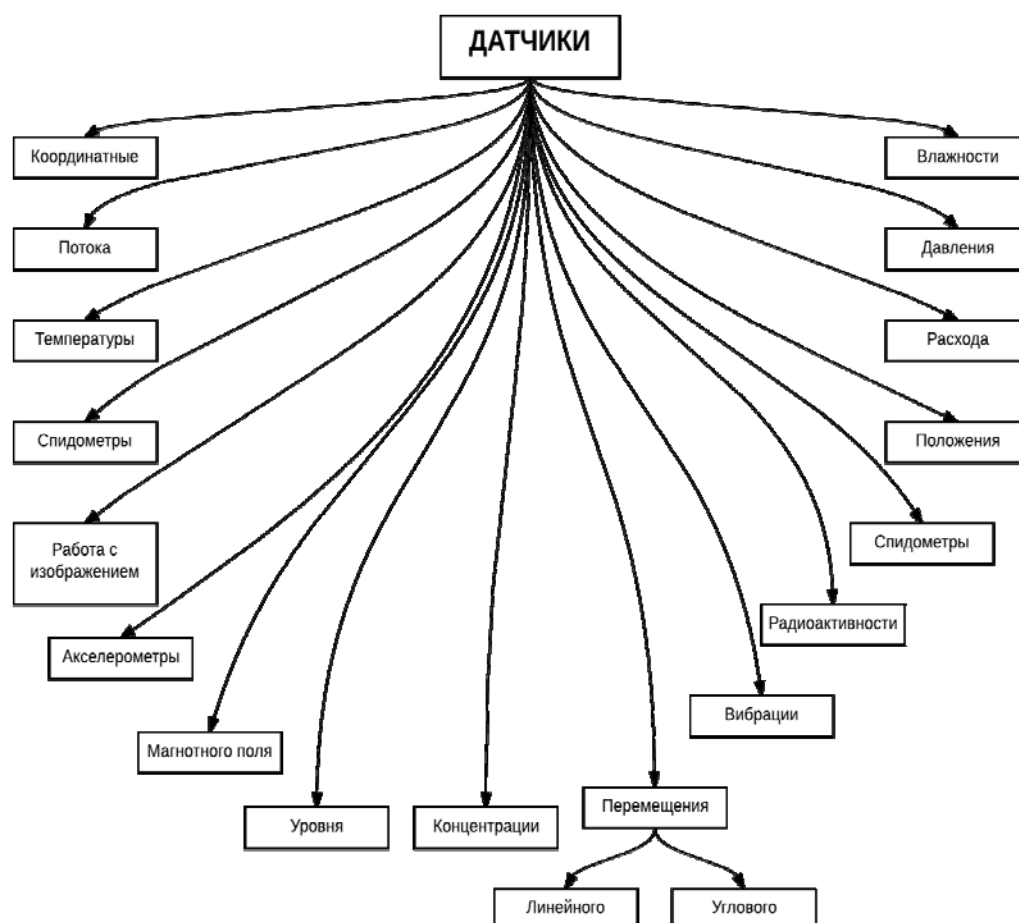


Рис. 1. Схема классификации датчиков

Проанализировав несколько предложенных классификаций, была разработана собственная универсальная и легко реализуемая в рамках реляционной базы данных классификация датчиков. Таким образом, наиболее подходящим основопола-

гающим критерием разбиения датчиков на группы и подгруппы оказался измеряемый параметр.

Список литературы

1. Гордин, Е. М. Основы автоматики и вычислительной техники / Е. М. Гордин, Ю. Ш. Митник, В. А. Тарлинский. – М. : Машиностроение, 2013. – 150 с.
2. Олссон, Г. Цифровые системы автоматизации и управления / Олссон Густав, Пиани Джангуидо. – СПб. : Невский Диалект, 2014. – 243 с.
3. Сазонов, В. В. Исследование реостатного датчика линейных перемещений : метод. указания к выполнению лабораторной работы / В. В. Сазонов. – М. : Машиностроение, 2013. – 40 с.
4. Шестаков, Д. Н. Микроволновые датчики промышленного применения / Д. Н. Шестаков. – СПб. : Невский Диалект, 2013. – 302 с.

Н. А. Пашков

Россия, Пенза, Пензенский государственный университет

МНОГОКРИСТАЛЛЬНЫЕ СИСТЕМЫ НА БАЗЕ ПЛИС. ОПТИМИЗАЦИЯ АЛГОРИТМА ПРОГРАММИРОВАНИЯ ПОДЧИНЕННЫХ ПЛИС ПО ИНТЕРФЕЙСУ SPI

Представлены примеры реализации платформ на базе нескольких микросхем ПЛИС. Описаны варианты оптимизации программной части для снижения временных затрат на конфигурирование микросхем.

Широкое распространение микросхем ПЛИС дало возможность быстрой разработки более гибких платформ по сравнению с полностью заказными интегральными схемами. Вычислительная мощность современных микросхем ПЛИС позволяет реализовывать на их основе достаточно сложные высокопроизводительные системы.

В зависимости от круга решаемых задач и объема реализуемых функций конечной системы, элементная база может включать одну и более микросхем ПЛИС.

Для некоторых функций обработки (например, шифрование) совместная реализация с основным алгоритмом работы системы нецелесообразна или вовсе невозможна из-за накладываемых на данный функционал ограничений, которые могут зависеть от требований заказчика или нормативно-технической документации. Как правило, обработка таких функций выполняется на отдельных узлах системы, в частности микросхемах ПЛИС.

Выполняемые микросхемой ПЛИС функции оказывают влияние на ее роль среди остальных микросхем ПЛИС. В случае, когда весь массив является равнозначным, все микросхемы ПЛИС конфигурируются автоматически при подаче питания (рис. 1). Данный способ подходит для систем с заранее известным или неизменным алгоритмом работы.

В случае необходимости обеспечить дополнительный контроль конфигурационных данных некоторых ПЛИС, обеспечить возможность выбора источника конфигурационных данных или возможность изменения части функционала в процессе

работы выполняется разделение массива схем ПЛИС на управляющие, выполняющие функции системного контроллера, и подчиненные, выполняющие функции сопроцессоров и различных схем поддержки (рис. 2).

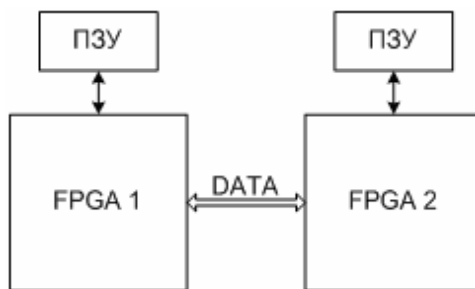


Рис. 1. Система с равнозначными микросхемами ПЛИС

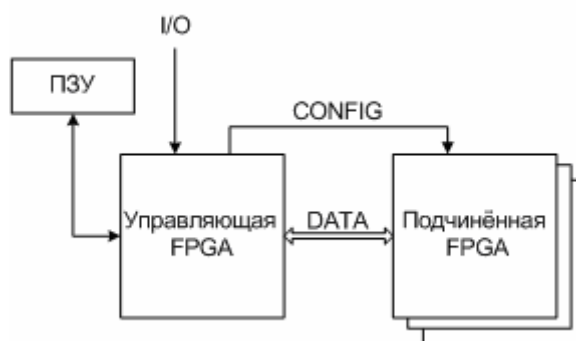


Рис. 2. Система с управляющей ПЛИС

Конфигурационные данные управляющих микросхем ПЛИС должны храниться в энергонезависимой памяти и при подаче питания на схему данные автоматически программируются в схему. Для подчиненных ПЛИС процесс конфигурирования схемы инициируется управляющей схемой. Для таких схем данные конфигурации могут располагаться как в энергонезависимой памяти, так и поступать извне по различным интерфейсам (например, JTAG, USB, Ethernet, UART).

В отличие от автоматического, ручное конфигурирование в процессе работы может занимать значительно больше времени. Увеличение продолжительности программирования схемы связано с необходимостью обмена данными, выполняемого по внутренней шине данных управляющей схемы, а также временные характеристики используемого интерфейса.

При программировании схемы в ручном режиме выполняются следующие действия:

- перевод ПЛИС в режим программирования, для чего на конфигурационных выводах микросхемы устанавливаются соответствующие сигналы;
- чтение конфигурационных данных из энергонезависимой памяти или внешнего интерфейса;
- проверка целостности данных прошивки;
- программирование ПЛИС;
- проверка результата программирования и перевод ПЛИС в рабочий режим.

В современных ПЛИС FPGA существует возможность выполнять конфигурирование по интерфейсу SPI. Такая возможность появилась благодаря наличию соответствующих выводов в микросхеме ПЛИС.

Протокол передачи по интерфейсу SPI представляет собой синхронный обмен данными между сдвигowymi регистрами ведущего и ведомого устройств. Однако в ряде случаев не требуется передачи данных в одном из направлений, например, при управлении микросхемами ЦАП надо только передавать, а при чтении данных из внешней Flash-памяти с SPI-интерфейсом требуется только принимать считанный блок байтов. В этих случаях для чтения данных необходимо в каждом цикле чтения байта предварительно произвести операцию «холостой» записи в буфер передатчика, после чего выполнить чтение байта данных. «Холостая» запись в данном случае служит для инициации очередного цикла обмена. Перед началом цикла обмена, в интерфейс SPI передается код операции.

Время конфигурирования напрямую зависит от количества целевых микросхем и «насыщенности» конфигурационных данных, которая влияет на размер файла прошивки. В процессе программирования системный контроллер принимает конфигурационные данные и сохраняет их для проверки и дальнейшего программирования в целевую схему. Однако из-за ограниченности внутренних ресурсов управляющей ПЛИС, большие файлы конфигурации необходимо хранить во внешнем ОЗУ, что приводит к дополнительным задержкам.

Использование интерфейсов с высокой производительностью для чтения/записи конфигурации или увеличение частоты работы схемы позволит сократить время программирования подчиненных ПЛИС. Однако для реализации данных методов необходимо изменять аппаратную платформу разрабатываемой системы, что приведет к увеличению времени и стоимости разработки. Оптимальным решением снижения временных затрат на конфигурирование ПЛИС является изменение программной и оптимизация программной составляющей системного контроллера.

Конфигурационные данные могут поступать из различных источников, однако начальная конфигурация, программируемая при старте системы, должна располагаться в энергонезависимой памяти. Оптимальным решением для хранения базовой прошивки является SPI Flash, позволяющая в полной мере использовать стандартный интерфейс программирования микросхем ПЛИС, а при необходимости использовать ее как статическую память для хранения произвольного набора данных (рис. 3).

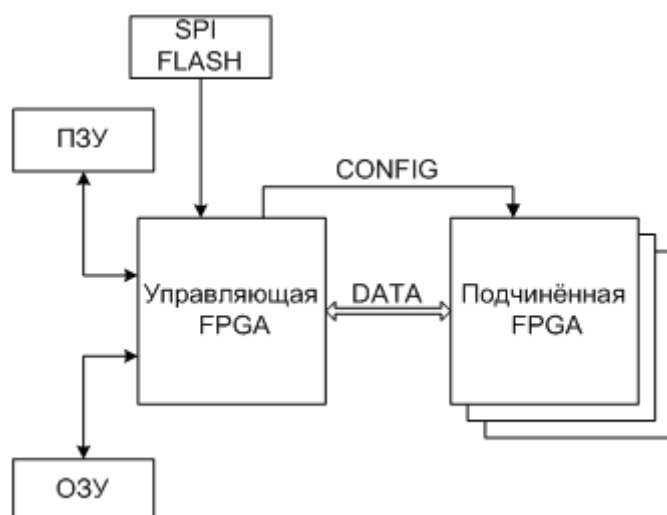


Рис. 3. Структура системы с flash-памятью для хранения конфигурации

После старта управляющей ПЛИС, выполняется настройка периферийного оборудования системы и установка рабочего режима. После этого из SPI Flash в память считывается конфигурационный файл для дальнейшего программирования.

ния подчиненной ПЛИС. Необходимость хранить промежуточные данные в памяти обусловлена невозможностью линейного обращения к содержимому SPI Flash.

Использование регистровой передачи позволит исключить из процесса программирования обмен с памятью. Смысл регистровой передачи заключается в организации алгоритма обмена таким образом, чтобы данные при чтении из статической flash-памяти, вместо пересылки в память для временного хранения на время чтения и проверки, пересылались сразу в целевую ПЛИС. Малое количество регистров общего назначения приводит к необходимости передачи данных в целевую схему «фреймами» небольшого размера, который зависит от количества используемых регистров. Такая модель обмена позволяет повысить скорость вхождения системы в рабочий режим и повысить ее надежность.

Еще один способ уменьшения времени программирования основан на использовании внутренних ресурсов контроллера SPI. В структуре контроллера SPI присутствуют приемное и передающее FIFO, которые в общем случае обеспечивают непрерывную передачу. Уменьшение задержек передачи конфигурации возможно за счет использования FIFO в качестве механизма предварительной выборки данных: перед началом цикла чтения конфигурации, в передающее FIFO контроллера SPI выполняется «холостая» запись данных по размеру передающего FIFO. Таким образом, непосредственно в цикле передачи, данные будут поступать без дополнительных задержек на обмен данными между сдвигowymi регистрами ведущего и ведомого устройств.

Совместное использование предложенных методов увеличения скорости программирования позволяет сократить затрачиваемое время в 1.5 - 2 по сравнению с классическим алгоритмом программирования подчиненных ПЛИС. Для получения более производительной системы допускается улучшение аппаратных характеристик, однако выполнение такой модернизации должно выполняться на этапе разработки аппаратной платформы, в то время как оптимизация программной компоненты возможна на любом этапе проектирования системы.

Список литературы

1. Березин, В. Программирование последовательных конфигурационных ПЗУ фирмы Altera по JTAG интерфейсу / В. Березин, Р. Золотуха // Компоненты и технологии. – 2005. – № 2.
2. Зотов, В. Инструментальные средства отладки устройств цифровой обработки сигналов, проектируемых на основе ПЛИС FPGA фирмы Xilinx серий Vitrex-6 и Spatran-6 / В. Зотов // Компоненты и технологии. – 2011. – № 10.
3. Новицкий, А. Синхронный последовательный интерфейс SPI в микроконтроллерах «от А до Я» и его реализация на примере ADuC70xx фирмы Analog Devices / А. Новицкий // Компоненты и технологии. – 2009. – № 3.

ГЕНЕРАЦИЯ КОДА ДЛЯ РАСПРЕДЕЛЕННОЙ АРХИТЕКТУРЫ С ПРИМЕНЕНИЕМ МНОГОУРОВНЕВОЙ ТРАНСЛЯЦИИ

Рассматриваются особенности генерации кода для распределенной архитектуры с применением многоуровневой трансляции программ, написанных на графических языках программирования логических контроллеров.

В последнее время в вопросах автоматизации технологических процессов наибольшее распространение получили решения на основе программируемых логических контроллеров (ПЛК). ПЛК являются достаточно мощным и универсальным средством. Другие решения, например, на основе микроконтроллеров, как правило, не пользуются особой популярностью, в силу их небольшой вычислительной мощности, и как следствие необходимости создания распределенных систем в случае реализации ресурсоемких алгоритмов. Описанный выше недостаток ограничивает применение микроконтроллеров до реализации небольших «легких» алгоритмов управления. Но при этом они имеют и явное преимущество – невысокая стоимость, чем не могут похвастаться ПЛК. Таким образом, даже при небольших затратах на аппаратное обеспечение, стоимость решения (речь идет о решении крупных задач, сравнимых с теми, которые решаются ПЛК) на базе микроконтроллеров суммарно обойдется дороже, чем на ПЛК, за счет реализации более сложного программного обеспечения (ПО) для распределенной системы.

Однако, если упростить и автоматизировать процесс разработки ПО для распределенной системы, то решения на базе микроконтроллеров теряют описанный выше недостаток, и становятся сравнимыми по сложности решаемых задач с решениями на базе ПЛК при меньшей стоимости. В свете выше сказанного, задача разработки способа упрощения и автоматизации разработки ПО для автоматизированных систем управления технологическим процессом (АСУ ТП) на базе распределенной системы микроконтроллеров становится в значительной степени актуальной.

Программирование микроконтроллеров осуществляется в большинстве случаев с применением языка С. Но несмотря на всю универсальность и мощь этого языка, он крайне неудобен для технологов, разрабатывающих алгоритмы управления. Именно по этой причине требуется привлечение к разработке С-программиста, которому также предстоит изначально изучить предметную область, а затем совместно с технологом разработать ПО для распределенной системы контроллеров. Все это требует временных и денежных затрат. Наиболее удобные средства разработки алгоритмов для технологов – это графические наглядные языки программирования, такие как FBD и SFC стандарта IEC 61131-3. Язык FBD (Function Block Diagram) описывает алгоритм в виде множества функциональных блоков, связанных между собой. Каждый функциональный блок имеет входы и выходы и выражает некую подпрограмму. Язык SFC (Sequential Function Chart) создан на базе математического аппарата сетей Петри, и описывает алгоритм как последовательность состояний и условий переходов между этими состояниями. Также для технолога даже с применением указанных языков будет проблематично реализовывать взаимодействие между вычислительными узлами распределенной сети. С учетом этих требований можно определить, что требуется алгоритм, позволяющий программу, написанную с применением языков SFC и FBD, для системы, состоящей из одного вычислительного узла (в данном случае виртуального), преобразовать к набору программ,

написанных с применением языка С, для распределенной системы, состоящей из нескольких микроконтроллеров.

Для реализации поставленной выше задачи предлагается способ многоуровневой трансляции. Так как программа, написанная с применением языков SFC и FBD, не может быть непосредственно транслирована на язык С и разбита на параллельно работающие модули для нескольких микроконтроллеров, целесообразно ввести промежуточный язык, который легко поддается разделению на параллельные ветви выполнения. Назовем этот язык – «язык Т».

Многоуровневая трансляция осуществляется в 3 этапа (рис. 1):

- 1) программа, написанная на SFC и FBD, транслируется в программу на FBD;
- 2) программа, написанная на FBD, транслируется в программу на языке Т;
- 3) программа, написанная на языке Т, транслируется в ряд программ на языке С.

На первом этапе исходный модуль, написанный на языках SFC и FBD, преобразуется в FBD-модуль. Это обусловлено определенной сложностью, связанной с прямой трансляции с SFC на Т, состоящей в том, что это усложнит весь алгоритм в целом и добавит некоторые дополнительные этапы. В то время как трансляция с FBD на Т осуществляется значительно проще, так как язык Т является развитием языка FBD.

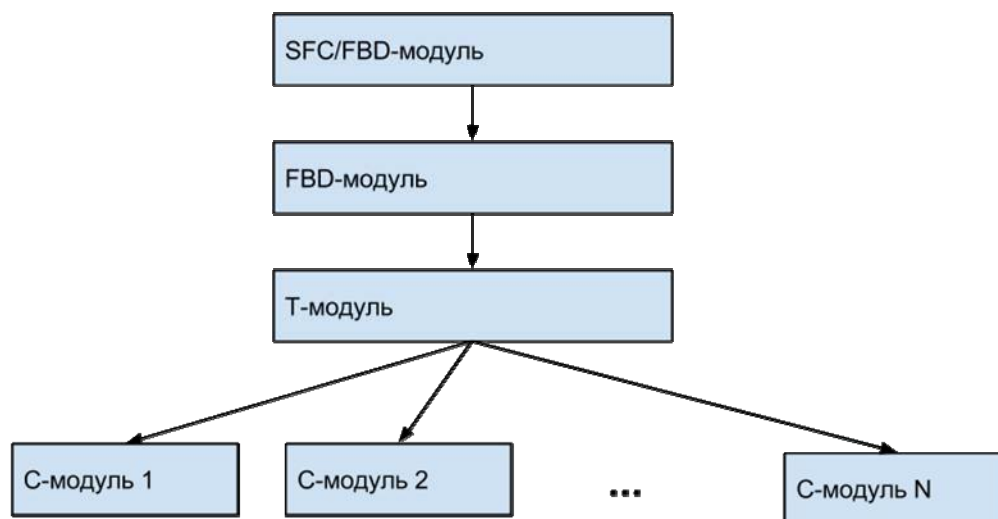


Рис. 1. Многоуровневая трансляция

На втором этапе трансляции FBD-модуль преобразуется в Т-модуль. Язык Т является функциональным языком программирования и дальнейшим развитием языка FBD, это продиктовано тем, что язык Т должен быть достаточно близок к графическим языкам программирования, используемыми технологами при разработке алгоритмов управления. Более подробное описание языка Т приведено в докладе «Алгоритм трансляции из функционального языка «Т» в программные модули на языке «С»» этого сборника.

Также на втором этапе производится верификация FBD-модуля, целью которой является выявление нарушений описанных выше ограничений, свойственных языку Т [1].

На третьем этапе происходит разбиение программы, написанной на языке FBD, на независимые модули, которые могут исполняться параллельно. Затем каждый из полученных модулей транслируется в программу на языке С.

Разбиение на независимые модули начинается с выбора очередного выхода схемы и последовательного выделения блоков, участвующих в вычислении выходного выражения [2]. Таким образом, если разбиение удалось, то каждый из выделенных алгоритмов будет представлен в виде дерева. Однако, у данного подхода есть недостаток: избыточность. При разбиении программы возможна ситуация, когда выражение в одном из зависимых от него блоках будет вычислено несколько раз, на разных микроконтроллерах.

Несмотря на данный недостаток, в общем случае, параллельное выполнение модулей способствует увеличению эффективности всей системы в целом за счет распределения ресурсов между микроконтроллерами сокращения общего времени обработки информации [3].

Таким образом, предложенный способ многоуровневой трансляции позволяет упростить и автоматизировать процесс разработки ПО для автоматизированных систем управления технологическим процессом (АСУ ТП) на базе распределенной системы микроконтроллеров. Технолог получает возможность разработки ПО с применением наглядных и привычных графических языков SFC и FBD, при этом получая в результате многоуровневой трансляции программы на С для микроконтроллеров. Таким образом, применение АСУ ТП на базе микроконтроллеров становится экономически целесообразным.

В дальнейшем планируется формализовать все этапы многоуровневой трансляции, и реализовать прототип подобного транслятора.

Работа выполнена в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития разработки научно-технологического комплекса России на 2014–2020» (соглашение № 14.574.21.0045 от 06.19.2014).

Список литературы

1. Бабичев, А. В. Распознавание и спецификация структур данных / А. В. Бабичев. – М. : Леланд, 2008.
2. Стратегия параллельной обработки массивов данных в системе объективного контроля радиотехнического комплекса радиолокационного дозора и наведения / Д. В. Пашенко, В. А. Мачалин, Д. А. Трокоз, М. П. Синев, А. Н. Токарев // Радиотехника. – 2010. – № 8. – С. 51–55.
3. Пашенко, Д. В. Проблемы построения многопоточной модели программного обеспечения экспертной системы авиационных радиолокационных комплексов / Д. В. Пашенко, Д. А. Трокоз // Известия высших учебных заведений. Поволжский регион. – 2011. – № 2 (18). – С. 21–30.

А. Р. Алеева, Д. В. Пашенко, А. С. Бычков

Россия, Пенза, Пензенский государственный университет

ОБЪЕКТИВНЫЙ КОНТРОЛЬ СИСТЕМ УПРАВЛЕНИЯ ОБЪЕКТАМИ НАЗЕМНОЙ ИНФРАСТРУКТУРЫ РАКЕТНО-КОСМИЧЕСКОЙ ТЕХНИКИ

Приведено исследование видов объективного контроля систем управления объектами наземной инфраструктуры ракетно-космического комплекса на примере заправки и слива ракетно-космической техники при подготовке к запуску.

Введение

Развитие ракетно-космической промышленности играет немаловажную роль в обороноспособности страны. В последние годы наметился определенный прогресс в развитии ракетно-космической промышленности России.

Производство ракетно-космической техники относится к категории сложных систем, и как всякая сложная система представляет собой совокупность большого числа подсистем и элементов, объединенных прямыми и обратными связями в многоступенчатую иерархическую структуру, по которой циркулируют информационные, энергетические и другие потоки (рис. 1).

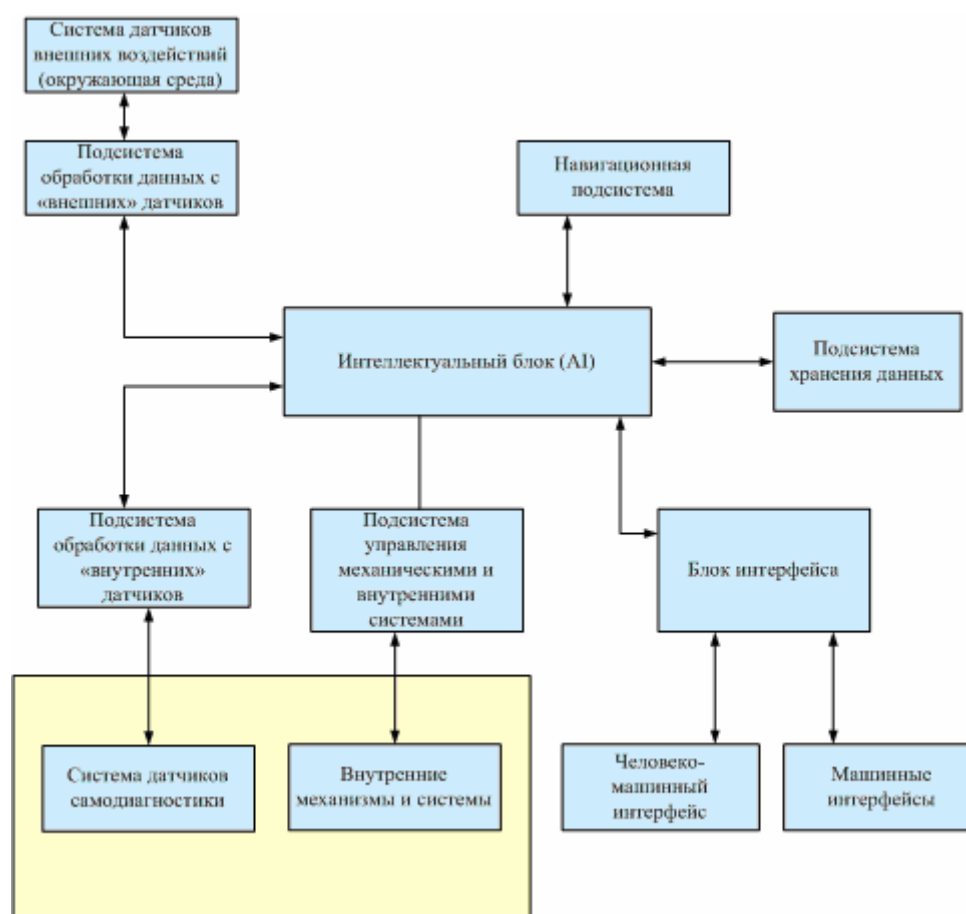


Рис. 1. Пример структурной схемы сложной технической системы

Ракетно-космический комплекс это – сложнейшая радиотехническая, электронная и информационно-вычислительная система, надежная и точная работа которой обеспечивается, в том числе, и наземной системой обработки информации.

Поэтому, как и в любой сложной системе, в ракетно-космическом комплексе необходимы и существуют аппаратно-программные средства управления наземной инфраструктуры ракетно-космической техники (РКТ). Без работы таких средств обеспечить качественное функционирование сложной системы РКТ практически невозможно.

Процесс заправки и слива РКТ при подготовке к запуску

Одним из самых технологически-сложных процессов являются процессы заправки и слива РКТ при подготовке к запуску. Автоматизация управления данными процессами представляет большой практический интерес. В настоящее время широкое применение получили управляющие системы на основе микроконтроллеров.

Согласно технологическому процессу – весь процесс заправки/слива разбивается на ряд автономных подзадач (операций). Рассмотрим более подробно структуру системы заправки (рис. 2).

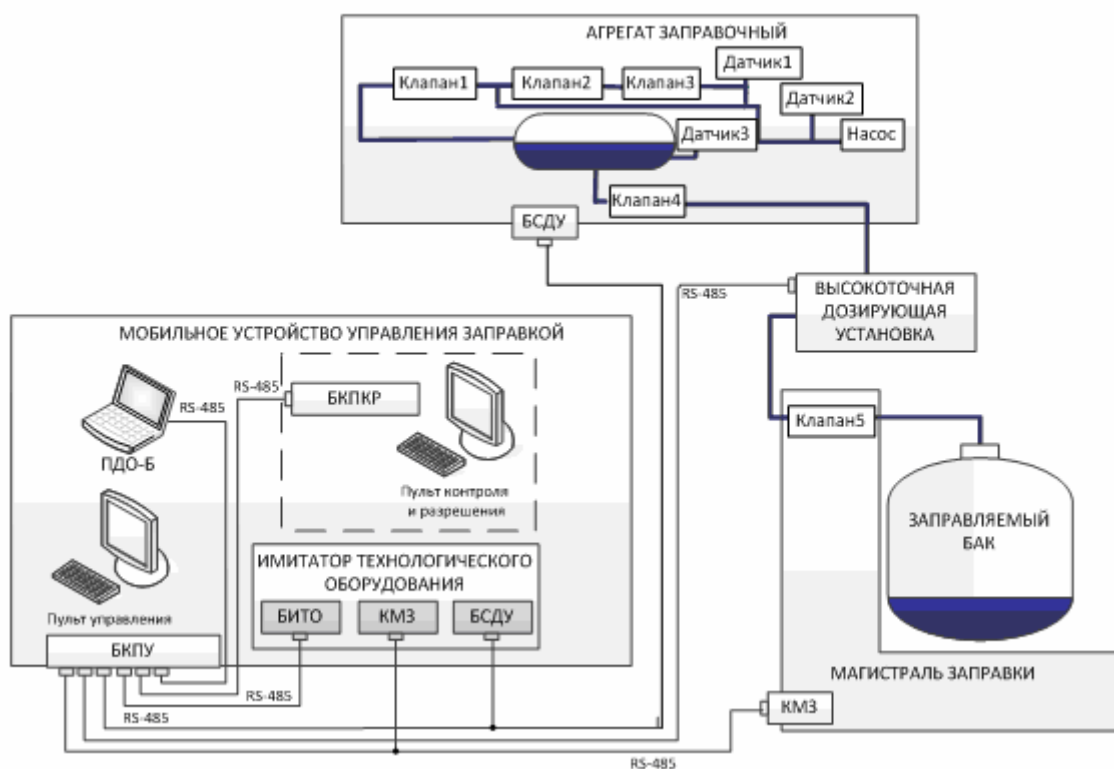


Рис. 2. Структура системы заправки

Условные обозначения:

ПДО-Б – Пульт документирования и отображения;

БКПУ – Блок контроллера пульта управления;

БКПКР – Блок контроллера пульта контроллера и решения;

БИТО – Блок имитатора технологического оборудования;

КМЗ – Контроллер магистрали заправки;

БСДУ – Блок сбора данных и управления.

Система состоит из мобильного устройства управления заправкой, заправочного агрегата, магистрали заправки, высокоточной дозирующей установки и заправляемых баков. Заправочный агрегат перегоняет насосами топливо из автозаправочных цистерн. Управление клапанами и насосами реализовано через управляющие реле. Высокоточная дозирующая установка обеспечивает высоко-точный контроль над температурой, давлением и выданной дозой топлива. Показания датчиков выводятся на табло и, руководствуясь ими, оператор в ручном режиме или контроллер в автоматическом режиме должны предпринимать определенные действия.

Одной из основных проблем, возникающих при испытании и эксплуатации автоматизированных систем управления, является обеспечение функций контроля процесса эксплуатации. Поэтому в их состав должны входить средства объективного контроля (СОК), обеспечивающие анализ состояния технических средств и действий операторов, реализованные в виде пульта документирования и отображения (ПДО). Программное обеспечение ПДО (ПО ПДО) должно быть построено с использованием технологии экспертных систем. Применение технологии экспертных систем (ЭС) позволяет расширить функциональные возможности СОК входящих в состав автоматизированных систем управления наземной инфраструктурой ракетно-космической техники.

ПО ПДО должно включать в себя:

- программу документирования;
- программу анализа.

Таким образом, была рассмотрена СОК на примере заправки и слива РКТ.

Математические модели объективного контроля процесса эксплуатации

В соответствии с процедурами СОК при проведении объективного контроля можно выделить три вида проводимого контроля:

- экспресс контроль проводится над ограниченным объемом данных с целью выявить неисправности аппаратуры и проанализировать форматизированные результаты работы комплекса за ограниченный промежуток времени;
- оперативный контроль проводится уже над всеми зарегистрированными в определенный промежуток времени данными с использованием более широкой номенклатуры сценариев контроля;
- специальный контроль предполагает детальное изучение всей задокументированной в процессе работы информации с использованием интерактивных режимов обработки и проверки задания.

Создание интеллектуальных систем анализа данных является сложной задачей особенно в области новых сложных технических систем и предполагает использование развитого комплекса математических методов и алгоритмов по формализации исходных данных.

В настоящее время известно большое число методик и алгоритмов по анализу данных и обработке экспертной информации. Выбор математического аппарата для разработки и реализации экспертных систем – сложный процесс. На такие системы накладываются ограничения, связанные со средой функционирования. Это работа в реальном масштабе времени, обработка большого объема информации за ограниченное время и т.д.

Проведем обзор основных методологий интеллектуального анализа данных в экспертных системах.

Продукционная модель. Главная идея этой группы методов состоит в организации БЗ в виде набора фактов, имеющих место в рассматриваемой предметной об-

ласти. Знания («факты») представляются в виде формул некоторой логики. В процессе работы в БЗ появляются новые факты, которые формируются при использовании операций логического вывода – поиска решения из набора исходных данных по заданным правилам (аксиомам). Параллельно с наполнением БЗ, идет процесс проверки непротиворечивости фактов в данной области.

Продукционная модель предполагает описание состояний в виде множества утверждений. Для удобства можно формулировать правила таким образом, чтобы внутри каждого из них переменные объединялись логическими операциями. Другими словами, БЗ нужно представить в виде системы продукций вида:

ЕСЛИ условие _ 1 ТО вывод _ 1.

Предикатные модели. В предикатных системах появляется аспект выводимости, поскольку исчисление предикатов позволяет формировать доказательства теорем на формальном языке. Законы теории исчисления предикатов дают преимущество в построении логических цепочек поиска решения.

Наиболее широкое применение нашла логика предикатов первого порядка, обладающая полнотой и выводимостью, что позволяет строить цепочки доказательств по аксиомам первого порядка.

Семантические сети. Основная идея метода состоит в представлении сложных объектов предметной области в виде модели набора сущностей и связей между ними. Графически семантические сети изображаются в виде направленных графов, где вершинам соответствуют сущности (объекты реального мира), а связи в виде помеченных дуг графа, отражают отношения между ними.

Отдельно в теории семантических сетей рассматриваются так называемые К-сети. Модели на их основе используют вопросно-ответные методики и исчисление предикатов с развитой системой индексирования объектов. В К-сети существуют три типа связей между блоками: отрицание, дизъюнкция и импликация. Одним из важнейших свойств К-сети является то, что она дает возможность представления произвольно сгруппированных кванторов существования и общности. Данное свойство позволяет гибко описывать ПО на выбранном языке, что необходимо не только для отражения логической структуры объекта, но и для минимизации числа правил в формальном описании.

В результате анализа рассмотренных методов проектирования, можно сделать вывод о возможности предпочтительного использования продукционных моделей и логики предикатов для реализации экспертной системы анализа работы РКТ, как методах наиболее близких к проектируемой системе.

Список литературы

1. Пашенко, Д. В. Представление знаний в системах объективного контроля авиационных комплексов радиолокационного дозора и наведения / Д. В. Пашенко, М. П. Синев // Известия высших учебных заведений. Поволжский регион. Технические науки. – Пенза : Информационно-издательский центр ПГУ, 2010.
2. Бармин, И. В. Технологические объекты наземной инфраструктуры ракетно-космической техники / И. В. Бармин, Н. М. Корнеев. – М., 2005
3. Советов, Б. Л. Моделирование систем / Б. Л. Советов, С. А. Яковлев. – М. : Высш. шк., 2001.
4. Пашенко, Д. В. Методика построения систем объективного контроля авиационных радиолокационных комплексов / Д. В. Пашенко, М. П. Синев // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2009. – № 4 (12). – С. 49–59.

А. С. Башев, А. Ю. Афонин

Россия, Пенза, Пензенский государственный университет

ПРЕИМУЩЕСТВА ИСПОЛЬЗОВАНИЯ ЯЗЫКА LUA ДЛЯ РАЗРАБОТКИ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Обоснованы преимущества внедрения скриптового языка программирования Lua в информационные системы, разрабатываемые на универсальном языке программирования C++.

Одной из первых практических задач при разработке программного продукта становится выбор языка программирования. Этот выбор сугубо индивидуален и может проводиться по множеству параметров.

На рис. 1 представлен график отражающий количество найденных вакансий в тексте которых встречается упоминание конкретного языка программирования. Данные представлены порталом headhunter.ru и ограничены регионом «Россия» и профессиональной областью «Информационные технологии, Интернет, телеком».

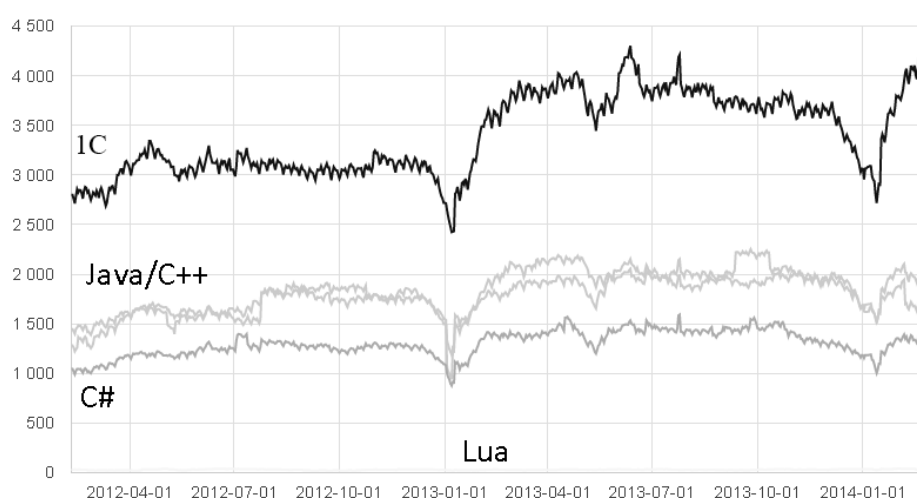


Рис. 1. Популярность языков программирования по версии headhunter.ru

Устоявшиеся универсальные языки, такие как Java, C++ и C# оправданно занимают лидирующую позиции, что подтверждается статистикой по версии TIOBE programming community index, оценивающей популярность языков программирования на основе подсчета результатов поисковых запросов, содержащих название языка.

Каковы же причины, которые могут инициировать использование «непопулярного» (исходя из представленного графика) языка Lua?

Предположим, что для разработки сложной информационно-вычислительной системы, программист выбирает универсальный язык программирования C++ поскольку обладает достаточным опытом разработки на нем и учитывает факт того, что для языка C++ существует огромная база различных библиотек, кодов, средств разработки, а код написанный на нем отличается высокой скоростью исполнения и переносимостью. Сам по себе язык C++ не тривиален и требует профессионального подхода к написанию кода. Программные модули информационной системы, состоящие из нескольких классов, образуют сложные связи приводящие к усложнению программной архитектуры, что увеличивает количество потенциальных ошибок и количество тестирующего кода.

Возможны случаи, когда определенная часть системы обособливается во вспомогательную библиотеку, что позволяет отделить часть программной логики в «черный ящик», но для этого архитектура информационной системы должна позволять реализовать данный подход или требовать серьезной переработки.

Современные статические анализаторы кода C++ указывают на большое количество программных ошибок в продуктах даже крупных компаний, значимая часть которых приводит к утечкам памяти, неработоспособности и неопределенному поведению информационной системы.

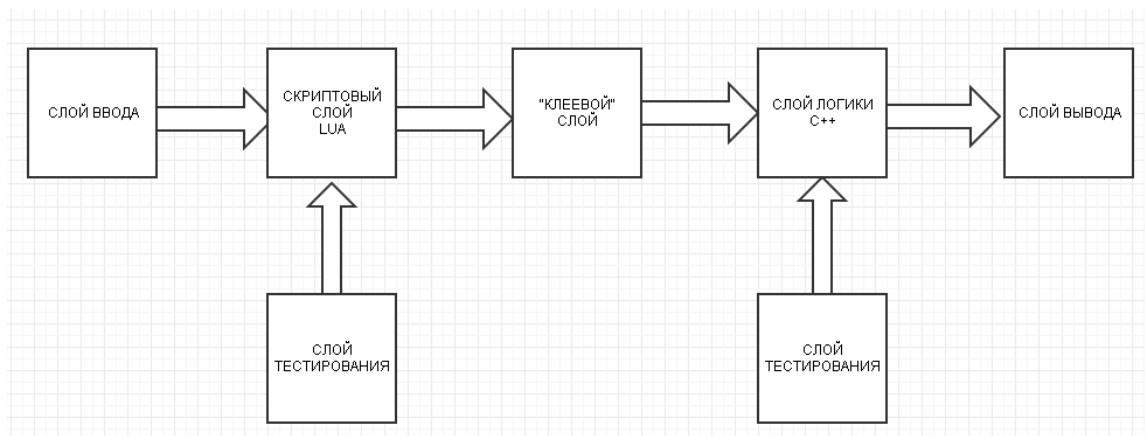


Рис. 2. Схема информационной системы с внедренным языком Lua

На рис. 2 представлена схема архитектуры информационной системы с внедренным скриптовым слоем написанным на языке Lua. Он содержит в себе простую, не требующую особого быстродействия или ресурсов памяти логику информационной системы, использующую функции из высокопроизводительного C++ слоя – требующего ручного управления памятью и использующие множество различных библиотек, недоступных в Lua.

Внедрение языка Lua в язык C++ требует небольшой подготовки по созданию промежуточного слоя между двумя языками и она не связана с изменением уже существующей логики информационной системы. При этом требуется написать интерфейс вызова Lua функций из C++ и наоборот, что подразумевает работу со стекком Lua (передача параметров в Lua функцию, возврат результатов и т.д.) и может быть упрощено с помощью множества вспомогательных библиотек.

Lua – мультипарадигменный язык программирования. Это позволяет использовать функциональный подход в скриптовом слое Lua, в отличие от C++ кода, использующего объектно-ориентированный подход и наоборот. Язык имеет простой и выразительный синтаксис, который позволяет ускорить скорость освоения на начальном этапе обучения.

Основная структура данных Lua – таблица, динамический ассоциативный массив, и поскольку язык является бестиповым, это позволяет создавать сложные структуры данных без необходимости объявления структур или классов.

В языке используется автоматический сборщик мусора, над которым, в случае необходимости можно получить косвенный контроль используя метаданные или прямой вызов. Метаданные – это мощный механизм, позволяющий определять наследование, перезагружать операторы и т.д.

Lua распространяется под лицензией MIT, что обеспечивает свободное использование в open source и коммерческих продуктах. Являясь мультиплатформенным языком, Lua возможно применять на любой платформе, для которой существует C компилятор. Исходный код на языке C является открытым.

Lua имеет большое количество модулей которые позволят убрать зависимости в подобных библиотеках в C++ слое и перенести большую часть логики в скриптовый слой.

Язык позволяет хранить данные из C++ посредством `user_data`, `lightuserdata`. Это позволяет хранить указатели на C++ объекты классов, вместе с логикой их отображения в таблицах Lua.

Интерпретируемая природа языка и его использование увеличивает общий уровень инкапсуляции кода. Lua программисты могут работать со скриптами и исполняемым файлом информационной системы, не имея доступа к исходному коду на C++. Более того, даже пользователи информационной системы смогут без труда модифицировать логику работы информационной системы под свои нужды, в случае необходимости.

Перспектива C++ программиста – создание высокопроизводительной архитектуры информационной системы и адекватного ее покрытия тестами, с рядом функций для использования в Lua скриптах. Это позволяет сконцентрироваться на вопросах производительности и потребления ресурсов, а не постоянной доработке архитектуры информационной системы из-за появления новой логики и написании тестов на подобные изменения.

Таким образом, разработка информационной системы разделяется на два «потока». Это создает возможность использование C++ «заглушек» для в Lua скриптах, что позволяет Lua программистам независимо создать новую логику, до тех пор пока «заглушки» будут заменены на реальные функции.

Lua программист может писать тесты на Lua для C++ функций которые он использует в своих скриптах, обеспечивая перекрестное межъязыковое тестирование.

Основная проблема Lua это производительность. По сравнению с C++ в некоторых случаях разница достигает от 3 до 75 раз [1]. К тому же стандартные механизмы не способны работать с объектами C++ классов. Но это решается написанием статических методов классов использующих объект класса как параметр или же подключением сторонних библиотек.

Существует еще несколько недоработок Lua, но поскольку интерпретатор языка имеет открытый исходный код, сообщество постоянно развивает и улучшает язык и даже создает модификации.

LuaJIT – одна из модификация языка Lua, использующая JIT компилятор, а также интерпретатор, написанный на ассемблере, полностью совместимая с Lua. Она отличается увеличенной производительностью по сравнению с Lua [2]. Что решает основную проблему языка в сравнении с C++. Также LuaJIT включает в себя поддержку улучшенных сопрограмм, побитовые операции и другие улучшения, и является достойным примером развития языка Lua.

Одним из таких значимых нововведений является библиотека FFI, позволяющая вызывать функции и создавать структуры данных языка C в Lua скрипте без использования слоя взаимодействия. Это позволяет значительно упростить работу по связыванию Lua и C++ кода. Ниже приведен пример вызова C функции с помощью библиотеки FFI.

```
local ffi = require("ffi")
ffi.cdef[[
    int printf(const char *fmt, ...);
]]
ffi.C.printf("Hello %s!", "world")
```

Язык программирования Lua обладает рядом недостатков и его популярность довольно низка в России, но, несмотря на это, он успешно используется во множестве сложных коммерческих программных продуктах мирового уровня, таких как World of Warcraft и Adobe Photoshop Lightroom [3]. Существуют варианты разработки на языке C++ с внедряемыми интерпретируемыми языками. Например, разработчики EVE Online используют Stackless Python в качестве языка разработки всей игровой логики, в то время как C++ отвечает за видео/аудио вывод на экран, работу с сетью и прочие требовательные по производительности процессы.

Список литературы

1. Lua vs C++ g++ | Computer Language Benchmarks Game Performance: x86/x64. – URL: <http://benchmarksgame.alioth.debian.org/u32/benchmark.php?test=all&lang=lua&lang2=gpp&data=u32>
2. Performance Comparison. – URL: <http://luajit.org/performance.html>
3. Where Lua is used. – URL: <https://sites.google.com/site/marbux/home/where-lua-is-used>

Л. Н. Бондаренко

Россия, Пенза, Пензенский государственный университет

О СЖАТИИ ТЕЛЕМЕТРИЧЕСКОЙ ИНФОРМАЦИИ ДЛЯ ЕЕ ОБРАБОТКИ МЕТОДАМИ ПРОНИ И РУТИСХАУЗЕРА

Рассмотрен подход к сжатию телеметрической информации для последующего применения методов Прони и Рутисхаузера. Этот подход основан на применении формул приближенного интегрирования и удалении «плохих» экспонент.

Для целей передачи цифровой телеметрической информации с последующей ее математической обработкой часто возникает необходимость в сжатии имеющихся данных. Это связано не только с требованием уменьшения времени передачи больших объемов данных, но и с применяемыми методами математического моделирования этих данных для последующей их обработки, заключающейся в оценивании данных, прогнозировании и т.п.

Действительно многие математические методы моделирования данных, использующие их аппроксимацию в виде линейной комбинации достаточно простых фиксированных функций, являются плохо обусловленными. Иначе, погрешности, имеющиеся в исходных данных для решения задачи аппроксимации и появляющиеся из-за неточности аналоговых методов измерений, ограничений аналого-цифрового преобразования и т.д., приводят к недостоверным результатам аппроксимации, что особенно часто проявляется при аппроксимации с достаточно большими объемами исходных данных.

Отметим, что сами аппроксимационные агрегаты являются средствами сжатия информации, так как они содержат значительно меньше информативных параметров, чем моделируемая ими исходная информация. Конечно, алгоритмы получения информативных параметров соответствующих аппроксимационных агрегатов могут быть достаточно сложны для вычисления этих параметров сразу после аналого-цифрового преобразования, например, с использованием микропроцессора, но вполне приемлемы после передачи сжатой более простыми методами телеметрической информации для компьютера.

Многие физические процессы и сигналы, применяемые на практике, могут хорошо аппроксимируются с помощью экспоненциальных сумм

$$x_n(t) = \sum_{k=1}^n c_k e^{\alpha_k t} \quad (1)$$

где параметры c_k и α_k , в общем случае, являются комплексными числами, а показатели α_k удовлетворяют условию $\operatorname{Re} \alpha_k \leq 0$ для всех k , что связано с устойчивостью рассматриваемых процессов и сигналов.

Задача аппроксимации сигнала $x(t)$ в промежутке $[t_0, T]$ суммой (1) нелинейна и не имеет эффективных методов решения, но существуют мощные алгоритмы Прони [1] и Рутисхаузера [2] экспоненциальной интерполяции на эквидистантных узлах. Эта задача состоит в определении $2n$ параметров в (1) по результатам $N = 2n$ измерений $\{x(t_i)\}_{i=0}^{N-1}$ сигнала $x(t)$ в равноотстоящие моменты времени $\{t_i\}_{i=0}^{N-1}$, $t_i = t_0 + i\Delta t \in [t_0, T]$, где Δt – шаг дискретизации, определяемый следующим соотношением $\Delta t \leq (T - t_0)/(N - 1)$, и $x(t_i) = x_n(t_i)$, $i = 1, \dots, N$.

Оригинальный алгоритм Г. Прони основан на двойном применении метода наименьших квадратов (МНК) и решении определенного характеристического уравнения [1, 3]. Эффективный алгоритм частных и разностей (QD-алгоритм) Г. Рутисхаузера основан на простых правилах ромба

$$\begin{cases} e_j^{(i)} = e_{j-1}^{(i+1)} + q_j^{(i+1)} - q_j^{(i)}, \\ q_{j+1}^{(i)} = q_j^{(i+1)} \frac{e_j^{(i+1)}}{e_j^{(i)}}, \end{cases} \quad (2)$$

которые применяются при $i = 0, 1, \dots, N - 1$, $j = 1, 2, \dots$ и следующих начальных условиях $e_0^{(i)} = 0$, $q_1^{(i)} = x(t_{i+1})/x(t_i)$.

Эти правила успешно решают задачу рациональной интерполяции функции на эквидистантных узлах, к которой с помощью z -преобразования может быть сведена исходная задача, причем для получения окончательного результата также требуется решение соответствующего характеристического уравнения [2, 3].

Алгоритмы Прони и Рутисхаузера сильно различаются по своим основным вычислительным характеристикам [3], но оба имеют плохую обусловленность.

Причем плохая обусловленность этих методов имеет различную природу: в методе Прони она, прежде всего, связана с плохой обусловленностью матриц в уравнениях МНК, а в методе Рутисхаузера – с накоплением ошибок вычислений при многочисленном применении формул схемы (2).

От структурного параметра n зависит сложность решения задачи экспоненциальной интерполяции и согласованность между (1) и реальным сигналом $x(t)$, причем для оценки согласованности на практике можно пользоваться максимальной приведенной погрешностью

$$\delta_n = \frac{\max_i |x(t_i) - x_n(t_i)|}{\max_i |x(t_i)|}. \quad (3)$$

Отметим, что параметр n является степенью характеристического уравнения как в алгоритме Прони, так и в алгоритме Рутисхаузера. Поэтому высокая чувствительность корней характеристических уравнений к изменению их коэффициентов при больших значениях n также может привести к недостоверности полученных ре-

зультатов. В частности, в сумме (1) может быть нарушено исходное условие $\operatorname{Re} \alpha_k \leq 0$ для всех k .

В общем случае, при решении задачи методом Прони количество измерений N за счет использования МНК может превышать $2n$, а в методе Рутисхаузера при $N \geq 2n$ схема QD-алгоритма (2) обрывается на требуемом шаге.

Так как телеметрические данные могут содержать значительные объемы измерений, то возникает задача сжатия этих данных с целью эффективного применения алгоритмов Прони и Рутисхаузера.

Для простоты будем считать, что $t_{N-1} = T$, число эквидистантных измерений с шагом Δt равно $N > 2n$ и $N - 1 = Mm$, где $M = 2n$, а число m велико.

Отсчеты сигнала разобьем на M одинаковых по длине блоков $\{t_i\}_0^m$, $\{t_i\}_m^{2m}, \dots, \{t_i\}_{(M-1)m}^{Mm}$ и проинтегрируем выражение (1) в промежутке

$$[t_0 + jm\Delta t, t_0 + j(m+1)\Delta t],$$

где $j = 0, 1, \dots, M-1$

$$\tilde{x}_n(t_0 + jm\Delta t) = \int_{t_0 + jm\Delta t}^{t_0 + (j+1)m\Delta t} \sum_{k=1}^n c_k e^{\alpha_k t} dt = \sum_{k=1}^n \frac{c_k e^{\alpha_k t_0}}{\alpha_k} (e^{\alpha_k m\Delta t} - 1) e^{\alpha_k jm\Delta t} = \sum_{k=1}^n \tilde{c}_k e^{\alpha_k (t_0 + jm\Delta t)}, \quad (4)$$

где $\tilde{c}_k = \frac{e^{\alpha_k m\Delta t} - 1}{\alpha_k} c_k e^{\alpha_k t_0}$, $j = 0, \dots, M-1$.

Сравнение выражения (4) с (1) показывает, что функция

$$\tilde{x}_n(s) = \sum_{k=1}^n \tilde{c}_k e^{\alpha_k s} \quad (5)$$

в узлах $s_j = t_0 + jh$, $h = m\Delta t$ принимает значения (4).

Таким образом, нахождение функции (5) решает задачу экспоненциальной интерполяции вспомогательной функции $\tilde{x}(s)$, принимающей в узлах $\{s_j\}_0^{M-1}$ значения $\{\tilde{x}_n(s_j)\}_0^{M-1}$, причем $h = m\Delta t$ – новый шаг дискретизации.

Так как параметры α_k в формулах (1) и (5) одинаковы, то соотношение

$$c_k = \alpha_k e^{-\alpha_k t_0} (e^{\alpha_k m\Delta t} - 1)^{-1} \tilde{c}_k$$

позволяет получить искомые параметры выражения (1) по известным параметрам соотношения (5).

При нахождении экспоненциальной суммы (5) также могут использоваться алгоритмы Прони и Рутисхаузера, но при этом исходные данные объемом N заменяются исходными данными объемом M , т.е. для этого должно быть предварительно выполнено сжатие информации.

Разбиение отсчетов на M блоков позволяет в принципе рассматривать внутри каждого блока свой шаг дискретизации, что приводит к частному случаю задачи экспоненциальной интерполяции с различными шагами дискретизации.

Для приближенного вычисления значений $\{\tilde{x}_n(s_j)\}_0^{M-1}$ в соответствии с выражением (4) можно использовать различные интерполяционные квадратурные формулы, позволяющие численно получать значения интегралов по известным значениям подынтегральных функций в заданной системе узлов.

При этом, очевидно, что предпочтение следует отдавать простейшим формулам с приемлемой оценкой погрешности.

Так формула трапеций приводит к следующему результату

$$\tilde{x}_n(s_j) \approx \Delta t \left(\frac{1}{2} x(s_j) + \sum_{i=1}^{m-1} x(s_j + i\Delta t) + \frac{1}{2} x(s_{j+1}) \right) \quad (6)$$

с оценкой погрешности $O\left(\frac{m(\Delta t)^3}{12}\right)$.

Более точной является формула Симпсона, при применении которой m предполагается четным числом

$$\tilde{x}_n(s_j) \approx \frac{\Delta t}{6} \left(x(s_j) + 4 \sum_{i=1}^{\frac{m}{2}} x(s_j + (2i-1)\Delta t) + 2 \sum_{i=1}^{\frac{m}{2}-1} x(s_j + 2i\Delta t) + x(s_{j+1}) \right) \quad (7)$$

с оценкой погрешности $O\left(\frac{m(\Delta t)^5}{2880}\right)$.

Вычисления с помощью формул (6) и (7) легко выполняются на микропроцессорах, что нельзя сказать о применении алгоритмов Прони и Рутисхаузера.

При использовании алгоритмов Прони и Рутисхаузера в практических вычислениях возможно получение корней характеристического уравнения с положительной действительной частью. Это сразу резко ухудшает максимальную приведенную погрешность (3), т. е. приводит к недостоверным результатам.

Простейшим приемом устранения этого недостатка указанных алгоритмов является исключение «плохих» корней характеристического уравнения, так как им часто соответствуют малые значения коэффициентов C_k в соотношении (1).

Имитационное моделирование в пакетах Matlab+Simulink и Maple алгоритмов Прони и Рутисхаузера с использованием сжатия данных дало хорошие значения приведенной погрешности (3). При этом для имитации исходной информации требуемого вида использовались датчики случайных чисел, а также «обрезание» числа верных значащих цифр с последующим округлением результата.

Список литературы

1. Марпл-мл., С. Л. Цифровой спектральный анализ и его приложения / С. Л. Марпл-мл. – М. : Мир, 1990. – 584 с.
2. Рутисхаузер, Г. Алгоритм частных и разностей / Г. Рутисхаузер. – М. : Изд-во иностр. лит., 1960. – 94 с.
3. Методы аппроксимации сигналов экспоненциальными суммами / Л. Н. Бондаренко, А. Г. Дмитриенко, М. Г. Мясникова, Б. В. Цыпин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2011. – Спец. вып. № 3. – С. 145–155.

С. П. Каргин

Россия, Пенза, Пензенский государственный университет

О СИНТЕЗЕ СХЕМ ИЗ НЕНАДЕЖНЫХ ЭЛЕМЕНТОВ В P_4

Найдены свойства четырехзначных функций, схемы которых можно использовать для повышения надежности исходных схем, реализующих функции из P_4

Введение и постановка задачи

В работе [1] описан функционально полный в P_3 базис, в котором на компромиссной основе согласованы математические и технические (МДП-техники) требования и интересы; а также рассмотрены некоторые аспекты синтеза электронных схем в этом базисе. В работе [2] построен функционально полный в P_4 базис, реализуемый в МОП-структурах.

Таким образом, определенный интерес представляет задача исследования надежности функционирования схем в полном конечном базисе из k -значных функций ($k \geq 3$). Задача построения надежных схем в произвольном полном базисе из трехзначных функций (т. е. $k = 3$) решена в диссертации О. Ю. Барсуковой [3].

В этой статье опишем свойства четырехзначных функций, схемы которых можно использовать для повышения надежности исходных схем.

Пусть $n \in N$, а P_4 – множество всех функций четырехзначной логики, т.е. функций $f(x_1, \dots, x_n): \{0, 1, 2, 3\}^n \rightarrow \{0, 1, 2, 3\}$. Рассмотрим реализацию функций из множества P_4 схемами из ненадежных функциональных элементов в произвольном полном конечном базисе B .

Будем считать, что схема из ненадежных элементов реализует функцию $f(\tilde{x}^n) (\tilde{x}^n = (x_1, \dots, x_n))$, если при поступлении на входы схемы набора $\tilde{a}^n$ при отсутствии неисправностей в схеме на ее выходе появляется значение $f(\tilde{a}^n)$.

Предполагается, что все базисные элементы ненадежны, переходят в неисправные состояния независимо друг от друга, а сами неисправности могут быть произвольными (например, инверсными или константными).

Пусть схема S реализует функцию $f(\tilde{x}^n)$, $\tilde{a}^n$ – произвольный входной набор схемы S , $f(\tilde{a}^n) = \tau$. Обозначим через $P_{f(\tilde{a}^n) \neq \tau}(S, \tilde{a}^n)$ вероятность появления ошибки на выходе схемы S при входном наборе $\tilde{a}^n$. Ясно, что $P_{f(\tilde{a}^n) \neq \tau}(S, \tilde{a}^n) = P_{\tau+1}(S, \tilde{a}^n) + P_{\tau+2}(S, \tilde{a}^n) + P_{\tau+3}(S, \tilde{a}^n)$. Отметим, что в выражениях $\tau+1$, $\tau+2$ и $\tau+3$ сложение осуществляется по mod 4.

Например, если входной набор $\tilde{a}^n$ схемы S такой, что $f(\tilde{a}^n) = 0$, то вероятность ошибки на этом наборе равна $P_{f(\tilde{a}^n) \neq 0}(S, \tilde{a}^n) = P_1(S, \tilde{a}^n) + P_2(S, \tilde{a}^n) + P_3(S, \tilde{a}^n)$.

Ненадежностью схемы S будем называть число $P(S) = \max\{P_{f(\tilde{a}^n) \neq \tau}(S, \tilde{a}^n)\}$, где максимум берется по всем входным наборам $\tilde{a}^n$ схемы S . *Надежностью* схемы S равна $1 - P(S)$.

Пусть базисные элементы независимо друг от друга переходят в неисправные состояния, а неисправности могут быть произвольными.

Пусть $\tilde{\alpha}^m, \tilde{\beta}^m$ – некоторые наборы с компонентами из множества $\{0,1,2,3\}$ длины m ($m \geq 3$). Обозначим через $\rho(\tilde{\alpha}^m, \tilde{\beta}^m)$ число координат, в которых наборы $\tilde{\alpha}^m$ и $\tilde{\beta}^m$ различаются.

Например, если $\tilde{\alpha}^4 = (0102), \tilde{\beta}^4 = (3101)$, то $\rho(\tilde{\alpha}^4, \tilde{\beta}^4) = 2$.

Пусть функция $g(\tilde{x}^m)$ обладает следующими свойствами одновременно: существуют такие наборы $\tilde{\alpha}^m, \tilde{\beta}^m, \tilde{\gamma}^m, \tilde{\delta}^m$, что

- 1) значения $g(\tilde{\alpha}^m), g(\tilde{\beta}^m), g(\tilde{\gamma}^m), g(\tilde{\delta}^m)$ попарно различны;
- 2а) для любого набора $\tilde{\alpha}_1^m$ ($\rho(\tilde{\alpha}^m, \tilde{\alpha}_1^m) \leq 1$) верно $g(\tilde{\alpha}_1^m) = g(\tilde{\alpha}^m)$;
- 2б) для любого набора $\tilde{\beta}_1^m$ ($\rho(\tilde{\beta}^m, \tilde{\beta}_1^m) \leq 1$) верно $g(\tilde{\beta}_1^m) = g(\tilde{\beta}^m)$;
- 2в) для любого набора $\tilde{\gamma}_1^m$ ($\rho(\tilde{\gamma}^m, \tilde{\gamma}_1^m) \leq 1$) верно $g(\tilde{\gamma}_1^m) = g(\tilde{\gamma}^m)$;
- 2г) для любого набора $\tilde{\delta}_1^m$ ($\rho(\tilde{\delta}^m, \tilde{\delta}_1^m) \leq 1$) верно $g(\tilde{\delta}_1^m) = g(\tilde{\delta}^m)$.

Наборы $\tilde{\alpha}^m, \tilde{\beta}^m, \tilde{\gamma}^m, \tilde{\delta}^m$ будем называть *характеристическими* наборами функции $g(\tilde{x}^m)$.

Обозначим через G_m множество функций $g(\tilde{x}^m)$ с перечисленными свойствами. Пусть $G = \bigcup_{m=3}^{\infty} G_m$.

Пример. Нетрудно проверить, что

- 1) функция $g_1(\tilde{x}^3) = \max\{\max\{\min(x_1, x_2), \min(x_1, x_3)\}, \min(x_2, x_3)\} \in G$, а ее характеристические наборы (000), (111), (222) и (333);
- 2) функция $g_2(\tilde{x}^4) = \max\{\min(x_1, x_2), \min(x_3, x_4)\} \in G$, а (0000), (1111), (2222) и (3333) – ее характеристические наборы;
- 3) функция $g_3(\tilde{x}^3) = x_1 + x_2 + x_3 \pmod{4} \notin G$.

Теорема 1. Для числа $|G_m|$ функций в множестве G_m справедливо неравенство: $4^{4^m-8m-4} - 4^{4^m-9m-4}(27m^2 - 9m + 6) \leq |G_m| \leq 4^{4^m-8m-4}$.

Теорема 2. Предположим, что любую функцию из P_4 можно реализовать схемой с ненадежностью не больше p . Пусть схема S_g реализует функцию $g(\tilde{x}^m) \in G$ с ненадежностью $P(S_g)$, причем v^0, v^1, v^2, v^3 – вероятности ошибок схемы S_g на характеристических наборах $\tilde{\alpha}^m, \tilde{\beta}^m, \tilde{\gamma}^m, \tilde{\delta}^m$ соответственно и $g(\tilde{\alpha}^m) = 0, g(\tilde{\beta}^m) = 1, g(\tilde{\gamma}^m) = 2, g(\tilde{\delta}^m) = 3$. Тогда произвольную функцию f можно реализовать такой схемой A , что

$$P(A) \leq \max\{v^0, v^1, v^2\} + mpP(S_g) + (2^m - m - 1)p^2.$$

Таким образом, получены следующие результаты:

1. Выявлены свойства функций, схемы которых можно использовать для повышения надежности исходных схем (теорема 2).
2. Получены верхняя и нижняя оценки для числа таких функций (теорема 1).

Исследование выполнено при финансовой поддержке РФФИ, номер проекта 14-01-00273.

Список литературы

1. Виноградов, Ю. А. О синтезе трехзначных МДП-схем / Ю. А. Виноградов // Математические вопросы кибернетики : сб. ст. – М. : Наука, 1991. – Вып. 3. – С. 187–198.
2. Виноградов, Ю. А. О синтезе четырехзначных квазикомплементарных МОП схем. Математические вопросы кибернетики : сб. ст. / Ю. А. Виноградов. – М. : Наука, 1999. – Вып. 8. – С. 298–300.
3. Барсукова О. Ю. Синтез надежных схем, реализующих функции двузначной и трехзначной логик : дис. ... канд. физ.-мат. наук / Барсукова О. Ю. – Пенза, 2014. – 87 с.

Е. И. Гурин, Р. Г. Скопинцев

Россия, Пенза, Пензенский государственный университет

ПОСТРОЕНИЕ ВСТРАИВАЕМЫХ СИСТЕМ С ПРОЦЕССОРНЫМИ ЯДРАМИ НА ПЛИС ФИРМЫ XILINX

Рассматриваются вопросы создания систем с процессором Microblaze в САПР Vivado с применением нового инструмента IP Integrator для ПЛИС Artix-7, программное обеспечение реализуется в SDK. Приведены результаты проверки работоспособности созданной системы на отладочной плате Nexys-4.

В 2012 г. фирма Xilinx выпустила САПР нового поколения Vivado [1–3], которая предназначена для замены в ближайшем будущем ISE Design Suite. Традиционно разработка систем с процессорными ядрами Microblaze в ISE Design Suite и в Vivado производилась с использованием EDK. Начиная с версии 2013.3 в САПР Vivado добавлен IP Integrator [3–6], который также позволяет создавать системы с процессорными ядрами и используется в настоящей статье. В документации в качестве примера разработки систем с процессорными ядрами Microblaze обычно рассматриваются системы на отладочных платах KC705 с ПЛИС Kintex-7 [3, 5–6]. В то же время на практике неплохо себя зарекомендовали относительно недорогие и достаточно удобные в использовании отладочные платы Nexys-4 фирмы Digilent. Особенно удобно их применение в учебном процессе в ВУЗах, где обычно используются несложные проекты, и для подключения этих плат достаточно одного USB-кабеля. Настоящая статья может быть полезна преподавателям ВУЗов, а также инженерам при разработке недорогих проектов на ПЛИС Artix-7.

Структура системы с подчиненным процессорным блоком показана на рис. 1.

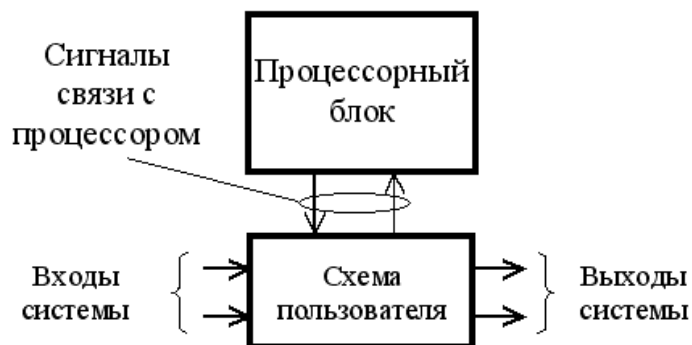


Рис. 1. Структура системы с процессорным блоком.

Внешние сигналы системы поступают в главный модуль, они могут передаваться в процессор непосредственно или после обработки – это зависит от выполняемой задачи. Внешние сигналы процессорного блока являются внутренними сигналами главного модуля. Сначала создается проект типа RTL в системе Vivado для ПЛИС XC7A100T-1CSG324C с использованием языка VHDL. Затем задается имя блока (например, mb1). Параметры процессора и локальной памяти Local Memory могут задаваться по указанию пользователя. В созданный процессорный блок надо добавить периферийные устройства, пусть в данном проекте это будут восьмиразрядный вход с переключателей (тип устройства AXI_GPIO) и последовательный интерфейс со скоростью 115200 (тип устройства AXI_Uartlite). Для проведения связей используется автоматический режим Run Connection Automation.

После этого необходимо задать ограничения, способы их задания в новой системе отличаются от предыдущих систем. Для платы Nexys-4 на сайте производителя www.digilentinc.com доступен файл ограничений Nexys4_Master.xdc, который можно взять за основу. Данный файл подключается к разрабатываемому проекту, после чего его необходимо открыть и изменить имена в этом файле в соответствии с именами в проекте. После изменений файл ограничений надо сохранить. Далее выполняется синтез и реализация проекта, и, наконец, генерация битового (загрузочного) файла.

После создания аппаратной части можно приступить к созданию программы в системе SDK. Для создания проекта задается его имя, остальные параметры не изменяются. В окне шаблонов Templates выбирается один из предлагаемых типов, например, Hello World. В созданном проекте в качестве стандартного устройства ввода-вывода автоматически назначается последовательный интерфейс axi_uartlite_0. После этого проект можно записывать в плату, в системе SDK это можно сделать из основного меню. Перед загрузкой плату необходимо подключить к компьютеру кабелем micro-USB и включить питание. Перед запуском программы на решение необходимо активизировать терминал, он должен быть настроен на работу со скоростью 115200. В результате работы программы командой print в стандартное устройство вывода будет передано сообщение Hello World, которое передается в компьютер и будет отображено в окне терминала (рис. 2).

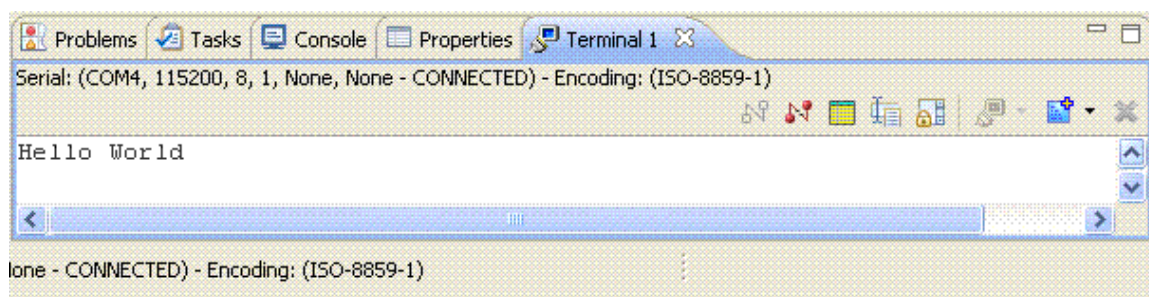


Рис. 2. Результат работы программы-шаблона Hello World

Для проверки работоспособности разработанной системы была составлена следующая простейшая программа.

```
#include "platform.h"
#include <stdio.h>
#include "xgpio_1.h"
#include "xparameters.h"
int main()
```

```

{ int i, dd=0, sw=0;
  init_platform();
  xil_printf(" \n\r vmb01 begin \n\r");
  while(1) {
    sw=Xil_In32(XPAR_GPIO_0_BASEADDR);
    if(sw != dd)
      xil_printf("\n\r sw=%d", sw);
    for(i=0; i<500000; i++)
      dd=sw;
  }
  return 0;
}

```

Интерес в приведенной программе представляют операторы ввода-вывода. С помощью функции `Xil_In32` данные считываются с переключателей на отладочной плате и присваиваются переменной `sw`. Константа `XPAR_GPIO_0_BASEADDR` определена в файле `xparameters.h`. Определение функции `Xil_In32` находится в файле `xil_io.h`, ссылка на который имеется в файле `xgpio_1.h`. Следует отметить, что для ввода данных могут использоваться и другие функции.

В следующей строке программы происходит сравнение нового значения `sw` и старого (задержанного) значения `dd`. Если они не равны (то есть состояние переключателей изменилось), то `sw` передается по последовательному интерфейсу функцией `xil_printf`. В следующих двух строках с помощью оператора `for` реализована задержка для исключения дребезга на переключателях. Приведенная программа работала при отключенной оптимизации.

Перед запуском программы переключатели `sw7...sw0` на отладочной плате Nexys-4 должны быть установлены в нижнее положение. Если активен терминал SDK, то после запуска программы в его окне будет выдана запись `vmb01 begin`.

Если переключатель `sw0` на отладочной плате Nexys-4 переключить в верхнее положение, в окне терминала SDK будет выдана запись `sw=1` (рис. 3). И далее после каждого изменения положения переключателей `sw7...sw0` в окне терминала будет выводиться их новое состояние в десятичной системе счисления.

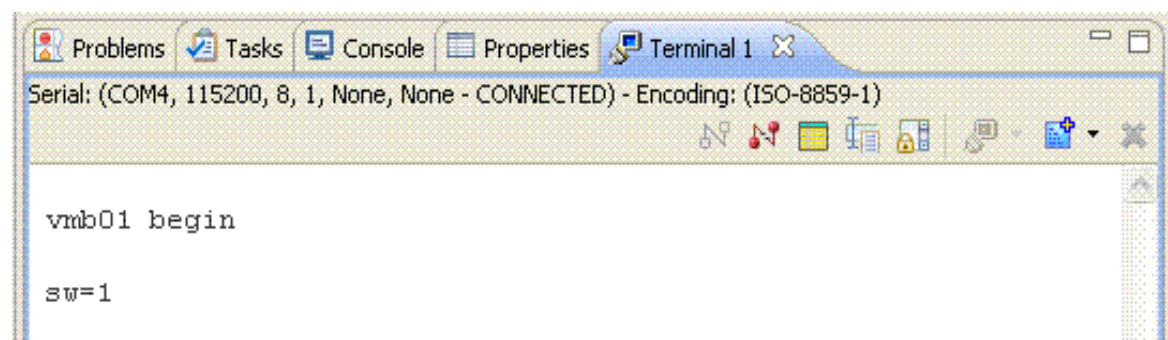


Рис. 3. Результат работы тестовой программы

В статье использовалась отладочная плата Nexys-4. Однако, рассмотренная выше последовательность действий может использоваться и во многих других похожих случаях при создании систем на основе процессора Microblaze.

Список литературы

1. Тарасов, И. Е. Маршрут проектирования ПЛИС Xilinx в САПР Vivado / И. Е. Тарасов // Компоненты и технологии. – 2012. – № 12.
2. Зотов, В. Ю. Конвертирование проектов цифровых устройств, разрабатываемых на основе ПЛИС и полностью программируемых систем на кристалле фирмы Xilinx в среде ISE Design Suite, в формат САПР Vivado Design Suite / В. Ю. Зотов // Компоненты и технологии. – 2013. – № 8–10.
3. Vivado Design Suite. User Guide. Embedded Processor Hardware Design // Xilinx, UG898 (v2013.3), October 2, 2013.
4. Тарасов, И. Е. Использование IP Integrator в САПР Vivado для ПЛИС серии 7 и UltraScale / И. Е. Тарасов // Компоненты и технологии. – 2013. – № 12.
5. Vivado Design Suite. User Guide. Designing IP Subsystems Using IP Integrator // Xilinx, UG994, October 2, 2013.
6. Vivado Design Suite Tutorial: Embedded Processor Hardware Design // UG940, October 2, 2013.

А. В. Клименко

**Россия, Москва, Московский авиационный институт
(Национальный исследовательский университет)**

АЛГОРИТМ АВТОМАТИЗИРОВАННОЙ РАЗРАБОТКИ ЭЛЕКТРОННЫХ УСТРОЙСТВ НА БАЗЕ ПОЛУЗАКАЗНЫХ ИНТЕГРАЛЬНЫХ СХЕМ НА ОСНОВЕ ПЛИС ПРОТОТИПИРОВАНИЯ

Рассматривается унифицированный алгоритм, реализуемый современными САПР электронных устройств, по осуществлению миграции проектов по маршруту ПЛИС-полузаказные интегральные схемы

С каждым годом доля ПЛИС на рынке интегральных схем (ИС) все увеличивается, что объясняется постепенным вытеснением альтернативных базисов – заказных (ASIC) и полузаказных интегральных схем (ПЗИС). Но, что вполне естественно, данная тенденция характерна не для всех отраслей промышленности. В частности, ограниченное применение ПЛИС имеют в аппаратуре специализированного назначения военного и космического применения. Это объясняется в первую очередь высокими требованиями по надежности, отказоустойчивости и радиационной стойкости, предъявляемыми к таким устройствам. ПЛИС, обладающие данными свойствами, имеют настолько высокую цену, что их использование окупается только при массовом производстве, не характерном для специализированной аппаратуры.

С другой стороны, в условиях стремительного роста технологий, использование полностью заказных ИС не приемлемо ввиду длительного цикла разработки, занимающего период времени до нескольких лет. Стоимость, основную часть которой составляют процессы создания фотошаблонов, таких устройств в условиях мелкосерийного производства еще выше, чем у аналогов на ПЛИС.

Единственным выходом из сложившейся ситуации является разработка аппаратуры на базе полужаказных ИС – в частности, базовых матричных кристаллов (БМК). По сравнению с полностью заказными ИС, в данной технологии количество необходимых фотошаблонов сокращено на порядок, что положительно сказывается на цене конечных устройств и сроках их разработки, которые составляют 1–2 месяца.

Тем не менее, производители электронной аппаратуры быстро пришли к выводу, что сроки и стоимость разработки можно снизить еще существенней, если использовать ПЛИС прототипирование. При данном подходе, образец первоначально тестируется на ПЛИС, и лишь после успешных результатов прототипирования устройство производится в базисе БМК. Такая технология дает практически 100 % гарантию того, что созданный БМК будет соответствовать требованиям разработчика, но процесс ее реализации связан с некоторыми сложностями.

Как известно, разработка электронных схем на базе ПЛИС ведется с помощью специализированных САПР, предоставляемых в большинстве своем самими производителями ПЛИС. В связи с этим, возможность осуществления миграции проектов ПЛИС в базисы полужаказных интегральных схем в основном зависит от политики фирм-производителей.

Большинство зарубежных компаний-разработчиков ПЛИС предоставляют свои конкурентные предложения в данной сфере, проанализировав которые можно выявить общие принципы реализации миграции. Унифицированный алгоритм осуществления данного процесса представлен на рис. 1.

Как видно из блок-схемы алгоритма, в качестве входных данных для миграции используется файл списка соединений (ФСС) проекта на ПЛИС (NetList). На заре развития технологии интегральных схем, программируемых пользователем, каждый производитель этого класса микросхем считал своим долгом разработать свой собственный формат ФСС, что конечно же не способствовало упрощению и удешевлению процесса разработки. Но в 1983 году представители ведущих компаний разработчиков САПР электронных устройств, таких как Daisy Systems, Mentor Graphics, Motorola, National Semiconductor, Tektronix и Texas Instruments совместно с представителями Калифорнийского университета в Беркли учредили комиссию по разработке унифицированного формата EDIF (Electronic Design Interchange Format). Появление единого формата должно было содействовать упрощению взаимодействия разработчиков электронных устройств под разные аппаратные платформы. В настоящее время EDIF имеет широкое распространение, что и обуславливает его использование в алгоритмах миграции проектов электронных устройств между различными базисами.

Создание формата EDIF было одной из первых попыток решить одну из важнейших задач, встающих перед разработчиками электронных устройств – задачу унификации взаимодействия различных САПР между собой. В настоящее время данная проблема до конца не решена, как и связанная с ней проблема унификации прототипирования ПЛИС на базе ПЛИС. Большинство производителей ПЛИС выпускают собственные имитаторы своих микросхем, и при этом не гарантируется работоспособность изделий при использовании имитаторов сторонних разработчиков. С другой стороны, помимо самостоятельной реализации процесса миграции проектов по маршруту ПЛИС-ПЛИС, разработчики электронных устройств могут воспользоваться услугами сторонних фирм, таких как Epson, Gigortix или KiaSemi, предоставляющие услуги по осуществлению данного процесса, и гарантирующие работоспособность конечных продуктов. В качестве входных данных от заказчика при данном подходе обычно требуются: ФСС или RTL-код, описывающие устройство, спецификации на IP-блоки сторонних производителей, которые может содержать система, временные ограничения, набор тестов, а также цоколевка микросхемы.

Вернувшись к блок-схеме алгоритма миграции проектов по маршруту ПЛИС-ПЗИС, рассмотрим пункт, отражающий различие в архитектурах двух технологий – пункт «Интерпретация LUT».

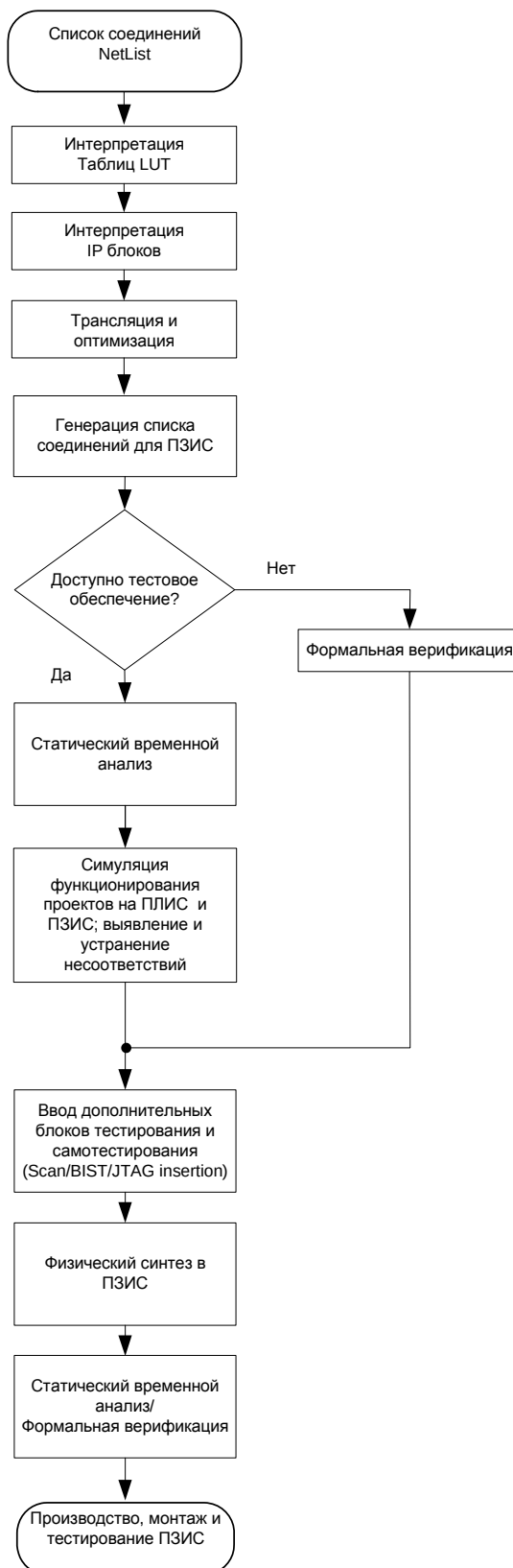


Рис. 1. Типовой процесс миграции проектов электронных устройств по маршруту ПЛИС-ПЗИС

Основа ПЛИС – базовые ячейки, как правило, содержащие элемент памяти (D-триггер), выходной мультиплексор и таблицу соответствия (LUT – Lookup table), реализующую логические функции. ФСС проекта электронного устройства на базе ПЛИС обычно содержит описание схемы устройства, основанное на определении функций, выполняемых базовыми ячейками, т.е. определяющее логические функции LUT. В ПЗИС отсутствуют подобные структуры, так как базовые ячейки микросхем данной технологии обычно содержат лишь набор нескоммутированных транзисторов. Этот факт при осуществлении миграции проектов ПЛИС в базис ПЗИС вызывает необходимость определения структур ПЗИС, функционально замещающих таблицы соответствия (LUT). Оптимальный выбор подобных структур составляет отдельную задачу.

Процесс интерпретации вышеупомянутых IP-блоков, которые можно найти в любом более или менее сложном проекте может также вызывать сложности, так как его реализация зависит от наличия спецификации от разработчика каждого блока.

В целом, основной задачей алгоритма миграции является генерация ФСС для базиса используемой ПЗИС. После осуществления этой наиболее трудоемкой части алгоритма производится добавление в схему устройства дополнительных модулей самотестирования, необходимых для осуществления проверок работоспособности конечного продукта после его производства. Также должны быть осуществлены циклы тестирования и реконфигурации сгенерированного ФСС, для достижения функциональной эквивалентности проектов на базе ПЛИС и ПЗИС.

О. А. Турчак, В. Н. Князев

Россия, Пенза, Пензенский государственный университет

МОДИФИКАЦИЯ АЛГОРИТМА КОДИРОВАНИЯ РЕЧИ MELP

Рассматриваются вопросы модификации алгоритма кодирования речи MELP с целью повышения качества синтезированной речи.

Речепреобразующие устройства (РПУ) обеспечивают сжатие речи для ее передачи или хранения в цифровом виде в диапазоне низких скоростей (1200–2400 бит/с). Высокая степень сжатия в диапазоне низких скоростей (от 50 до 100 раз) не только позволяет экономить пропускную способность каналов связи и дорогостоящие носители информации (элементы памяти), но также обеспечивает возможность повышения помехозащищенности передачи информации за счет снижения количества информационных бит, передаваемых в единицу времени. Основная цель РПУ состоит в кодировании только наиболее важных аспектов речи. Вокодеры в основном используются для воспроизведения записанных сообщений, для передачи зашифрованного сигнала по узкополосным каналам ВЧ радио, в сотовой цифровой телефонии, цифровых схемах с мультиплексированием.

Одним из распространенных алгоритмов кодирования речи является алгоритм MELP (Mixed Excited Liner Prediction). MELP-кодер основан на традиционной модели кодирования с линейным предсказанием (Linear Predictive Coding, LPC), но, кроме того, содержит пять дополнительных особенностей:

- 1) смешанное возбуждение;
- 2) использование аperiodических импульсов;
- 3) адаптивное улучшение спектрального оценивания;
- 4) рассеяние импульсов возбуждения;
- 5) моделирование амплитуд коэффициентов Фурье. [1]

Смешанное возбуждение выполнено с использованием многополосной смешивающей модели. Эффект этого смешанного возбуждения должен уменьшить гудение, обычно связываемое с LPC вокодерами, особенно в широкополосном акустическом шуме.

Когда входная речь вокализована, MELP-кодер может синтезировать использование периодических или аperiodических импульсов. Аperiodический импульс чаще используется в моменты перехода между вокализованными и невокализованными сегментами речевого сигнала. Это свойство дает декодеру воспроизвести ошибочный импульс без представления тональных звуков.

Адаптивный спектральный фильтр расширения основан на полюсах фильтра синтеза линейного предсказания. Его использование расширяет структуру форманты синтетической речи и уточняет соответствие между синтетической и естественной полосовыми формами волны. Это также делает синтетическую речь более естественной.

Рассеяние импульса выполнено с использованием фиксированного фильтра, основанного на спектральном сглаженном триангуляционном импульсе. Этот фильтр распространяет энергию возбуждения внутри периода основного тона, несколько уменьшая резкость качества синтетической речи.

Первые десять величин Фурье определяются из пиков Фурье, преобразуют сигнал остаточного предсказания. Информация в этих коэффициентах уточняет точность речи в полученной модели в наиболее важных для восприятия низких частотах. Это увеличивает качество синтетической речи особенно тогда, когда присутствует фоновый шум. [2]

В отличие от классической LPC- модели, в алгоритме MELP, в той или иной степени, воплощены достоинства и устранены недостатки двух моделей речевого кодирования LPC и MBE, которые перечислены ниже.

Многополосность представления функции смешанного возбуждения, в отличие от однополосного возбуждения, позволяет значительно точнее синтезировать все речевые звуки, как звонкие гласные и глухие фрикативные, так и смешанные.

Акустический шум не усиливается и не искажается дополнительно вокодером, т.е. более точно передается реальная акустическая обстановка. Небольшой акустический шум (на уровне менее -20 дБ) дополнительно подавляется в синтезаторе.

Наличие системы прямой коррекции ошибок, хотя и незначительно, но обеспечивает повышение качества речи при наличии ошибок в тракте передачи.

В результате объединения двух моделей возбуждения (LPC и MBE), вокодер MELP имеет существенное повышение качества синтезированной речи при сравнительно низкой вычислительной сложности.

Наряду с неоспоримыми достоинствами, алгоритм MELP имеет и свои недостатки.

Информация об амплитудах первых десяти спектральных гармоник, передаваемая на декодер, вносит очень незначительный вклад в повышение качества речи, и ощущается только при нахождении речевого источника в абсолютно тихой акустической обстановке.

Параметр усиления передается только дважды за фрейм, т.е. каждые 11,25 мс, что недостаточно для нормального воспроизведения динамичных участков речи и звука «р-р». Это вызывает «картавость» синтезированной речи.

Формат передачи кадра выглядит следующим образом.

Скорость передачи составляет 2400 бит/с. Так как все кадры содержат 54 бита, длина кадра – 22,5 мс.

Таблица 1 показывает, как 54 бита в MELP-кадре распределены среди передаваемых параметров.

Таблица 1

Распределение параметров в MELP-кадре

Параметры	Вокализованный режим	Не вокализованный режим
LSF	25	25
Величины Фурье	8	–
Усиление (2 на кадр)	8	8
Период основного тона	7	7
Полосование огласованности	4	–
Апериодический флаг	1	–
Защита от ошибок	-	13
Бит синхронизации	1	1
Итого	54	54

Опираясь на проведенный анализ и исследования различных моделей речевых кодеров, был разработан алгоритм кодирования речи для передачи на скорости 2400 бит/с, в основу которого положена MELP-модель, модифицированная с целью устранения присущих ей недостатков и повышающая качество синтезированной речи.

Алгоритм MELP был модифицирован следующим образом:

- во-первых, длительность кадра была сокращена до 20 мс, что позволило более точно передавать динамику работы речевого тракта за счет более частого обновления параметров синтезирующего фильтра, вследствие этого были пересчитаны коэффициенты окна Хэмминга;
- во-вторых, параметр усиления в модифицированном алгоритме передается три раза за кадр, т.е. каждые 6,7 мс. Это позволило не только устранить «картавость» в звучании синтезированной речи, но и в целом повысить ее разборчивость и натуральность;
- в-третьих, была исключена передача гармоник Фурье, что практически не отразилось на качестве звучания, но позволило снизить вычислительную сложность алгоритма и освободить недостающие биты для передачи параметров;
- в-четвертых, для уменьшения загрузки аппаратных ресурсов все кадры рассматриваются как вокализованные.

Модифицированный алгоритм формирует на выходе кодированные речевые параметры, приведенные в табл. 2.

Алгоритм реализован для ЦПОС (Цифровой Процессор Обработки Сигнала) семейства TMS320C55x фирмы Texas Instruments.

Реализация удовлетворяет следующим требованиям:

- функционирует в режиме реального времени;

- совместима со всеми ЦПОС в рамках семейства TMS320C55x;
- достигнуто побитовое соответствие при обработке базы речевых сигналов.

Таблица 1

Кодированные речевые параметры

Наименование параметра	Количество бит на фрейм
Синхробит	1
LSFs	25
Основной тон	6
Три усиления	$5 + 3 + 3 = 11$
Огласованность полос	4
Флаг аperiodичности	1
Общее количество бит на кадр	48

В результате реализации данного алгоритма на языке Си для TMS320C55x удалось уменьшить производительность с 300 MIPS (миллион операций в секунду) до 25 MIPS и снизить объем программного обеспечения. Полученный кодек РПУ позволяет его использовать в аппаратуре с низким энергопотреблением.

Список литературы

1. Беллами, Дж. Цифровая телефония : пер с англ. / Дж. Беллами. – М. : Эко – Трендз, 2004. – 640 с.
2. McCree, A. V. A Mixed Excitation LPC Vocoder for Low Bit Rate Speech Coding / A. V. McCree, T. P. Barnwell // IEEE Transactions on Speech and Audio Processing. – 1995. – Vol. 3, № 4. – P. 242–250.

В. В. Кутузов

Россия, Пенза, Пензенский государственный университет

СИСТЕМА ВЕРИФИКАЦИИ АЛГОРИТМОВ УПРАВЛЕНИЯ, ПРЕДСТАВЛЕННЫХ НА ЯЗЫКЕ СНДА

Описывается проблема верификации алгоритмов, формализованных с использованием аппарата событийных недетерминированных автоматов и система позволяющая упростить процесс верификации таких моделей.

Системы обработки информации обычно очень требовательны к аппаратным ресурсам. Поэтому они часто строятся на основе многопроцессорной или распределенной архитектуры. При разработке такого типа систем часто возникает проблема синхронизации данных между отдельными вычислительными блоками (ядрами, процессорами и т.д.) и поддержания когерентности этих данных. Для решения подобных задач разработано большое количество алгоритмов синхронизации, но в большинстве случаев они реализованы программно. Но программная реализация не всегда позволяет добиться нужного быстродействия, особенно в системах с интенсивным обменом и с низкими требованиями к латентности передаваемых данных, например, системах реального времени. В таком случае применяется аппаратная реализация алгоритмов синхронизации.

Разработка аппаратных средств, реализующих алгоритм синхронизации, необходимо разработать его формальное описание на одном из входных языков. Одним из самых удобных является аппарат событийных недетерминированных автоматов (СНДА) [1], обеспечивающий как формальное описание, так и структурную реализацию механизмов синхронизации.

В связи со сложностью многих алгоритмов синхронизации процесс верификации полученного формального описания является крайне важным. Существует множество систем верификации, но для любого из них требуется сначала преобразовать формальное описание с описание на входном языке верификатора.

В данной работе предлагается система позволяющая выполнить пошаговое моделирование алгоритма описанного на языке СНДА или преобразовать его формальное описание в описание на входном языке верификатора SMV[2].

Главное окно программы и вкладка ввода формального описания алгоритма в виде системы канонических уравнений (СКУ) представлены на рис. 1.

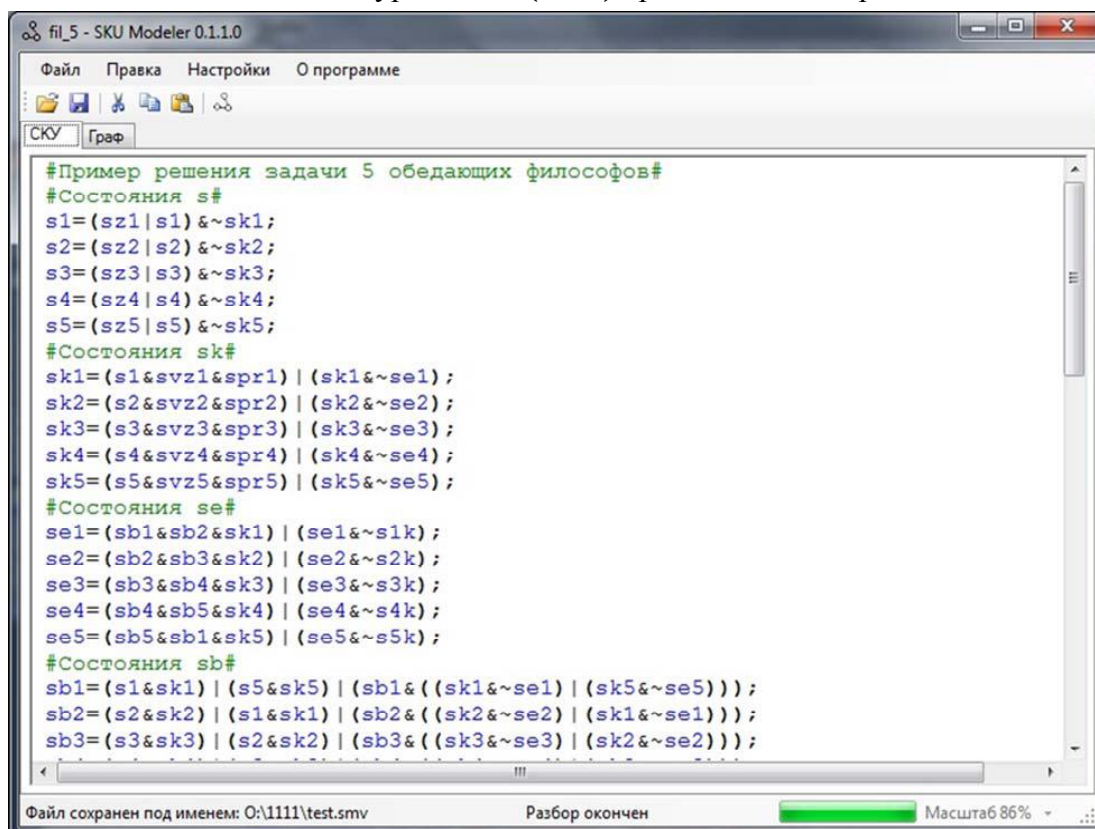


Рис. 1. Вкладка ввода исходной СКУ СНДА

Система позволяет преобразовать входную СКУ в модель, представленную в виде графа. На графе квадратными символами отображаются входные сигналы, круглыми символами состояния автомата, а цветными линиями связи между сигналами и состояниями. Так же при моделировании сигналы и состояния могут закрашиваться цветом, что говорит либо об активности сигнала, либо о том, что автомат находится в данном состоянии.

После генерации модели возможно либо пошаговое моделирование, при котором пользователь сам подает команду на переход к следующему шагу и задает входные воздействия, либо автоматическое, при котором переход к следующему

шагу переходит автоматически через заранее определенные пользователем промежутки времени, а входные воздействия берутся из таблицы сигналов, так же заранее заполненной пользователем. Пример построенного графа с результатами моделирования представлен на рис. 2.

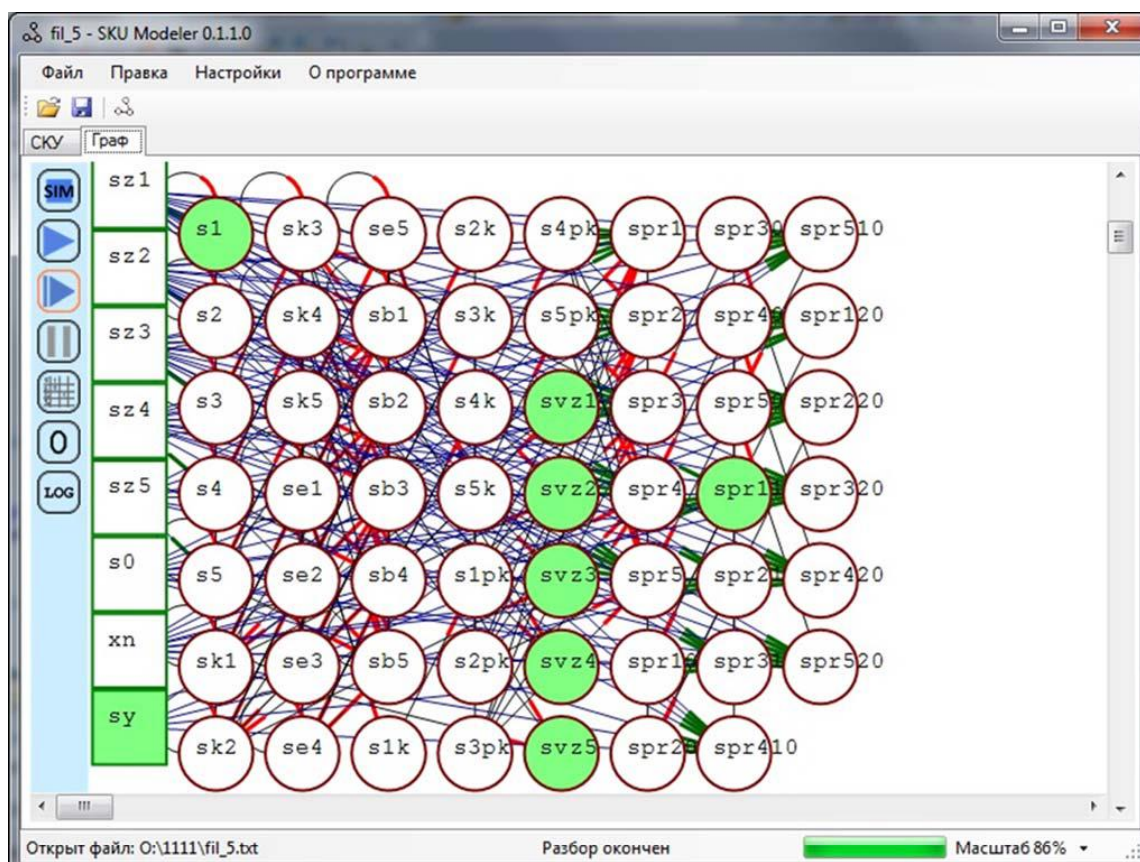


Рис. 2. Вкладка Граф

Кроме моделирования непосредственно в системе, возможна генерация описания на входном языке верификатора SMV, которое в дальнейшем может быть использовано для проверки достижимости всех состояний автомата. А так же генерация VHDL представления автомата, которое может быть использовано непосредственно при разработке аппаратной реализации алгоритма синхронизации и описание автомата на языках C++ и C# подходящее для программной реализации.

Список литературы

1. Вашкевич, Н. П. Недетерминированные автоматы в проектировании систем параллельной обработки : учеб. пособие / Н. П. Вашкевич. – Пенза : Изд-во Пенз. гос. ун-та, 2004. – 280 с.
2. Верификация автоматных программ / С. Э. Вельдер, М. А. Лукин, А. А. Шалыто, Б. Р. Яминов. – СПб. : СПбГУ ИТМО, 2011. – 242 с.

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ

В. И. Волчихин, С. А. Зинкин

Россия, Пенза, Пензенский государственный университет

ОПЕРАЦИОННАЯ СЕМАНТИКА РАЗВЕРТЫВАНИЯ МОБИЛЬНЫХ ОБЪЕКТОВ В ВИРТУАЛИЗИРОВАННЫХ СЕТЯХ ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ

Предлагается при размещении объектов в вычислительной сети использовать элементы самоорганизации, при которой программные агенты могут проверять наличие той или иной ситуации в сети и в зависимости от результатов проверки выполнять те или иные действия по изменению этой ситуации. Предлагаемый подход к проектированию виртуализированных систем хранения и обработки данных, в частности, облачных сред, грид-систем и самоорганизующихся беспроводных сетей позволяет осуществить простой переход от абстрактного проектирования к структурному проектированию.

Введение

В работе рассмотрены принципы формализации операций развертывания объектов в вычислительной сети и предложены способы визуализации возникающих ситуаций. Необходимость в таких действиях возникает при разработке облачных вычислительных сред, грид-систем, распределенных систем хранения данных, т.е. при организации таких систем, где на передний план выступают проблемы виртуализации и консолидации ресурсов хранения и обработки данных. В отличие от общепринятого определения виртуализации как абстракции вычислительных ресурсов, скрывающей от пользователя собственную реализацию, в настоящей работе поставлена задача использования виртуализации и одной из ее разновидностей – консолидации ресурсов для повышения производительности распределенной вычислительной системы за счет рационального размещения программных и информационных компонентов.

Согласно новой трактовке виртуализации ресурсов программные модули и информационные объекты могут перемещаться по сети, создавая улучшенные условия для собственного выполнения и обработки, а также для выполнения соседних процессов. Главным образом это улучшение происходит за счет естественного распараллеливания процессов обработки данных.

Наиболее показательным примером для виртуализации такого рода является обработка данных в агентно-ориентированных системах и сетях, при которой программные агенты могут перемещаться по запросу на нужный вычислительный ресурс, а другие программные агенты отвечают за хранение и перемещение обрабатываемых данных.

В таких системах трудно обеспечить рациональное развертывание программных компонент и данных, поэтому в настоящей работе предлагается при размещении объектов в сети использовать элементы самоорганизации, при которой программные агенты могут проверять наличие той или иной ситуации в сети и в зависимости от результатов проверки выполнять те или иные действия по изменению этой ситуации.

1. Абстрактный конструктор развертывания объектов

Определим абстрактный механизм, или конструктор, развертывания компонент в распределенной среде как кортеж

$$\mathbf{K} = (A, Y, S, S^*, F_s, P_s, R, M, P, F),$$

где A – множество объектов, или компонентов; Y – множество узлов сети; S – множество ситуаций, возникающих в сети при развертывании объектов, S^* – множество элементарных или начальных ситуаций, составляющих систему образующих для алгебры ситуаций; F_s – множество операций развертывания; P_s – множество отношений развертывания; R – множество правил, которые должны соблюдаться при начальных или конечных ситуациях; M – множество процедур, осуществляющих смену ситуаций развертывания (процедуры реализуются модулями сетей абстрактных машин (CeAM), определенными в работе [1]); P и F – множества исходных и результирующих отношений и функций (рассматриваемых в качестве функциональных отношений) развертывания; им соответствуют одноименные предикаты и функции.

При помощи множества S^* зафиксирована система образующих, включающая множество элементарных ситуаций (начальное размещение агентов, менеджеров, данных, результатов на узлах сети). Пусть система операций, принимающих значения в множестве ситуаций – вариантов пространственных размещений объектов, содержит перечисленные в работе [1] d -операции, реализуемые на абстрактном уровне элементами из множества F_s . Посредством суперпозиции данных операций и системы образующих S^* в множестве S порождаются новые ситуации пространственного размещения объектов. Таким образом, нами был сформулирован подход к определению d -алгебры размещения объектов, или алгебры пространственных ситуаций. Собственно d -алгебра определена в абстрактном конструкторе $\mathbf{K}$ как тройка (S, S^*, F_s) , где множество ситуаций S является носителем алгебры.

Исходные и результирующие ситуации удобно задавать не только в виде сложных выражений для описания ситуаций, но и в виде интерпретации обычной сигнатуры $\Sigma = P \cup F$, т.е. в виде отношений, связывающих объекты из множества A_s узлами из множества Y . Дальнейшие свойства абстрактного конструктора $\mathbf{K}$ поясним на примерах и последующих описаниях выполняемых операций.

2. Логико-алгебраические спецификации и операционная семантика пространственных операций

2.1. d -операции размещения мобильных объектов в сети

Рассмотрим модифицированный и расширенный перечень рекомендуемых к реализации в распределенных системах и сетях хранения и обработки данных пространственных операций (*deployment*-операций, или d -операций):

1. $A_i \odot n_j$ – размещение объекта A_i на узле n_j сети;
 $A_i \odot x$ – вариант: размещение объекта A_i на произвольном узле сети.
2. $A_i \oplus A_j$ – размещение объектов A_i и A_j на одном и том же узле;
 $\oplus(A_i, A_2, \dots, A_n)$ – обобщение данной операции на случай вектора объектов.
3. $A_i \oslash A_j$ – размещение объектов A_i и A_j на различных узлах;
 $\oslash(A_i, A_2, \dots, A_n)$ – обобщение данной операции на случай вектора объектов.
4. $A_i \circ A_j$ – произвольное размещение объектов A_i и A_j в сети;

- ①($A_1, A_2, \dots, A_n$) – обобщение данной операции на случай вектора объектов.
5. $A_i \textcircled{\text{C}} A'_i$ – создание копии A'_i объекта A_i на одном и том же узле;
 $A_i \textcircled{\text{C}}(A'_i, A''_i, \dots, A_i^{(n)})$ – обобщение операции на случай вектора копий.
6. $A_i \textcircled{\text{C}} A'_i$ – создание копии A'_i объекта A_i и размещение ее на другом узле;
 $A_i \textcircled{\text{C}}(A'_i, A''_i, \dots, A_i^{(n)})$ – обобщение операции на случай вектора копий: все созданные копии размещаются на разных узлах.
7. $A_i \textcircled{\text{C}} A'_i$ – создание копии A'_i объекта A_i и размещение ее на произвольном узле сети;
 $A_i \textcircled{\text{C}}(A'_i, A''_i, \dots, A_i^{(n)})$ – обобщение операции на случай вектора копий: все созданные копии размещаются на произвольных узлах.
8. $\textcircled{\text{D}} A_i$ – уничтожение объекта A_i ;
 $\textcircled{\text{D}}(A_1, A_2, \dots, A_n)$ – обобщение операции на случай вектора объектов.
9. $\textcircled{\text{D}} A_i$ – из множества, содержащего исходный объект A_i и все его копии, выбирается и сохраняется в сети произвольный объект, а остальные уничтожаются.
10. $\textcircled{\text{D}} A_i$ – в сети сохраняется исходный (родительский) объект, а все его копии уничтожаются.
11. $A_i, n_k \textcircled{\text{D}} n_j$ – перемещение объекта A_i с узла n_k на узел n_j ; вариант операции $A_i \textcircled{\text{D}} n_j$ – перемещение объекта A_i на узел n_j в случае, если достоверно известно, что он находится в сети.
12. $A_i \textcircled{\text{X}} A_j$ – в сети остается либо объект A_i , либо объект A_j в зависимости от выполнения условия x (операция « x -дизъюнкция»);
 $[x_1, x_2, \dots, x_n](A_1 \vee A_2 \vee \dots \vee A_n)$ – обобщение операции « x -дизъюнкция» на случай вектора объектов: при выполнении операции лишь одно из условий должно принимать значение *true*, в противном случае результатом операции будет неопределенный объект A^N .
13. $A_i \textcircled{*} A_j$ – вспомогательная операция d-конкатенации объектов: объекты A_i и A_j «сосуществуют» в одной и той же сети;
14. $\textcircled{*}(A_1, A_2, \dots, A_n)$ – обобщение данной операции на случай вектора объектов.

2.2. Использование свойств пространственно-временных отношений для перехода от функциональной архитектуры систем хранения и обработки данных к их системной архитектуре

Сложность и эффективность тех или иных структурных решений во многом зависит от временного совмещения работы функциональных модулей (ФМ). Ранее [4] нами были рассмотрено применение алгебры сценариев и пространственных ситуаций (алгебры развертывания, или размещения, объектов) к проектированию интеллектуальных систем управления внешней памятью. Темпоральные операторы, используемые в алгебре сценариев, позволяют получать новые ФМ, а операторы развертывания позволяют получать новые варианты пространственных ситуаций. Например, применяя темпоральные операторы « $\updownarrow$ » (выполнить одновременно), « $\downarrow$ » (выполнить «возможно одновременно»), « $\parallel$ » (причинно обусловленное пересечение интервалов реализации сценариев) и « $\parallel^*$ » (причинно обусловленное перекры-

тие интервала реализации одного сценария интервалом реализации другого сценария), можно порождать новые сценарии из определенных изначально сценариев x и y , составляющих систему образующих в алгебре сценариев:

$$z_1 = x \uparrow y; z_2 = x \mid y; z_3 = x \parallel y; z_4 = x \parallel^* y,$$

где x, y – некоторые сценарии i -го уровня; а z_1, z_2, z_3, z_4 – сценарии $(i+1)$ -го уровня, или составные ФМ. Решая задачу покрытия базовых функциональных структур, или сети функциональных модулей (ФМ), физическими узлами вычислительной сети, будем рассматривать сами ФМ как разновидности сценариев.

Применяя операторы развертывания « $\parallel$ » (размещение различных ФМ на одном и том же физическом узле), « $\uparrow$ » (размещение различных ФМ на различных физических узлах), « $\circ$ » (произвольно допустимое размещение ФМ), получаем в результате новые пространственные сценарии, или ситуации. Например, в результате применения операторов развертывания

$$z_1' = x \uparrow y; z_2' = x \circ y; z_3' = x \parallel y$$

над ФМ x и y формируются новые ситуации z_1', z_2' и z_3' , соответствующие непротиворечащему семантике конкретной операции развертывания сценарию размещения ФМ на сети физических узлов. Данным ситуациям соответствуют составные ФМ. Для одноименных темпоральным операциям « $\uparrow$ », « $\mid$ », « $\uparrow$ », « $\parallel$ », « $\parallel^*$ » и операциям « $\parallel$ », « $\uparrow$ », « $\circ$ » развертывания (размещения) бинарным предикатам сформулируем правила размещения базовых функциональных структур на физических узлах. Так, правило

$$(\forall \delta)(\forall y)[\mid (x, y) \subset \parallel (x, y)],$$

при выполнении которого «возможно (потенциально) одновременно» выполняемые ФМ x и y размещаются на одном и том же физическом узле и тем самым теряют возможность параллельного выполнения, соответствует критерию минимума аппаратных затрат, а правило

$$(\forall \delta)(\forall y)[\mid (x, y) \subset \uparrow (x, y)],$$

при реализации которого «возможно одновременно» выполняемые ФМ x и y размещаются на различных физических узлах (тем самым обеспечивается возможность их параллельного выполнения), соответствует критерию возможной максимальной производительности за счет параллельного выполнения функциональных модулей.

В проектируемых системах хранения и обработки данных обычно учитываются ограничения объема памяти физических узлов и реальные выгоды от параллельного выполнения некоторых из них невелики, поэтому для многих проектируемых систем нередко достаточно выполнения следующего более общего правила

$$\begin{aligned} &(\forall \delta)(\forall y)[\mid (x, y) \subset \circ (x, y)] \sim \\ &\sim (\exists \delta)(\exists y)[\mid (x, y) \supset \parallel (x, y)] \& (\exists \delta)(\exists y)[\mid (x, y) \supset \uparrow (x, y)] \vee \\ &\vee (\forall \delta)(\forall y)[\mid (x, y) \supset \parallel (x, y)] \vee (\forall \delta)(\forall y)[\mid (x, y) \supset \uparrow (x, y)], \end{aligned}$$

при реализации которого «возможно одновременно» выполняемые ФМ размещаются либо на одном и том же физическом узле, либо на разных узлах.

Аналогично, для темпорального отношения « $\uparrow$ » в реализованных системах хранения и обработки данных может соблюдаться правило

$$\begin{aligned}
& (\forall \delta)(\forall y)[\uparrow(x, y) \supset \textcircled{1}(x, y)] \sim \\
& \sim (\exists \delta)(\exists y)[\uparrow(x, y) \supset \textcircled{11}(x, y)] \& (\exists \delta)(\exists y)[\uparrow(x, y) \supset \textcircled{4}(x, y)] \vee \\
& \vee (\forall \delta)(\forall y)[\uparrow(x, y) \supset \textcircled{11}(x, y)] \vee (\forall \delta)(\forall y)[\uparrow(x, y) \supset \textcircled{4}(x, y)],
\end{aligned}$$

означающее возможность размещения выполняемых разновременно ФМ как на одних и тех же, так и на различных физических узлах.

Для темпоральных отношений «||*», «||» и «↑» размещение ФМ x и y связано с выполнением следующих правил:

$$\begin{aligned}
& (\forall \delta)(\forall y)[||^*(x, y) \supset \textcircled{4}(x, y)]; \\
& (\forall \delta)(\forall y)[|| (x, y) \supset \textcircled{4}(x, y)]; \\
& (\forall \delta)(\forall y)[\uparrow(x, y) \supset \textcircled{1}(x, y)] \sim \\
& \sim (\exists \delta)(\exists y)[\uparrow(x, y) \supset \textcircled{4}(x, y)] \& (\exists \delta)(\exists y)[\uparrow(x, y) \supset \textcircled{11}(x, y)] \vee \\
& \vee (\forall \delta)(\forall y)[\uparrow(x, y) \supset \textcircled{4}(x, y)] \vee (\forall \delta)(\forall y)[\uparrow(x, y) \supset \textcircled{11}(x, y)].
\end{aligned}$$

Проиллюстрируем использование свойств пространственно-временных отношений для перехода от функциональной архитектуры систем хранения и обработки данных к системной архитектуре. В работе [4] приведен пример СеАМ-реализации следующего сценария обработки данных в сетевой системе хранения данных:

$$F_0 = F_1 \uparrow ((F_2 \downarrow F_3) | (F_4 \uparrow [\alpha] (F_5 \vee A^E) \uparrow F_6)) \uparrow (F_7 ||^* F_8) \uparrow F_k.$$

Не вдаваясь в детали выполнения указанных операций, опишем на основе приведенных выше правил ситуацию развертывания используемых ФМ в сети физических узлов:

$$F_0 = F_1 \textcircled{1} ((F_2 \textcircled{11} F_3) \textcircled{4} ((\textcircled{11}(F_4, F_5, F_6))) \textcircled{1} (F_7 \textcircled{4} F_8) \textcircled{1} F_k).$$

Данное выражение соответствует ситуации развертывания, обеспечивающей реализацию параллелизма при повышенных аппаратных затратах. Если же при реализации необходима некоторая экономия физических узлов в ущерб производительности, то возможно модифицировать последнее выражение следующим образом:

$$F'_0 = (F_1 \textcircled{11} ((F_2 \textcircled{11} F_3) \textcircled{11} ((\textcircled{11}(F_4, F_5, F_6)))) \textcircled{1} (F_7 \textcircled{4} F_8) \textcircled{1} F_k).$$

Заключение

Предложены принципы реализации методов и средств проектирования распределенных систем хранения и обработки данных, основанные на использовании непосредственно исполняемых формальных спецификаций и обеспечивающих построение масштабируемых систем с автоматизацией процессов реконфигурации структуры и миграции данных. Эти спецификации адекватно описывают процессы управления путями доступа к ресурсам, поиска и выборки данных, копирования и репликации данных.

Статья подготовлена в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития НТК России на 2014–2020 г.» (Соглашение № 14.574.21.0045 от 19.06.2014).

Список литературы

1. Волчихин, В. И. Самомодификация топологии и развертывания объектов в виртуализированных сетях хранения и обработки данных. I: Темпоральные и пространственные операции / В. И. Волчихин, С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2014. – № 3.
2. Колмогоров, А. Н. Математическая логика / А. Н. Колмогоров, А. Г. Драгалин. – М. : Едиториал УРСС, 2005. – 240 с.
3. Многоуровневое структурное проектирование программ. Теоретические основы, инструментарий / Е. Л. Ющенко, Г. Е. Цейтлин, В. П. Грицай, Т. К. Терзян. – М. : Финансы и статистика, 1989. – 208 с.
4. Зинкин, С. А. Функционально-структурная реализация интеллектуальных систем управления внешней памятью ЭВМ и систем / С. А. Зинкин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2008. – № 1. – С. 14–24.

А. Ю. Якупов, А. А. Малкин, Ю. С. Колесникова

**Россия, Москва, Московский авиационный институт
(Национальный исследовательский университет)**

АППАРАТНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА СЖАТИЯ ИЗОБРАЖЕНИЙ БЕЗ ПОТЕРЬ С ЭФФЕКТИВНЫМ ИСПОЛЬЗОВАНИЕМ ПАМЯТИ, ОРИЕНТИРОВАННАЯ НА КОСМИЧЕСКОЕ ПРИМЕНЕНИЕ

Представлено описание аппаратной реализации сжатия на основе алгоритмов дельта-кодирования и кодирования длин серий с использованием малого количества памяти и ресурсов базового матричного кристалла (БМК) или программируемой логической интегральной схемы (ПЛИС). Рассмотрен формат, обеспечивающий надежную доставку сжатого изображения по каналу КА-Земля.

При разработке аппаратного сжатия изображений на БМК или ПЛИС для космического применения стоит задача уменьшения объема используемых ресурсов БМК (ПЛИС) и памяти, а также обеспечения надежной доставки данных сжатого изображения по каналу КА-Земля.

Рекомендованный стандарт сжатия изображений консультативного комитета по космическим информационным системам (CCSDS) описывает практические рекомендации реализации сжатия без потерь на борту космического аппарата на основе целочисленного дискретного вейвлет-преобразования. Согласно стандарту CCSDS для сжатия целого кадра требуется объем памяти, эквивалентный размеру кадра. Специфика и сложность алгоритма CCSDS требует большого объема ресурсов БМК (ПЛИС). Кроме того, большая латентность выдачи данных может служить препятствием для применения данного алгоритма в системах реального времени.

В целях минимизации объема используемой памяти и ресурсов БМК был разработан алгоритм сжатия на основе дельта-кодирования и кодирования длин серий. Для оценки степени сжатия в качестве входных данных использовалось монохромное изображение поверхности Земли с орбитального спутника. Степень сжатия данного изображения разработанным алгоритмом с учетом служебной информации обеспечения надежной доставки данных составляет в среднем 2,65, что сопостави-

мо со степенью сжатия алгоритмом WinRAR согласно табл. 1. В настоящее время проводится сравнение с алгоритмом сжатия стандарта CCSDS.

В целях обеспечения надежности доставки данных по каналу КА-Земля был разработан пакетный формат представления сжатых данных. Распространение ошибки ограничивается областью пакета сжатых данных. Потеря пакета не влияет на декодирование других пакетов. Динамическое изменение длины пакета позволяет варьировать соотношение степени сжатия и количества теряемых данных при возникновении ошибки в пакете.

Таблица 1

**Степень сжатия монохромного изображения поверхности Земли
с орбитального спутника**

Метод сжатия	Степень сжатия
WinRAR	2,84
ZIP	2,179
Хаффман	2,17
Хаффман + кодирование длин серий	2,42
Дельта-кодирование + кодирование длин серий	2,65

В целях ограничения размера выходной информации из-за низкой степени сжатия некоторого сегмента изображения были введены дополнительные возможности конфигурирования схемы сжатия, позволяющие ограничить размер выходной информации с применением адаптивного усечения векторов пикселей в зависимости от требований к пропускной способности

Входная кадровая информация разбивается на блоки данных по M отсчетов ($A_1, \dots, A_M$, где A_1 – значение первого отсчета блока данных, A_M – последнего). Блоки данных в результате обработки преобразуются в сегменты сжатых данных.

Существует три типа сегментов сжатых данных (далее сегменты). Каждый сегмент имеет заголовок сегмента и тело сегмента. Размерность заголовка сегмента постоянна и равна РЗС.

Сегмент 1-го типа – сегмент повторяющихся блоков данных. Размерность тела сегмента постоянна и равна РТС1. В тело сегмента записывается количество блоков данных (КТС1), в которых значения отсчетов равны между собой ($A_1 = \dots = A_M$) и опорному значению сегмента A_0 . Если сегмент является первым, то в качестве опорного значения A_0 берется значение 0, иначе берется значение A_M блока данных предыдущего сегмента сжатых данных. Упаковка данных внутри сегмента 1-го типа приведена на рисунке 1.

Сегмент 2-го типа – сегмент сжатого блока данных. Тело сегмента содержит M значений разностей $P_1, \dots, P_M$ со знаками $3_1, \dots, 3_M$, где $P_k = |A_k - A_{k-1}|$, $3_k = 1$, если $A_{k-1} < A_k$, иначе $3_k = 0$ ($k = 1, \dots, M$). Если сегмент является первым, то в качестве опорного значения A_0 берется значение 0, иначе берется значение A_M блока данных предыдущего сегмента сжатых данных. Внутри тела одного сегмента сжатых данных все разности имеют одну и ту же разрядность, которая равна разрядности максимальной разности P_k в этом сегменте. Разрядность разностей равна PP , мин, если максимальная разрядность разностей меньше PP , мин. В случае если максимальная разрядность разностей меньше или равна PP , макс, то используется сегмент 2-го типа, иначе используется сегмент 3-го типа. Упаковка данных внутри сегмента 2-го типа приведена на рис. 1.

Сегмент 3-го типа – сегмент несжатого блока данных. Тело сегмента содержит M значений отсчетов блока входных данных без изменения ($A_1, \dots, A_M$). Упаковка данных внутри сегмента 3-го типа приведена на рис. 1.

Заголовок пакета начинается с поля-разделителя пакетов, которое служит для обозначения начала пакета. Разделитель пакетов состоит из одного байта, значение которого равно 255. Далее в заголовке следует значение порядкового номера пакета (нумерация пакетов начинается с 1). За ним следует информация о положении в кадре отсчета А1 первого блока данных в первом сегменте этого пакета: номер строки Нстр и номер отсчета в строке Нотсч. Заголовок пакета оканчивается опорным значением пакета Апакет. В качестве Апакет берется значение отсчета А1 первого блока данных в первом сегменте этого пакета. Упаковка данных внутри пакета приведена на рис. 1.



Рис. 1. Упаковка данных внутри сегментов и пакета

В случае если первый сегмент сжатых данных в пакете является сегментом 1-го типа, то опорное значение сегмента равно Апакет, а Нстр и Нотсч определяют положение отсчета А1 первого блока данных сегмента.

В случае если первый сегмент сжатых данных в пакете является сегментом 2-го типа, то $З_1$ и P_1 в этом сегменте значения не имеют, так как были вычислены относительно данных последнего сегмента предыдущего пакета. Значение A_1 первого блока данных в пакете равно Апакет, а значения $A_2, \dots, A_M$ вычисляются из $З_2, \dots, З_M$ и $P_2, \dots, P_M$ данного сегмента относительно опорного значения Апакет.

В случае если первый сегмент сжатых данных в пакете является сегментом 3-го типа, то Нстр и Нотсч определяют положение отсчета А1 этого сегмента данных и его значение должно совпадать с Апакет.

Сегменты сжатых данных упаковываются в область полезной информации пакета без пропусков и выравнивания, т.е. первый бит заголовка очередного сегмента сжатых данных добавляется вплотную за последним битом предыдущего сегмента сжатых данных. Если в текущем пакете в области полезной информации недостаточно места для нового сегмента сжатых данных, то незаполненные биты области полезной информации заполняются нулевыми значениями, а новый сегмент сжатых данных передается в следующем пакете. Размещение сегментов сжатых данных в области полезной информации показано на рис. 2. В области полезной информации пакета ПКТб недостаточно места для сегмента СЕГа+1, поэтому он помещен в область полезной информации пакета ПКТб+1.

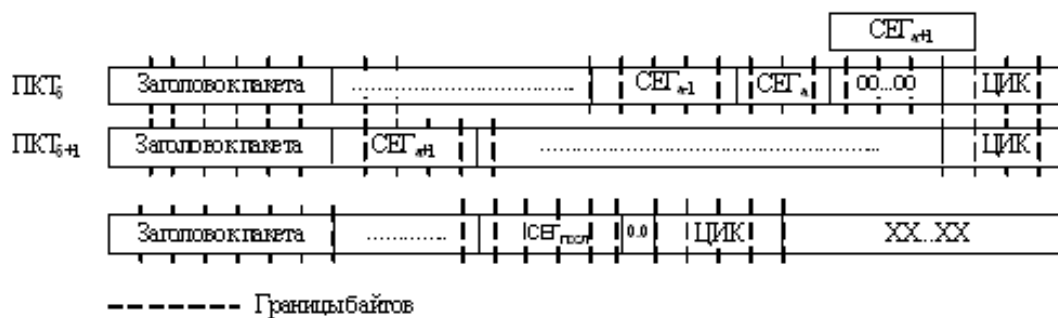


Рис. 2. Размещение сегментов и ЦИК в пакете

Циклический избыточный код (ЦИК) пакета вычисляется для заголовка пакета и области полезной информации и хранится в последних четырех байтах пакета. В случае если обработаны все блоки входных данных кадра, то ЦИК помещается непосредственно после последнего сегмента сжатых данных в область полезной информации, при этом первый бит циклического избыточного кода выравнивается по границе слова сжатой информации внутри области полезной информации. Незаполненные биты между циклическим избыточным кодом и последним сегментом сжатых данных заполняются нулевыми значениями. Размещение ЦИК после последнего сегмента сжатых данных приведено на рис. 2, где СЕТ_{посл} – последний сегмент сжатых данных, X – незначащие данные.

При разрядности пикселя монохромного изображения поверхности Земли, равного 10 бит, при $M = 8$, $PЗС = 3$, $PТC1 = 8$ бит, $РР$, мин = 1 бит, $РР$, макс = 6 бит достигается наиболее высокая степень сжатия данным алгоритмом (см. табл. 1). Аппаратная реализация данного алгоритма обеспечивает потоковое сжатие изображения без потерь с низким показателем латентности (~14 тактов) на максимальной частоте 78 МГц. Объем требуемого входного буфера – 8 входных отсчетов. Объем занимаемых ресурсов аппаратной реализации данного алгоритма в ПЛИС ProAsic 3 АРА1000 составляет 6,72 %.

А. С. Башев, А. Ю. Афонин

Россия, Пенза, Пензенский государственный университет

МЕТОДЫ ХРАНЕНИЯ И СОЗДАНИЯ СМЕШАННОГО ЛАНДШАФТА

Приведен обзор методов создания реалистичного смешанного ландшафта и хранения полученных данных о местности.

При разработке моделей, реализующих взаимодействующей с окружающей средой, возникнет задача передачи разнообразия окружающего мира. При этом полученный ландшафт должен быть реалистично правдоподобным и не выходить за рамки выбранного биома или множества биомов.

Существует подход основанный на возможности определенным образом отцифровать реально существующий участок местности, используя карты, фотографии. Но он крайне трудоемок и ограничен исходными данными.

Альтернативой создания смешанной местности является использование алгоритмов создающих ландшафт псевдослучайным образом.

Существует несколько методов хранения информации о ландшафте:

- карта высот;
- иррегулярная сетка вершин;
- блочное хранение карты ландшафта.

В случае использования карты высот данные ландшафта представляются в виде 3-мерного массива X, Y, Z . Обычно карту высот хранят в графическом монохромном файле. Это позволяет легко вносить изменения и просматривать данные в любом графическом редакторе. Координаты точки X, Y задается положением конкретного пикселя на изображении. Высота Z представляется яркостью пикселя – чем ярче, тем выше находится точка. В случае использования большего количества цветов градация высот может расширяться от базовых 256 значений для монохромного представления.

С помощью этого способа можно представить достаточно обширные пространства с одним существенным недостатком – избыточностью данных. Для описания простой плоскости требуется множество точек, хотя можно бы было обойтись тремя (исходя из уравнения плоскости).

В случае использования цветного изображения для хранения высоты используется только несколько каналов. Остальные каналы доступны для хранения дополнительной информации о ландшафте.

Так как вершинные точки расположены регулярно, можно более реалистично проводить динамическое освещение без дополнительных модификаций.

Еще один способ представления ландшафтных данных – иррегулярная сетка вершин и связей между ними. Иррегулярная сетка хранятся в виде трехмерной модели. Это дает основной выигрыш по сравнению с картами высот, поскольку в данном случае необходимо хранить только значения высот каждой вершины и связи этих вершин. Недостаток подхода заключается в сложности пересчета при динамическом освещении, поскольку вершины расположены достаточно удаленно и неравномерно друг от друга.

Существует альтернативный вариант использования карты весов. Вместо высот в ней хранятся индексы блоков ландшафта. Блоки могут быть регулярными и иррегулярными. Этот способ имеет проблему со стыковкой различных блоков, но позволяет создавать огромные ландшафты.

Кроме самих ландшафтов в таких блоках можно хранить и информацию о зданиях, строениях, растениях и т.д. Также появляется возможность создания нескольких вариантов одного и того же сегмента с разной степени детализации.

Наибольшее распространение получило хранение данных ландшафта в виде карты высот. Большинство алгоритмов используют многомерные массивы в качестве структур данных, поэтому это удобно в использовании и просто в реализации.

Ландшафт должен иметь плавность и некую зависимость на карте высот. Для решения этой задачи используется шум Перлина. Данный алгоритм применяется в компьютерной графике для увеличения реализма или графической сложности поверхности геометрических объектов.

Шум Перлина – это градиентный шум, состоящий из набора псевдослучайных единичных векторов в некоторых точках пространства и интерполированных функцией сглаживания между ними [1]. Для генерации шума Перлина в одномерном пространстве необходимо для каждой точки этого пространства вычислить значение шумовой функции, используя направление градиента в указанной точке.

В отличие от растровых текстур, шум Перлина является процедурной текстурой, и поэтому он не занимает память. Но, вместе с тем, исполнение алгоритма требует неких вычислительных ресурсов. Текстура, созданная с помощью этого алго-

ритма, имеет 2 цвета – белый и черный. Соответственно, цвет можно использовать, как значение высоты. Каждую добавленную последующую шумовую функцию называют октавой. В определенный момент октава приобретает слишком высокую частоту, чтобы быть отображаемой. Некоторые реализации шума Перлина автоматически складываются из шумовых функций до тех пор, пока пределы разрешения экрана (или другой среды) не будут достигнуты.

Иногда амплитуда становится слишком мала, чтобы оказывать влияние на результат, в этом случае следует прекратить добавление октав. Когда именно это произойдет, зависит от уровня стойкости функции Перлина. Меньшее число октав в купе с линейной интерполяцией сильно убыстряет алгоритм, но больше число октав дает более детализированный результат.

Можно создать шум функции Перлина с различными характеристиками и с использованием других частот и амплитуд на каждом шагу. Например, чтобы создать гладкие холмы, можно использовать функции шума Перлина с большими амплитудами на низких частотах, и очень малой амплитудой на высоких частотах. Или для пологой, но очень скалистой плоскости выбрать малые амплитуды для низких частот.

Шум Перлина служит чаще всего для пост-обработки результатов других алгоритмов или шумовых функций, поскольку сам по себе выдает слишком «случайно выглядящие» результаты.

Алгоритм «diamond-square» – достаточно простой, распространенный и рекурсивный по сути (в случае только «square» варианта) метод создания смешанного ландшафта [2]:

- 1) задание на создаваемой местности квадрата с 4 вершинами;
- 2) зададим случайных значений этим вершинам;
- 3) шаг «square»:
 - разделение квадрата на 4 под-квадрата по серединам его сторон. Значение полученных вершин будет средним значением между соседними точками;
 - вычисление срединной точки каждого квадрата таким же усреднением, но с добавлением при этом некоторой случайной величины (чаще всего зависимой от размеров квадрата);
 - в случае «square» варианта алгоритма вычисление продолжается для каждого полученного квадрата.
- 4) шаг «diamond»:
 - вычисление высоты пары точек «слева» и «справа». Затем проводится усреднение значения высот лежащих на серединах сторон от полученной на шаге «square» центральной точки и прибавление случайного отклонения;
- 5) повторение шагов 3–4 до необходимой детализации ландшафта.

Важно заметить, что обсчет нужно вести «слоями», сначала для всех квадратов выполнить шаг «square» – затем для всех ромбов выполнить шаг «diamond» – и перейти к меньшим квадратам.

Кроме необходимости использовать обход в ширину вместо обхода в глубину, есть еще одна тонкость – ситуация на краях ландшафта.

Дело в том, что на этапе «diamond» алгоритм использует высоту точек, которых находятся за пределами текущего квадрата и, возможно, всей карты. Простым вариантом решения данной проблемы будет принятие высоты подобных точек за константу.

Данный алгоритм хорош тем, что значения можно перемножать и складывать различные фрагменты фрактального ландшафта с разными коэффициентами и тем

самым, добиваться отличных результатов. К тому же его простота позволяет создавать огромные пространства с минимальной затратой ресурсов.

Существует и более простой способ создания ландшафтов с использованием холмового алгоритма [3]:

- 1) создание двумерного массива и инициализация его нулевым уровнем;
- 2) определение случайной точки на ландшафте или около его границ (за границами), а также задание случайного радиуса в заранее заданных пределах. В выбранной точке создается холм заданного радиуса;
- 3) возвращение ко второму шагу и так далее до выбранного количества шагов. От него потом будет зависеть внешний вид ландшафта;
- 4) проведение нормализации ландшафта
- 5) проведение сглаживания ландшафта.

Фактически холм – половина шара, чем больше радиус – тем выше и больше холм.

$$z = r^2 - ((x_2 - x_1)^2 + (y_2 - y_1))^2,$$

здесь (x_1, y_1) заданная точка; r – выбранный радиус; (x_2, y_2) – высота холма.

В случае необходимости можно достаточно просто реализовать озерные ландшафты, создав простой, достаточно холмистый ландшафт и переместив уровень воды вверх или вниз.

Данный алгоритм далеко не универсален и дает в целом однотипные результаты, которые в любом случае будут представлять набор холмов, что выглядит не слишком реалистично.

Создание ландшафта по алгоритму использующего диаграммы Вороного начинается со случайного бросания точек на карту. Затем по этим точкам строится диаграмма Вороного (и, соответственно, триангуляция Делоне), а на ней выполняется несколько итераций релаксации Ллойда, чтобы избавиться от слишком мелких полигонов. Далее случайным образом выбираются полигоны, подлежащие заполнению водой, а высота точек, принадлежащих остальным полигонам, устанавливается равной кратчайшему расстоянию до уровня воды. Этот алгоритм является самым сложным и ресурсоемким, но дает очень реалистичные и разнообразные результаты.

Список литературы

1. Perlin Noise. – URL: http://freespace.virgin.net/hugo.elias/models/m_perlin.htm
2. Алгоритм «diamond-square» для построения фрактальных ландшафтов. – URL: <http://habrahabr.ru/post/111538/>
3. Генерация трехмерных ландшафтов. – URL: <http://www.ixbt.com/video/3dterrains-generation.shtml>

ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ АЛГОРИТМОВ ДЕКОДИРОВАНИЯ КАСКАДНОЙ СИСТЕМЫ КОДИРОВАНИЯ НА ОСНОВЕ СВЕРТОЧНЫХ КОДОВ

Исследуются корректирующие свойства каскадной системы помехоустойчивого кодирования и дается оценка помехоустойчивости алгоритмов декодирования каскадной системы кодирования на основе сверточных кодов.

Подавляющее число современных систем связи работает в цифровом виде. Из-за наличия помех в каналах связи сбой при приеме любого элемента вызывает искажение цифровых данных, что может привести, особенно в космических системах связи, к катастрофическим последствиям.

Высокоэффективным средством борьбы с помехами в цифровых системах связи является применение помехоустойчивого кодирования, основанного на введении искусственной избыточности в передаваемое сообщение, что приводит к расширению используемой полосы частот и уменьшению информационной скорости передачи.

Данная работа направлена на исследование ошибкообнаруживающих и корректирующих свойств каскадной системы помехоустойчивого кодирования, а также в ней дается оценка помехоустойчивости алгоритмов декодирования каскадной системы кодирования на основе сверточных кодов. За основу системы кодирования берется каскадная система. Каскадный принцип построения системы помехоустойчивого кодирования (СПК) заключается в использовании внешнего и внутреннего кодов. На рис. 1 представлена каскадная СПК, где внешний кодер по отношению канала передачи или хранения данных – осуществляет кодирование каждого из n_2 символов своим кодом (n_1, k_1) . Совокупность двух кодеров называют расширенным кодером с результирующей длиной $n_1 n_2$, а совокупность двух декодеров – расширенным декодером [1, 2, 6].



Рис. 1. Каскадная система кодирования

В данной СПК в качестве внешнего кодера используется кодер Рида – Соломона, а в качестве внутренних – кодеры сверточных кодов. Для построения внешнего кодера был выбран код Рида-Соломона исходя из того, что кодер и декодер, основанные на алгоритме работы кода Рида-Соломона, позволяют обнаруживать двукратные и исправлять однократные ошибки, возникающие в процессе записи информации. Также веским основанием для использования кода Рида-Соломона явился тот факт, что при фиксированных длине кода (n) и количестве информационных разрядов (k) не существует кода, у которого минимальное расстояние больше, чем у кода Рида-Соломона. Коды Рида-Соломона всегда оказываются короче

всех других кодов над тем же алфавитом. Для построения внутреннего кодера был выбран класс сверточных кодов исходя из того, что сверточное кодирование, применяемое вместе с декодированием Витерби, является в настоящее время одним из наиболее широко используемых методов исправления ошибок. Причина этого состоит в простоте реализации, а также в относительно большом достигаемом при этом выигрыше от кодирования.

В данной СПК сверточные коды используются в качестве канальных кодов, которые являются нелинейными для данной системы. Признаки тактовой и кодовой синхронизации, диспаратность кодовых слов, характеризующая уровень постоянной составляющей, функция спектральной плотности мощности сигнала, в основном, обеспечиваются путем введения избыточности в исходные комбинации символов, а также за счет преобразования канальных сигналов (биимпульсное кодирование) и относительного кодирования (смена правил кодирования в зависимости от структуры исходной последовательности символов) [1, 2, 5].

Избыточность и структура кодов данного типа может использоваться и для обнаружения ошибок. При определении корректирующей способности кодов учитывалось то обстоятельство, что коды данного типа имеют небольшую длину, поэтому для их декодирования можно применять табличные методы, реализуемые с помощью ПЗУ. Эти методы полнее используют корректирующую способность кодов. Отметим, что коды такого типа целесообразно конструировать с помощью матриц Адамара.

Согласно известной классификации сверточные коды можно разделить на систематические и несистематические.

Систематические коды позволяют получить на приемной стороне оценку информационных символов, не производя декодирования или какой-либо иной обработки принимаемых символов. В кодовых несистематических последовательностях информация непосредственно не содержится, они должны строиться так, чтобы при отсутствии ошибок ее можно было легко восстановить в декодере.

Достоинством несистематических сверточных кодов является то, что их минимальное свободное расстояние выше, чем у систематических при прочих равных условиях. Это объясняется тем, что систематический сверточный код можно получить из соответствующего несистематического, исключив один из сумматоров по модулю 2, что и приводит к уменьшению минимального свободного расстояния кода.

Были получены, приведенные в табл. 1, максимальные значения минимального свободного расстояния для систематических и несистематических сверточных кодов со скоростью $R = 1/2$ при различной длине m сдвигающего регистра.

Таблица 1

Максимальные значения минимального свободного расстояния

m	Несистематический код – $d_{\min}$	Систематический код – $d_{\min}$
2	3	3
3	5	4
4	6	4
5	7	5

Для известной схемной реализации декодеров было проведено сравнение алгоритмов декодирования Евклида и Витерби (табл. 2). Выбор этих схем декодирования обуславливается достаточно простой их реализацией [3, 4].

Сравнение алгоритмов декодирования

Характеристика кода	Алгоритм Евклида	Алгоритм Витерби
Длина кода	15	72
Количество информационных разрядов	9	60
Количество контрольных разрядов	6	12
Избыточность	0,67	0,167
Скорость	0,6	0,83
Мин. расстояние Хемминга	7	13

В ходе моделирования каскадной системы помехоустойчивого кодирования в среде Matlab Release 13 было получено, что декодер, построенный по алгоритму Евклида, обнаруживает двухразрядные и исправляет одnorазрядные ошибки, а декодер, построенный по алгоритму Витерби, обнаруживает и исправляет двухразрядные ошибки.

Сравнивая декодеры, построенные по алгоритму Евклида и по алгоритму Витерби можно сделать вывод, что алгоритм декодирования Витерби является более оптимальным при разработке внешнего декодера, построенного на основе кода Рида-Соломона (120, 104). Декодер Витерби является более помехоустойчивым, быстрым, а также обладает более высокой корректирующей способностью.

Проанализировав полученные результаты, можно сделать вывод, что для существенного улучшения помехоустойчивости при использовании сверточных кодов необходимо увеличивать скорость передачи символов, а, следовательно, и ширину полосы пропускания, например, в 2 раза при относительной скорости передачи кода $1/2$ или в $4/3$ раза при относительной скорости $3/4$. Таким образом, применение сверточных кодов оказывается особенно выгодным в спутниковых системах связи, энергетический потенциал которых ограничивается мощностью бортового ретранслятора, т. е. в каналах, где определяющим фактором является ограничение мощности, а не полосы частот. В системах с ограниченной энергетикой кодирование позволяет уменьшить необходимое отношение сигнал – шум, оптимальным образом распределить мощность ретранслятора между каналами и увеличить число каналов.

Тема данной работы может получить свое дальнейшее развитие в области исследования корректирующих свойств кодов и их реализации, так как теория кодирования позволяет создать устройства с еще более мощными корректирующими способностями.

Список литературы

1. Савельев, Б. А. Повышение качества информации в средствах хранения информационных систем / Б. А. Савельев // Качество информации : тез. докл. к Всесоюз. науч.-техн. конф. – М. : МИИТ, 1988. – С. 36–40.
2. Савельев, Б. А. Коррекция ошибок биимпульсными кодами / Б. А. Савельев // Проектирование и эксплуатация информационно-вычислительных комплексов. – Саратов : Изд-во Саратов. гос. ун-та, 1986. – С. 72–75.
3. Кларк, Дж. Кодирование с исправлением ошибок в системах цифровой связи : пер. с англ. / Дж. Кларк, Дж. Кейн. – М. : Радио и связь, 1987. – 392 с.
4. Витерби, А. Д. Принципы цифровой связи и кодирования / А. Д. Витерби, Дж. К. Омура : пер. с англ. под ред. К. Ш. Зигангирова. – М. : Радио и связь, 1982. – 536 с.
5. Блейхут, Р. Теория и практика кодов, контролирующих ошибки / Р. Блейхут. – М. : Мир, 1986. – 576 с.
6. Морелос-Сарагоса, Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение / Р. Морелос-Сарагоса ; пер. с англ. В. Б. Афанасьева. – М. : Техносфера, 2006. – 320 с.

Л. В. Архипов

Россия, Пенза, Пензенский государственный университет

СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Проводится обзор истории развития систем контроля версий программного обеспечения и делается вывод о целесообразности внедрения данных систем.

На сегодняшний день можно утверждать, что информационная эпоха вступила в свои права. Глобализация и массовая культура, информационная война и социальные сети, электронные правительство, документооборот и даже электронное государство – термины, которыми современного человека уже давно нельзя удивить. Как это обычно и бывает за всеми новыми понятиями и стоящими за ними переменами, преобразившими человечество к третьему тысячелетию, стоит идея о новом подходе к ведению наиболее эффективной экономики, суть которого состоит в управлении информацией. Причиной появления этого нового подхода стал значительный прогресс в организации связи и масштабной компьютеризации, которые и позволяют сегодня обеспечивать уровень управления информацией недостижимый в прошлом веке. Например, активно создаются различные информационные и автоматизированные системы, без которых сейчас уже нельзя обойтись крупным фирмам, банкам, производственным предприятиям и государственным структурам. Они в свою очередь представляют собой специальное программное обеспечение, разработка которого может представлять сложную и трудоемкую задачу, решаемую командой специалистов.

Во второй половине XX в. были созданы международная спецификация требований программного обеспечения и отечественный ГОСТ, определен стандарт качества для разработки программного обеспечения. Тем не менее время обеспечение качества остается важной проблемой, вызывающей ряд вопросов. Например, тестирование, являющееся немаловажной частью разработки программного обеспечения (ПО), на которую может быть затрачено половина времени и ресурсов, позволяет определить не столько корректность работы, сколько уровень качества ПО, его соответствие поставленным задачам. И в тоже время тестирование может оставаться слабо регламентированным процессом, которому уделяется недостаточное внимание при разработке. Кроме того нельзя отрицать, что ошибки, которые выявляются при тестировании могли быть заложены еще на этапе проектирования, что показывает необходимость тщательного контроля процесса разработки с момента ее начала. Одним из механизмов упрощения задачи управления разработкой являются системы контроля версий (СКВ).

Данный механизм при обеспечении широкого набора возможностей является достаточно гибким в использовании. Основной его задачей является организация максимально эффективной коллективной работы над проектом и контроль изменений, которые происходят в ходе этой работы.

История развития систем контроля версиями восходит ко времени, когда еще только создавались первые версии операционной системы Unix, а до появления массовых серийных ПЭВМ было еще десять лет. В 1972 г. Марк Рочкинд из Bell Labs заканчивает работу над Source Code Control System (SCCS, «Система Контроля Исходного Кода»), разработанную для IBM System/370, под управлением OS/MVT. В дальнейшем была создана версия и для PDP-11 под управлением операционной системы Unix, а со временем саму систему и вовсе включили в спецификацию для Unix-подобных ОС. SCCS являлась самой распространенной системой на протяжении десятилетия пока инициативу не перехватила RCS. Revision Control System

(«Система Контроля Ревизий») была разработана в середине 80-х гг. Вальтером Ти-чи из Университета Пердью и используется до сих пор при локальной работе с файлами. Даже в современной ОС Mac OS X данная утилита устанавливается вместе с Developer Tools. Будучи свободным ПО, как и многие современные СКВ, RCS оказалась более развитой вариацией своей предшественницы. Храня полную историю о файлах, основанную на применении механизма дельта-компрессии, RCS унаследовала лишь один существенный на то время недостаток – организацию коллективной работы посредством блокировок.

Чтобы лишить RCS присущего недостатка, в тех же 80-х гг. на основе данной системы Дик Грюн разработал набор shell-скриптов, названный CVS (Concurrent Versions System, «Система Одновременных Версий»). Позднее система была переписана на языке Си Брайаном Берлинером, приняв вид в котором используется до сих пор. Благодаря своему основному нововведению, а именно организации коллективной работы команды разработчиков, CVS затмила по популярности всех своих предшественниц. И в наше время эта система является самой широко используемой СКВ. С созданием CVS так же было определено одно из направлений развития СКВ, подвид известный как ЦСКВ («Централизованная Система Контроля Версий»). Используя механизмы хранения файлов согласно формату RCS, CVS позволяет управлять не только локальными файлами, но и группами файлов расположенных в общей директории. Реализация подобного управления осуществляется за счет клиент-серверной архитектуры, в которой вся информация о версиях хранится на сервере, предоставляющем пользователям рабочие копии файлов.

Конечно, с появлением нового, появлялись и неизвестные трудности. Несмотря на свою популярность и новаторство, CVS со временем раскрыла и несколько недостатков. Начали появляться новые системы, призванные разрешить вскрывшиеся проблемы, и со временем их число растет. Это такие ЦСКВ как Subversion (SVN), Perforce, Clearcase и другие. Впрочем, попытки исправить уже сделанное не оказались достаточно эффективным способом решения проблем. Со временем у разработчиков сформировался другой взгляд на дальнейшее направление развития СКВ.

Спустя несколько лет, после успешного дебюта CVS корпорация Sun Microsystems разрабатывает свою СКВ, реализовавшую в себе иное видение организации коллективной работы, отличное от архитектуры центрального сервера. И вот в начале 1990-х годов появляется первая из так называемых распределенных систем контроля версий (РСКВ), получившая название TeamWare. У TeamWare не было центрального сервера, где хранилась информация всех изменениях файлов, а пользователи получали бы данные по запросу. Напротив, каждая локальная копия TeamWare была полной копией проекта, содержавшей в себе всю историю изменений. За основу хранения истории, TeamWare в отличие от CVS, использовавшей RCS, использовала механизмы предложенные уже в позабытой на тот момент SCCS. Основным пользователем TeamWare была сама корпорация Sun, в которой это была (за небольшим исключением) единственная используемая система контроля версий. Несмотря на то, что TeamWare по большей части разрабатывалась для внутренних нужд корпорации, система так же использовалась для контроля ряда крупных проектов, таких как Solaris и Java. Однако с открытием исходных кодов этих продуктов, в них стали использоваться новые системы контроля, а со временем сама TeamWare перестала поддерживаться разработчиком. Впрочем, на этом история развития РСКВ не закончилась. Напротив, по мере роста сообщества разработчиков, руководствующегося принципами свободного и открытого ПО, росло и число новых систем [6].

В 1983 г. Ричард Столлман предпринял попытки создания свободной UNIX-подобной операционной системы, одновременно с этим закладывая основу для

движения свободного ПО. Затянувшаяся разработка проекта GNU получает новый толчок для развития в начале 90-х гг., с публикацией Линусом Торвальдсом ядра Linux. В это же время сеть интернет становится все более доступной. Работа над GNU/Linux и свободными Unix-подобными ОС начинает вестись целым сообществом разработчиков, разделенных расстоянием, но поддерживающих общие идеи и работающих над одним делом. В это же время начинают вскрываться проблемы CVS, и возрастает потребность в создании более совершенной СКВ. Многие современные системы, разработаны под эгидой движения свободного ПО, например, достаточно популярная сейчас Git, которая была создана Линусом Торвальдсом при работе над ядром Linux и выпущена в 2005 г. Все современные системы ориентированы на кроссплатформенность, а многие даже поддерживают работу друг с другом, хотя бы в плане переноса репозитория. Так например, РСКВ Mercurial поддерживает преобразование репозитория из систем CVS, Git, DARCS и имеет инструментарий для ведения параллельной работы с Subversion.

Таким образом, можно сделать вывод, что на сегодняшний день современные СКВ можно разделить в основном на системы, использующие централизованные и распределенные репозитории. Есть и некие гибридные варианты, ярким примером которых является Arch – система предоставляющая несколько моделей для построения РСКВ, и в тоже время способная создать и централизованную систему. Еще одним немаловажным критерием для деления СКВ является механизм хранения изменений файлов. На данный момент он так же насчитывает две модели: «моментальных снимков» и «набора изменений». Выбор в данном случае зависит от того, что для разработчика важнее – компактность репозитория, или же быстрый доступ к версиям файлов.

В итоге можно отметить, что СКВ прошли достаточно долгий путь развития, а причина сегодняшнего разнообразия систем заключается не столько в конкуренции, сколько в специфике задач, которые эти системы должны решать. Разработчик, будь то простой пользователь или крупная компания должен совершать выбор системы, исходя из понимания особенностей задач, которые перед ним стоят. Иными словами СКВ сейчас это полноценное техническое средство, обязательное к наличию в инструментарии разработчика ПО, и от его выбора тоже многое зависит. И в тоже время нельзя назвать СКВ лишь инструментом, так как СКВ предлагает полноценную идеологию разработки ПО и имеет свою терминологию. Однако предлагается разработка регламента по использованию СКВ разработчиком при реализациях крупных проектов с участием больших коллективов.

Список литературы

1. Майерс, Г. Искусство тестирования программ / Г. Майерс, Т. Баджетт, К. Сандлер. – 3-е изд. – М. : Диалектика, 2012. – 272 с.
2. Chacon, S. Pro Git / S. Chacon, B. Straub. – 2 nd ed. – 2014. – 570 p.
3. Система управления версиями. – URL: <http://dic.academic.ru/dic.nsf/ruwiki/44659>
4. Контроль версий. – URL: <http://hlab.org/development/versions/index.html>
5. Системы управления версиями для Linux. – URL: <http://www.ibm.com/developer-works/ru/library/l-vercon/index.html>
6. Швецов, В. В. Алгебраические модели процесса мониторинга распределенного программного проекта / В. В. Швецов, П. П. Макарычев // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2009. – № 3. – С. 62–70.

В. В. Савченко, Р. А. Васильев

Россия, Нижний Новгород, Нижегородский государственный лингвистический университет им. Н. А. Добролюбова

АВТОМАТИЧЕСКАЯ ОЦЕНКА КАЧЕСТВА РЕЧИ ПО КРИТЕРИЮ МИНИМУМА ТРЕБУЕМОЙ ИЗБЫТОЧНОСТИ РЕЧЕВОГО СИГНАЛА

Представлены программная реализация и результаты экспериментального исследования автоматической оценки качества устной речи диктора, основанной на критерии минимума требуемой избыточности речевого сигнала на выходе речевого тракта диктора.

В связи с непрерывным развитием и распространением в технике связи цифровых систем передачи и нелинейной обработки речевого сигнала автоматическая оценка качества речи становится в настоящее время одной из наиболее актуальных задач. Между тем, на данный момент не существует для нее строгого критерия ввиду фундаментальной проблемы многокритериальности качества речи. Оценка минимума требуемой избыточности (ОМТИ) сигнала на выходе речевого тракта диктора [1] – это попытка преодолеть указанную проблему с позиций строгого, теоретико-информационного подхода [2]. Ее эффективность в определенном смысле подтверждают следующие результаты проведенного авторами экспериментального исследования.

В основе ОМТИ используется выражение для относительной величины требуемой избыточности речевого сигнала по формуле относительной частоты

$$\hat{\alpha} = m_0 / N \times 100\% \quad (1)$$

появления нечетких (сомнительных) фонем в серии из N последовательных наблюдений $\mathbf{x}$ над речевым сигналом на интервале длиной в одну минимальную речевую единицу каждый. При этом признаком указанной нечеткости фонем в процессе их восприятия условным слушателем служило ограничение сверху

$$J^{-1} \sum_{j=1}^J \rho(\mathbf{x} / \mathbf{x}_j) \leq \rho_0 \quad (2)$$

на величину их информационного рассогласования по Кульбаку-Лейблеру в среднем по кластеру $\{\mathbf{x}_j\}$ одноименных (однофонемных) речевых единиц из фонетической базы данных диктора. Здесь J – объем кластера. Это известный [2] критерий минимума информационного рассогласования (МИР).

Точность и надежность ОМТИ нетрудно охарактеризовать с помощью классического неравенства Бернулли:

$$D\{|m_0 / N - \alpha| \leq \varepsilon\} > 1 - \alpha(1 - \alpha) / (\varepsilon^2 N) = \beta.$$

Задаваясь допустимой шириной доверительного интервала $\varepsilon \leq 0,05$, для случая $\alpha \leq 0,1$ будем иметь доверительную вероятность β порядка 0,99 и выше при объеме выборки N , равном 10 тысячам единиц и более. Если учесть, что размерность вектора $\mathbf{x}$ составляет [2] минимум 80 отсчетов речевого сигнала при частоте дискретизации 8 кГц, объем выборки N достигает требуемого значения на интервале наблюдения длиной порядка полутора-двух минут. Это говорит о беспрецедент-

но высоких динамических свойствах предложенной оценки. Причем, на практике именно относительная величина (1) часто представляется предпочтительной по сравнению с абсолютной величиной теоретико-информационного показателя качества речи. Например, это справедливо в задачах психологического тестирования личности по принципу сопоставления двух ОМТИ из выражения (1): полученных до и после нагрузки на диктора. Задача такого рода была подробно рассмотрена далее – в качестве предмета экспериментальных исследований.

Для экспериментальных исследований ОМТИ был разработан экспериментальный образец информационной системы [3], реализованный в виде компьютерной программы PLD (Phonetic Lie Detector) на платформе Java SE 1.6. Программа позволяет выполнять все операции над речевым сигналом x согласно алгоритму обработки (2). Ее главное окно показано на рис. 1.

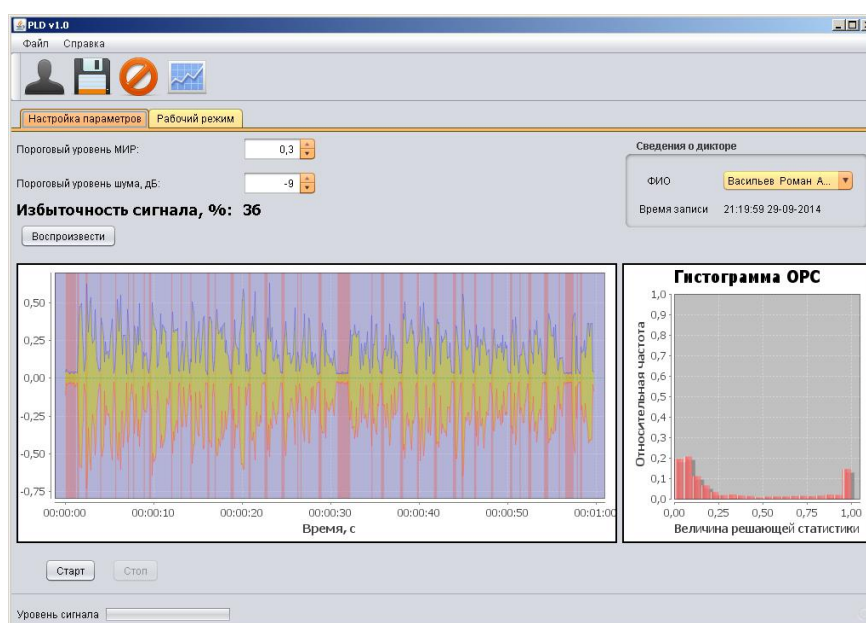


Рис. 1. Главное окно программы PDL

Здесь под «Избыточностью сигнала» понимается выборочная оценка ОМТИ согласно выражению (1). А «Пороговый уровень МИР» $\mu_0 \leq 1$ определяет правую часть неравенства (2) согласно пропорциональной зависимости $\rho_0 = \mu_0 K - 1$, где K – коэффициент пропорциональности. И, наконец, «Пороговый уровень шума, дБ» – это стандартный параметр амплитудной селекции речевого сигнала из его аддитивной смеси с фоновым шумом микрофона. В процессе экспериментальных исследований он устанавливался на уровне 2–3 СКО шума в отсутствие полезного сигнала.

Для экспериментальных исследований была составлена группа из четырех дикторов: три мужчины разного возраста и примерно одного уровня образования и одна женщина, все без явно выраженных дефектов речи. Цель проведенных далее исследований – изучение влияния физической или эмоциональной нагрузки на диктора на качество его речи на подсознательном, фонетическом уровне. Для этого был применен комплекс специальных аппаратных и программных средств, в том числе микрофон Genius и программа Audacity 1.2. Частота дискретизации встроенного АЦП была установлена равной 8 кГц – это стандартное ее значение при обработке разговорной речи.

В процессе проведенных исследований для каждого диктора в режиме настройки программы, сначала вычислялась предварительная оценка ОМТИ. При этом каждым из дикторов в комфортном режиме был прочитан один и тот же художественный текст – из первой главы романа А.С. Пушкина "Капитанская дочка" – объемом в одну стандартную машинописную страницу. Продолжительность речевого сигнала здесь составляла не менее одной минуты. Формируемые речевые сигналы сначала были записаны в память ПК в формате wav. Его роль выполнял современный ноутбук Acer Extensa 5620, 2 Гбайт ОЗУ, Windows 7. И далее, уже в рабочем режиме, с использованием предварительной оценки ОМТИ исследовалась динамика качества речи диктора в зависимости от условий его монолога. Соответствующее окно программы показано на рис. 2.



Рис. 2. Рабочий режим программы PDL

При этом физические нагрузки на каждого диктора имитировались путем выполнения предварительных (перед чтением текста) физических упражнений (до 10 подряд приседаний, отжиманий и т.п.), а эмоциональная нагрузка – путем увеличения времени чтения текста в интервале от 5 до 45 мин. Полученные результаты сводятся к следующим выводам.

1. Исследуемый показатель качества речи диктора практически инвариантен к выбору текста для чтения, времени и длительности его записи и, вместе с тем, сильно критичен по отношению пороговому уровню МИР, а также к физической и эмоциональной нагрузке на диктора в процессе чтения им текста.

2. При увеличении физической нагрузки избыточность речи в общем случае пропорционально возрастает, причем, сначала резко, а потом, по мере вовлечения организма диктора в режим интенсивной физической нагрузки, медленнее. Причем у спортсменов процесс адаптации протекает быстрее.

3. Установлена также явная тенденция увеличения избыточности речи на фоне общей нестабильности ее качества при длительном эмоциональном напряжении диктора. При этом динамика избыточности имела характер колебаний – точно в соответствии с естественными колебаниями степени сосредоточенности диктора на

конкретном тексте. Отметим при этом, что в общем случае указанные колебания постепенно затухали во времени, причем, на определенном (повышенном) уровне избыточности речи диктора.

Таким образом, можно утверждать, что ОМТИ – это относительная характеристика, которая нацелена не на сравнение речи разных дикторов, а на исследование влияния разного рода факторов на качество речи конкретного диктора. Анализируя по ОМТИ в процессе речеобразования текущие отклонения речевого сигнала диктора от его же нормы в заведомо комфортных условиях, мы можем установить как факт отклонения качества речи диктора от нормы, так и факт его ненормального физического или эмоционального состояния, причем, с оценкой допустимых пиковых значений нагрузок на данного диктора. Поэтому свое наиболее широкое практическое применение оценка ОМТИ должна найти по направлению психологических исследований особенностей личности по ее речевому сигналу.

Список литературы

1. Пат. на полезную модель № 102138. Устройство для фонетического анализа и тестирования речи / В. В. Савченко, Д. Ю. Акатьев / Роспатент. – № 2010135569 ; заявл. 25.08.2010. – Оп. в БИ, 2011, № 3.
2. Савченко, В. В. Информационная теория качества речи / В. В. Савченко // Известия вузов России. Радиоэлектроника. – 2011. – Вып. 1. – С. 17–27.
3. Информационная система тестирования эмоционального состояния личности по голосу : программа для ЭВМ / А. В. Савченко, В. В. Савченко, Д. Ю. Акатьев, И. В. Губочкин / Роспатент. – № 2013611003 ; зарег. 09.01.2013.

Е. А. Войнова, А. Н. Токарев

Россия, Пенза, Пензенский государственный университет

СРАВНИТЕЛЬНЫЙ АНАЛИЗ И ВЫБОР СУБД ДЛЯ СИСТЕМЫ УПРАВЛЕНИЯ НАЗЕМНОЙ ИНФРАСТРУКТУРОЙ РАКЕТНО-КОСМИЧЕСКОЙ ТЕХНИКИ

Рассматривается вопрос выбора системы управления базами данных для системы управления наземной инфраструктурой ракетно-космической техники. Проведен сравнительный анализ систем управления базами данных, как PostgreSQL, Oracle, MS SQL и MySQL.

При проектировании системы управления наземной инфраструктурой ракетно-космической техники одним из важных вопросов, является выбор системы управления базами данных (СУБД) для обеспечения хранения данных, обрабатываемых в проектируемой системе. К СУБД системы управления наземной инфраструктурой ракетно-космической техники предъявляются такие требования, как: соответствие стандарту SQL, полная поддержка надежных транзакций. В настоящее время современные информационные системы чаще всего работают под управлением таких СУБД как PostgreSQL, Oracle, MS SQL или MySQL. Поэтому для корректной работы системы управления наземной инфраструктурой ракетно-космической техники необходимо правильно оценить возможности различных СУБД. В данной статье проводится сравнительный анализ функций и возможностей СУБД PostgreSQL, Oracle и MS SQL, MySQL, с целью наиболее рационального использования для проектируемой системы.

Oracle предлагает наиболее эффективные методы для управления материализованными представлениями. Материализованные представления можно создавать как результат любого SQL выражения. Поддерживается автоматическое и мануальное обновление представлений. Дополнительно поддерживаются: неблокирующие обновления, сжатие данных, построение обновлений на предварительно вычисленных данных. Основные преимущества Oracle Database:

Real Application Cluster обеспечивает работу одного экземпляра базы данных на нескольких узлах grid, позволяя управлять нагрузкой и гибко масштабировать систему в случае необходимости;

Automatic Storage Management позволяет автоматически распределять данные между имеющимися ресурсами систем хранения данных;

Oracle Database позволяет автоматически управлять уровнями сервиса и тиражировать эталонные конфигурации в рамках всей сети.

простые средства разработки;

специальные механизмы Oracle Database позволяют оптимизировать и корректировать SQL-запросы, выявлять и прогнозировать ошибки;

максимальный размер базы данных Oracle может достигать 8 экзабайт;

переносимые табличные пространства, система управления потоками данных Oracle Streams и модель распределенных SQL-запросов.

Microsoft SQL Server, в отличие от Oracle Database, применяет вместо материализованных представлений так называемые “индексированные представления”. Индексированные представления сохраняются в базе данных прозрачно для пользователя, каждое изменение данных в базовой таблице приводит к пересчету данных в индексированном представлении. Преимущество решения Microsoft в том, что оптимизатор запросов с большей вероятностью обратится к индексированному представлению, в случае если пользователь в SQL выражении вычисляет соответствующий уровень агрегации. Кроме того, данные в индексированных представлениях всегда соответствуют данным в таблицах. К ключевым преимуществам Microsoft SQL Server можно отнести:

обеспечение возможности динамического шифрования;

поддержка автоматического восстановления любых страниц данных;

обеспечение сжатия исходящего потока журнала и сведения к минимуму нагрузки на сеть;

регулятор ресурсов позволяющий добиться стабильного и предсказуемого времени отклика на запросы со стороны конечных пользователей;

управление на основе политик, позволяющее управлять несколькими имеющимися экземплярами SQL Server;

Также следует обратить внимание и на СУБД MySQL. Это самая распространенная полноценная серверная СУБД. MySQL – функциональная, свободно распространяемая СУБД, которая успешно работает с различными сайтами и веб приложениями. К преимуществам MySQL можно отнести: простоту в работе, поддержку большинства функционала SQL; большое количество функций обеспечивающих безопасность; масштабируемость – MySQL легко работает с большими объемами данных и легко масштабируется; скорость – упрощение некоторых стандартов позволяет MySQL значительно увеличить производительность.

При этом MySQL имеет ряд недостатков, среди которых выделяют: ограничения функционала MySQL, заложенные для обеспечения быстродействия некоторых особо требовательных приложений, а также проблемы с надежностью - из-за некоторых способов обработки данных MySQL иногда уступает другим СУБД по надежности.

Учитывая вышесказанное, можно сделать вывод, что в проектах, в которых требуется строгое соответствие стандартам MySQL – не лучший выбор. Еще одним важным аспектом, при выборе БД для проектируемой платформы является многопоточность. В MySQL довольно легко выполняется параллельное чтение, но параллельные операции чтения-записи могут создать проблемы. В связи с этим становится ясно, что MySQL не подходит для использования в системе управления наземной инфраструктурой ракетно-космической техники.

Самым профессиональным из всех свободно-распространяемых СУБД является PostgreSQL. Она максимально соответствует стандартам SQL. PostgreSQL, или Postgres, стараются полностью применять ANSI/ISO SQL стандарты своевременно с выходом новых версий. От других СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного и/или реляционного подхода к базам данных. Например, полная поддержка надежных транзакций, т.е. атомарность, последовательность, изоляционность, прочность (Atomicity, Consistency, Isolation, Durability (ACID)). Благодаря мощным технологиям PostgreSQL очень производительна. Параллельность в PostgreSQL достигнута благодаря реализации управления многовариантным параллелизмом (MVCC), что также обеспечивает соответствие ACID. В PostgreSQL очень легко управлять своими процедурами, называемые хранимые процедуры. Эти функции упрощают использование постоянно повторяемых операций. Среди достоинств PostgreSQL можно выделить:

PostgreSQL – бесплатное программное обеспечение с открытым исходным кодом;

большое количество дополнений – несмотря на огромное количество встроенных функций, существует очень много дополнений, позволяющих разрабатывать данные для этой СУБД и управлять ими;

расширения – существует возможность расширения функционала за счет сохранения своих процедур;

объектность – PostgreSQL это не только реляционная СУБД, но также и объектно-ориентированная с поддержкой наследования и много другого

PostgreSQL также имеет некоторые недостатки:

при простых операциях чтения PostgreSQL может значительно замедлить сервер и быть медленнее своих конкурентов;

популярность – по своей природе, популярностью эта СУБД похвастаться не может, хотя и присутствует довольно большое сообщество;

хостинг – в силу выше перечисленных факторов иногда довольно сложно найти хостинг с поддержкой этой СУБД.

В табл. 1 рассмотрены основные функции СУБД PostgreSQL, Oracle и MS SQL.

Нельзя не отметить, что Oracle 11G и Microsoft SQL Server представляют собой платные СУБД, поэтому использование в качестве СУБД для единой базовой платформы управления наземной инфраструктурой ракетно-космической техники PostgreSQL более предпочтительно.

Таким образом, применение PostgreSQL является идеальным выбором, когда необходимо максимальное соответствие стандартам SQL, когда надежность и целостность данных в приоритете над быстротой выполнения простых операций чтения.

Также, в случае, если необходимо использовать сложные пользовательские процедуры, то PostgreSQL имеет встроенную поддержку для них. Сложная струк-

тура данных – по сравнению с другими открытыми СУБД PostgreSQL предоставляет больше возможностей для создания сложных структур данных без необходимости жертвовать какими-либо аспектами. Более того, переход на платные СУБД, например Oracle, с PostgreSQL будет довольно просто осуществить по сравнению с другими бесплатными СУБД. И хотя PostgreSQL имеет ряд недостатков, они являются незначительными при проектировании системы управления наземной инфраструктурой ракетно-космической техники.

Таблица 1

Основные функции СУБД PostgreSQL, Oracle и MS SQL Oracle 11

	Oracle 11G	Microsoft SQL Server	PostgreSQL
Rebuild	Да	Нет	Да
Refresh	Да	Да	Да
Оптимизация запросов	Да	Да	Нет
Аналитические функции	Да	Да	Нет
MERGE оператор	Да	Да	Нет
Виртуальные колонки	Да	Да	Нет
Сжатие данных	Да	Да	Да
Анализ запросов	Да	Да	Да
Планировщик задач	Да	Да	Да
Моделирование схемы данных	Да	Да	Да
Эффективный импорт данных	Да	Нет	Нет

И в заключение, стоит обратить внимание, PostgreSQL поддерживает кроссплатформенность, т.е. возможность использовать как в системах под управлением ОС Windows, так и под управлением ОС Linux, более того, в PostgreSQL отсутствует ограничение на использование RAM, не ограничено использование CPU, количества ядер (процессоров), возможно создание БД неограниченного размера, ну и конечно, несомненным плюсом является поддержка отказоустойчивых кластеров.

Список литературы

1. PostgreSQL 9.3.5 Documentation. 1996–2014 / The PostgreSQL Global Development Group. – URL: <http://www.postgresql.org/docs/current/static/>
2. Библиотека Microsoft SQL Server. – URL: <http://msdn.microsoft.com/ru-ru/library/bb545450.aspx>
3. Oracle Database Online Documentation 11g Release 1 (11.1). – URL: http://docs.oracle.com/cd/B28359_01/index.htm

А. Н. Токарев, Е. А. Войнова

Россия, Пенза, Пензенский государственный университет

ХРАНЕНИЕ ДРАЙВЕРОВ ДЛЯ ПОДКЛЮЧАЕМЫХ МОДУЛЕЙ В СИСТЕМЕ УПРАВЛЕНИЯ НАЗЕМНОЙ ИНФРАСТРУКТУРОЙ РАКЕТНО-КОСМИЧЕСКОЙ ТЕХНИКИ

Рассматривается вопрос выбора способа хранения драйверов для подключаемых модулей в систему управления наземной инфраструктурой РКТ. Анализируется возможность использования для этой цели специализированной СУБД, либо хранения данных в классической базе SQL-типа в двух вариантах – хранение драйверов непосредственно в таблицах БД, либо хранение ссылок на файлы в файловой системе.

Применительно к рассматриваемой системе управления, драйвера для подключенных модулей представляют собой некоторые бинарные данные (исполнимые файлы и библиотеки). Вопрос хранения и доступа к таким данным в используемой клиент-серверной системе является частным случаем проблемы систематизации, хранения и доступа к бинарным данным.

Одним из решений для хранения бинарных данных достаточно большого объема является использование специализированных баз данных или надстроек над ними. В качестве примера можно привести систему Riak – распределенную базу данных NoSQL с открытым исходным кодом, и Riak CS – программное обеспечение с открытым исходным кодом для облачных систем хранения. Оба решения предназначены для хранения больших объемов критически важных неструктурированных данных, к которым пользователям и приложениям предоставляется моментальный доступ. Это могут быть, например, медицинские карты пациентов больницы, архивированные метеорологические сводки, коллекции видеозаписей в крупных медиасетях, пользовательские данные и настройки приложений для мобильных устройств, а также многое другое.

Основным отличием Riak от других баз данных NoSQL является ее исключительная отказоустойчивость – это решение отлично подойдет для сред, в которых недопустимы простои. Благодаря встроенной функции репликации данных и децентрализованной конфигурации, Riak функционирует даже при нарушении связности сети и отказе отдельных устройств – в ситуациях, которые привели бы к потере данных в обычных базах. Еще одним преимуществом решения Riak является достаточная простота эксплуатации, обслуживания и расширения системы. Пользователи могут добавлять узлы и серверы в кластер или удалять их из него. При этом перебалансировка данных в кластере выполняется автоматически. При добавлении новых устройств производительность системы только увеличивается, почти пропорционально увеличению емкости. При отключении узлов система продолжает обеспечивать чтение и запись данных.

Программное обеспечение Riak CS разработано на базе Riak и предоставляет достаточно большие функциональные возможности, в частности поддержку больших объектов и интерфейсов прикладных программ и получать разнообразную статистику хранения и доступа. Riak CS обеспечивает поддержку больших объектов и многокомпонентную передачу, позволяя работать даже с объектами терабайтных размеров. При загрузке в систему Riak CS объект делится на части, которые одновременно передаются потоком, сохраняются и реплицируются в базовый кластер. При извлечении каждый блок связывается с метаданными. Все данные в системе

реплицируются, а при отказе одного из узлов его функции автоматически начинают выполнять другие узлы, поэтому доступ к данным гарантирован даже при отказе части оборудования.

Распределенная база данных состоит из нескольких систем, которые подключаются непосредственно к серверу и хранят данные в виде пар «ключ-значение». Использование накопителя Seagate Kinetic упрощает управление системами хранения данных в виде пар «ключ-значение», файловой системой, диспетчером логических томов, RAID-контроллером и подключенными устройствами, заменяя их простым сетевым сокет-интерфейсом. Поскольку количество накопителей в системе не связано жестко с количеством шасси в серверной стойке, можно увеличивать емкость и производительность облака в соответствии с потребностями.

Однако, поскольку рассматриваемая система хранения драйверов является подсистемой общего хранилища, а системы типа Riak для хранения неструктурированных данных – специализированной БД, при ее использовании в качестве основной базы придется пойти на некоторые жертвы. Либо использовать ее совместно с одной из классических БД SQL-типа для хранения основного набора данных. Естественно, в данном случае повышается сложность написания и поддержки программного обеспечения системы. Кроме того, количество файлов и объем данных при хранении драйверов не является критическим.

По этой и ряду других причин для хранения файлов драйверов имеет смысл использовать либо собственный специализированный API для хранения драйверов в виде файлов, либо использовать возможности хранения бинарных данных в самой базе SQL.

У каждого из этих подходов есть свои плюсы и минусы.

Преимущества хранения файлов в БД:

Конкурентный доступ. СУБД самостоятельно поддерживает корректность одновременного доступа на изменение одной и той же записи;

Файлы, хранящиеся в базе, не подвержены заражению вирусами;

Имеется полный программный контроль над файлами (например, отдача файлов по привилегиям);

Возможность полнотекстового поиска (например, в xml).

Для хранения файлов в файловой системе необходимо разработать алгоритм их идентификации. Самый простой вариант – файлы хранить в определенном каталоге на диске, а в БД хранить в базе путь к каждому файлу. Однако, в ряде случаев оптимальнее в конфигурации приложения указать корневой каталог для файлов, а относительный путь уже строить в коде, основываясь на идентификаторе. В итоге путь хранить в БД необязательно, поэтому пропадает один столбец данных, плюс появляется возможность гибкой настройки, и появляется единообразие.

Преимущество хранения в виде файлов:

В общем случае доступ к файлам осуществляется быстрее, чем получение записи из БД (даже если используются сокеты), поскольку отсутствует прослойка в виде СУБД, и используется только драйвер файловой системы;

Удобство бэкапа – восстанавливать можно по отдельным файлам, без необходимости раскрывать весь дамп;

Количество файлов может быть очень большим без потери производительности;

Кэширование файлов осуществляет автоматически ОС и сервер. В случае использования СУБД кэширование больших файлов вызывает потерю производительности из-за вытеснения простых запросов из кэша.

Следует отметить также некоторые особенности.

Если файлы хранятся в БД, то кэш БД будет занят файлами, а не записями из других таблиц, т.е. в общем случае нет способа приоритезировать запросы и поделить место в памяти между файлами и данными, что в свою очередь может ухудшить производительность для простых запросов с данными.

И, наконец, главная особенность хранения отдельных файлов: если файлы хранятся на диске, а метаданные о них в БД, тут в такой системе хранения отсутствует транзакционная целостность. Это значит, что появляются такие проблемы:

- файлы не удаляются при удалении соответствующей записи БД;

- проблемы при одновременной попытке обновления файла

- нарушение синхронизации между БД и файловой системой при откате транзакции;

- при резервном копировании и восстановлении информации в БД может возникнуть рассинхронизация с файловой системой;

- файлы не подчиняются ограничениям доступа, наложенным с помощью БД.

В СУБД PostgreSQL для хранения файлов с произвольным содержимым непосредственно в БД широко используются два типа данных: BYTEA и LargeObjects.

BYTEA – это двоичная строка переменной длины, которая в отличие от обычной строки специально позволяет хранить байтовое представление таких символов как ноль и других непечатаемых символов.

Тип BYTEA поддерживает два внешних формата для ввода и вывода: исторический «escape» формат PostgreSQL и «шестнадцатеричный» формат.

Шестнадцатеричный («hex») формат кодирует двоичные данные как два шестнадцатеричных разряда на байт, старший разряд идет первым. Строка целиком состоит из последовательности «\x». Шестнадцатеричный формат совместим с широким диапазоном внешних приложений и протоколов, а также он имеет тенденцию к более быстрому преобразованию данных, чем в случае escape формата, так что его использование является предпочтительным.

«Escape» формат является традиционным форматом для типа BYTEA в PostgreSQL. Он использует подход представления двоичной строки как последовательности ASCII символов, во время конвертации байты, которые не могут быть представлены как ASCII символы, заменяются специальными escape последовательностями.

Однако, как показывает практика, в некоторых случаях существует проблема с записью в поле типа BYTEA данных большого объема. При этом на сервере возникает ошибка переполнения памяти, так как сервер пытается выделить памяти значительно больше, чем объем сохраняемых данных. Это связано с перекодировкой из клиентской кодировки в кодировку сервера (т.к. BYTEA передается как строка), и с тем, что сервер пытается нестроковые символы привести к Escape-последовательностям.

PostgreSQL имеет специальный тип для хранения больших двоичных файлов – Large Objects. Работа с ними происходит быстрее, чем с BYTEA, и не возникает проблем выделением памяти на сервере. Единственное неудобство заключается в том, что работать с этим типом можно только внутри транзакций, т.е. для чтения и записи необходимо явно стартовать транзакцию. Тип Large Object хранит бинарные данные в отдельной таблице в специальном формате, и ссылается на эту таблицу из поля обычной таблицы с помощью значения типа OID. Large Object не требует большого количества памяти на сервере для обработки запросов. Однако, у этого ти-

па есть и свои особенности. Удаление строки из таблицы со ссылкой на Large Object не удаляет сам объект физически, для этого нужно произвести отдельную операцию. Также, Large Objects имеют некоторые особенности, связанные с безопасностью – любой пользователь БД имеет доступ к Large Objects безотносительно уровней доступа к обычным таблицам.

Поэтому, прежде чем выбрать тот или иной вариант, следует тщательно проанализировать варианты использования системы и степень важности того или иного фактора при доступе к таким данным.

Список литературы

1. Basho: создание эффективной и недорогой объектной системы хранения с помощью распределенных баз данных с открытым кодом. – URL: <http://rss.seagate.com/ru/ru/case-studies/basho-master-cs/>
2. PostgreSQL documentation. – URL: <http://www.postgresql.org/docs/manuals/>

Д. В. Горбачев

Россия, Оренбург, Оренбургский государственный институт менеджмента

МЕТОДИЧЕСКИЙ ПОДХОД К РАЗРАБОТКЕ ИНФРАСТРУКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ ЛЕЧЕБНО-ДИАГНОСТИЧЕСКОГО ПРОЦЕССА ПРИ ОКАЗАНИИ МЕДИЦИНСКОЙ ПОМОЩИ

Опыт внедрения медицинских информационных систем (МИС) показывает, что специфика медицинских организаций учитывается не всегда и не в полной мере. Это, в свою очередь, приводит к не согласованной работе элементов МИС. В данной работе предлагается подход к построению МИС, основывающийся на существующих потоках данных между участниками лечебно-диагностического процесса.

Основным процессом, реализуемым медицинской организацией является лечебно-диагностический процесс (ЛДП), суть которого заключается в оказании медицинской помощи населению. В современной медицинской организации этот процесс протекает в соответствии со стандартами (их в настоящее время более тысячи) и порядками оказания медицинской помощи по видам нозологий. Применение стандартов и порядков в значительной мере регламентирует ЛДП, что, в свою очередь, позволяет формализовать подпроцессы, операции и работы диагностики, лечения, профилактики и реабилитации.

Кроме того, для любой медицинской организации, как для экономического объекта, характерными являются следующие бизнес-процессы.

В современных медицинских организациях и лечебно-диагностический, и бизнес-процессы активно автоматизируются с помощью информационных систем и программно-аппаратных комплексов. Наиболее известные примеры приложений комплексной автоматизации ЛДП с помощью медицинской информационной системы (МИС) приведены в [1].

Анализ большинства, из приведенных, МИС показывает, что основной функцией, реализованной в них, является информационно-учетно-аналитическая, предполагающая ведение информационных баз и справочников, учет отношений «*медицинская организация* ↔ *пациент*» и формирование отчетов по медицинской статистике. Вместе с тем, большой интерес представляет исследование и решение проблемы организации эффективного информационного взаимодействия между участниками лечебно-диагностического процесса и обеспечение управленческого персонала необходимой для принятия решений информацией.

Таким образом, задача современного развития МИС заключается в разработке такой методической платформы проектирования инфраструктуры МИС, которая бы наилучшим образом обеспечивала решение задач лечебно-диагностического и бизнес-процессов, а также предоставляла необходимые сервисы в соответствии с социально экономической спецификой данного объекта. Поскольку наименее формализованным в настоящее время является ЛДП, то в дальнейшем в исследовании проводится разработка моделирующей платформы именно для него.

Решение этой задачи выполняется в соответствии шести уровневой моделью (рис. 1):

- 1) разработка вербальных моделей лечебно-диагностического процессов – обеспечивается понимание масштаба системы (медицинской организации), характера внутренних и внешних взаимосвязей;
- 2) разработка типовых алгоритмов основных операций ЛДП – позволяет перейти к решению задач локальной формализации процессов с помощью типовых математических схем моделирования;
- 3) разработка типовых формальных моделей, описывающих ЛДП и позволяющих получать оценки качественных характеристик;
- 4) разработка способа интерпретации информационных взаимосвязей модельных объектов и реальных объектов участников информационного обмена в целях получения количественных оценок параметров связей;
- 5) формирование технических требований к элементам инфраструктуры с учетом интенсивности и объема информационного потока и специфики данных в этом потоке;
- 6) решение задачи проектирования, как задачи комплектования инфраструктуры МИС оборудованием с необходимыми функциональными характеристиками и наименьшей стоимостью.

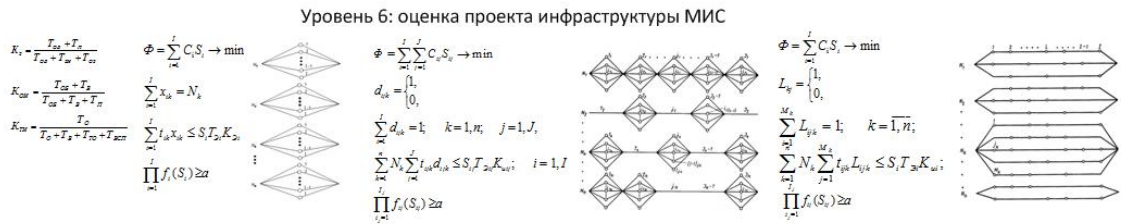
Концептуальная основа построения архитектуры системной аналитико-имитационной модели (АИМ) процессов медицинской организации была заложена при выполнении исследования [2]. В соответствии с введенным математическим формализмом основополагающим принципом при построении АИМ является блочно-модульный принцип.

Отличительной особенностью предлагаемого подхода является то, что каждый модуль представляет собой модель ЛДП, а объединение отдельных модулей в единую систему осуществляется посредством операторов сопряжения, определяемых в терминах алгебраической модели.

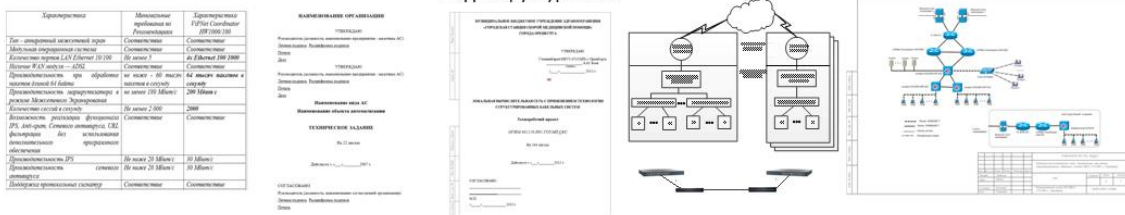
Развитие положений системного подхода к анализу действующих факторов, определяющих ход ЛДП, позволяет сформировать базу показателей и параметров моделирования.

В основу этой базы закладывается классификация множества характеристик системы по ряду признаков, подразделяющихся на две основные группы: определяющие (основные) Δ^* и второстепенные (не основные) $\bar{\Delta}$ показатели, параметры и характеристики. В число и тех и других входят:

- факторы, характеризующие внешнюю среду, F_6 ;
- параметры, определяющие назначение системы, H_6 ;
- оценочные показатели эффективности процессов, P_6 .



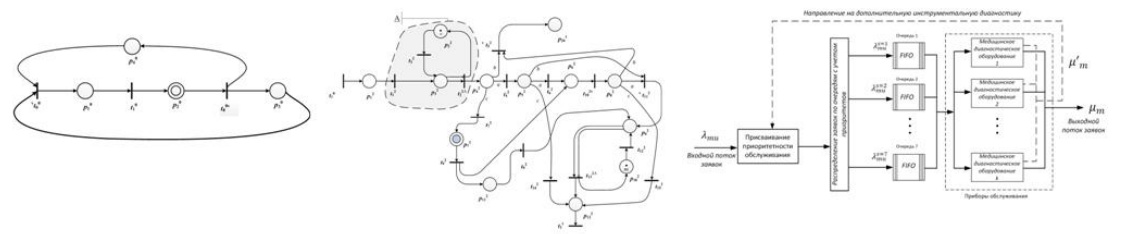
Уровень 5: интерпретация результатов имитационного моделирования. Формирование требований к инфраструктуре МИС



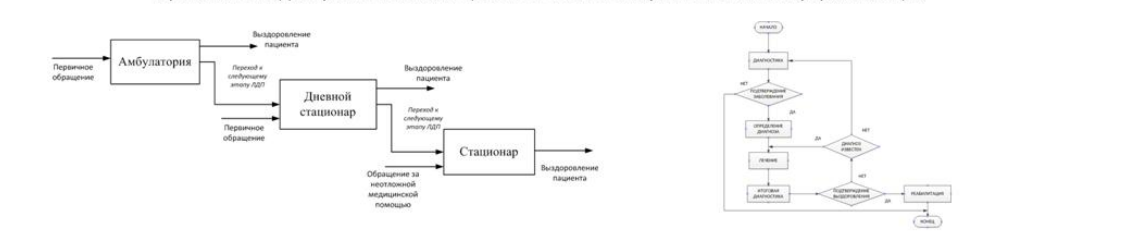
Уровень 4: имитационное моделирование на ЭВМ. 3-й этап формализации



Уровень 3: моделирование с помощью типовых математических схем. 2-й этап формализации



Уровень 2: моделирование с помощью блок-схем и алгоритмов. 1-й этап формализации



Уровень 1: моделирование лечебно-диагностического и бизнес-процессов. Вербальное моделирование

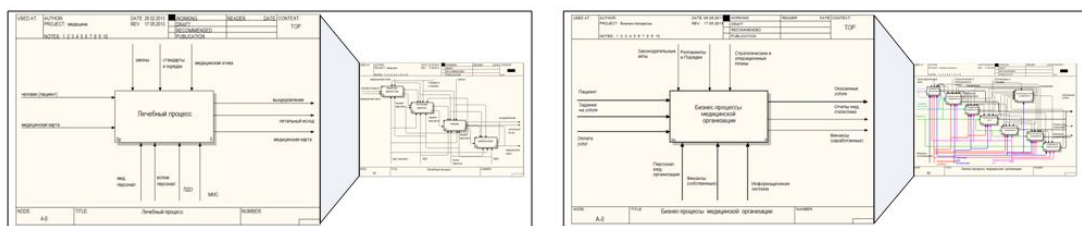


Рис. 1. Уровневый подход к решению проблемы формирования МИС

В свою очередь, факторы F_{ϵ} отражают условия, в которых развивается процесс: система нормативных и регламентирующих документов, ресурсное обеспечение, научные и методические положения и требования.

Параметры H_c включают в себя характеристики процессов по уровням моделирования: экономические показатели, медико-технологические показатели, показатели организационно-управленческого характера.

Показатели P_o , представляют собой оценки эффективности лечебно-диагностического и бизнес-процессов.

Весь комплекс показателей можно записать как

$$\mathfrak{R} = \Delta^* \cup \bar{\Delta} = \langle F_B^* \cup H_C^* \cup P_O^* \rangle \cup \langle \bar{F}_B \cup \bar{H}_C \cup \bar{P}_O \rangle,$$

где множества F_{ϵ} , H_c , P_o имеют следующую структуру:

$$F_{\epsilon} = \{f_{\epsilon}^{(1)}, f_{\epsilon}^{(2)}, \dots, f_{\epsilon}^{(k)}\};$$

$H_c = \langle X_{(i)} \cup M_{(i)} \rangle$, где $X_{(i)} = \{x_{(i)}^1, x_{(i)}^2, \dots, x_{(i)}^l\}$ – характеристики бизнес-процессов,

$M_{(i)} = \{\mu_{(i)}^1, \mu_{(i)}^2, \dots, \mu_{(i)}^l\}$ – медико-технологические показатели ЛДП;

$P_c = \langle \{\kappa\} \cup \hat{M}_{(i)} \cup \hat{Z}_{(i)} \rangle$ – где κ – критерий эффективности функционирования медицинской организации, $\hat{M}_{(i)}$ – оценки медико-технологических показателей, характеризующие качество организации ЛДП; $\hat{Z}_{(i)} = \{\hat{z}_{(i)}^1, \hat{z}_{(i)}^2, \dots, \hat{z}_{(i)}^v\}$ – оценки экономических затрат на реализацию управленческих решений.

Разделение показателей на группы позволяет определить их место в общей иерархии и их сравнительную значимость.

Всю совокупность показателей $\mathfrak{R}$ целесообразно разбить на следующие классы:

- фиксируемые (W) – задаются до начала разработки модели и не изменяются в процессе моделирования;
- управляемые (G) – подлежат выбору при моделировании;
- оценки качества (Q) – характеризуют уровень соответствия системы предъявляемым требованиям.

При этом, множества $W = \{w_1, w_2, \dots, w_m\}$, $G = \{g_1, g_2, \dots, g_n\}$, $Q = \{q_1, q_2, \dots, q_n\}$, являются собственными непересекающимися подмножествами $W \cup G \cup Q = \emptyset$.

Множество $G \cup Q = \mathfrak{M}W$ определяет изменяемую (оптимизируемую) часть моделируемой системы, а множество W – ограничения типа равенств, неравенств, дискретности и другие, в которых происходит поиск решения задачи оптимизации.

В достижении цели проводимого исследования, заключающейся в решении задачи приведения в соответствие инфраструктуры информационной системы и системы ЛДП медицинской организации, данная классификация показателей играет роль аналитической платформы, поскольку является полностью формализованной частью имитационной модели. А также она составляет основу базы данных программной модели (рис. 2).

Каждая база данных информационного массива АИМ представляет собой реляционную систему, хранящую сведения о характеристиках и параметрах ЛДП.

Ядро имитационной системы предоставляет человеко-машинные интерфейсы для взаимодействия оператора с моделью. Интерфейсы имеют три режима организации такого взаимодействия: на основе шаблонов, конструктора, мастера. С помо-

щью интерфейса оператор осуществляет построение схемы моделирования, устанавливает время и число прогонов модели, определяет эндогенные и экзогенные переменные, определяет структуру бизнес-процессов. Кроме того, модуль интерфейсов имеет связь с другими модулями ядра.

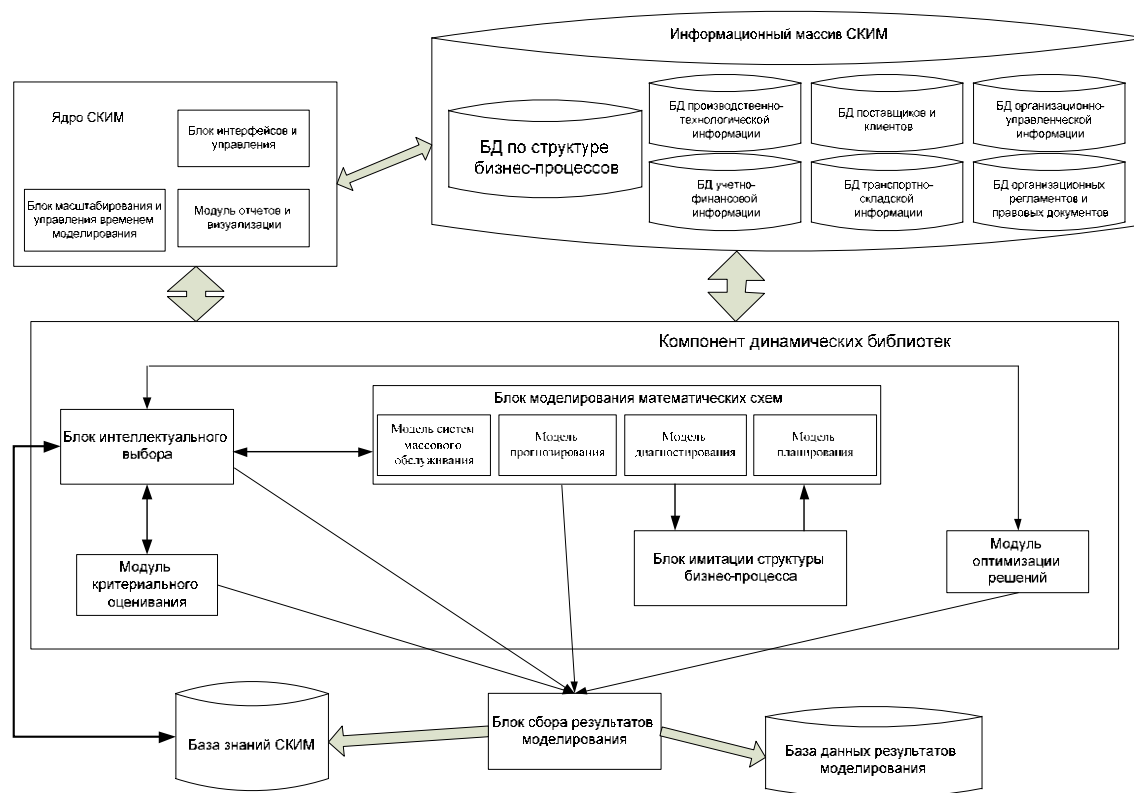


Рис. 2. Архитектура СКИМ моделирования бизнес-процессов

Связь с модулем визуализации позволяет формировать, сохранять и выводить на печать отчеты по работе модели. Взаимодействие с блоком масштабирования и управления временем позволяет установить временной масштаб моделирования и дискреты наблюдения и фиксации результатов моделирования. Таким образом, ядро системы моделирования посредством манипуляций оператора создает модельную ситуацию и регламентирует процесс имитации функционирования предприятия.

В блоке интеллектуального выбора решаются задачи формирования альтернатив и области принятия решений. Действием, предваряющим формирование альтернатив, является анализ и оценка задачи управления, в результате чего делается вывод о том, является ли модельная ситуация задачей управления, или сформированная задача поставлена некорректно по отношению к целям имитационного моделирования. Непосредственное формирование альтернатив происходит с использованием информации из баз данных АИМ, с учетом модельной ситуации, а также уже накопленных знаний. При этом реализуются алгоритмы интеллектуального анализа и поиска знаний в базах данных (DM и KDD). Найденные альтернативы (возможно несколько) и создают область решения задачи управления, в которой выбирается подмножество, характеризующее конкретную обстановку.

Блок моделирования математических схем реализует соответствующие алгоритмы математических моделей и позволяет получать численные характеристики

показателей ЛДП. Выстраивание алгоритмов моделирования подпроцессов ЛДП в последовательную цепочку запусков осуществляется с помощью блока имитации структуры лечебно-диагностического процесса. При этом сама структура ЛДП формируется на основе ситуационной схемы оператором путем включения тех или иных операций процесса с учетом модели пациента.

Модули критериального оценивания и оптимизации решений оценивают насколько эффективно управление ЛДП и возможно ли его совершенствование.

Все результаты, имеющие значение с точки зрения имитации подпроцессов ЛДП и выраженные в виде числового показателя или показателя признака (1, 0), посредством блока сбора результатов помещаются в базу данных. Этот же блок, исходя из поступающей в него информации, выполняет функции формирования знаний, которые помещаются в базу знаний. Все модули АИМ являются реентерабельными.

Таким образом, предлагаемый методический подход позволяет создать эффективный инструментарий исследований подпроцессов и операций лечебно-диагностического процесса медицинской организации, получить необходимые для его оценки значения показателей и принять решения о комплектовании инфраструктуры медицинской информационной системы.

Список литературы

1. URL: <http://www.cnews.ru/reviews/free/publichealth2011/>
2. Горбачев, Д. В. Обоснование технологии ремонта изделий РЭА боевых средств ЗРС фронтового звена за счет оптимизации структуры системы ТО и ремонта : дисс. ... канд. техн. наук : 20.02.17 : защищена 30.06.1997 / Горбачев Д. В. – СПб. , 1997. – 301 с.

О. Б. Графов

Россия, Курск, Юго-Западный государственный университет

МЕТОД И АЛГОРИТМЫ МЯГКОГО ДЕКОДИРОВАНИЯ КОДОВ РИДА-СОЛОМОНЫ ДЛЯ КОРРЕКЦИИ ОШИБОК В АРХИВНОЙ ОПТИЧЕСКОЙ ПАМЯТИ

Предлагаются метод и алгоритмы мягкого декодирования кодов Рида-Соломона для коррекции ошибок в архивной оптической памяти.

Предлагаемый метод мягкого декодирования кодов Рида-Соломона представляет собой развитие подхода к списочному декодированию кодов Рида-Соломона на основе использования синдромов, введенного Егоровым С.И. [1].

Списочное декодирование обеспечивает исправление $t = t_C + \tau$ (τ – число дополнительно исправляемых ошибок) ошибки путем аналитического продолжения алгоритма Берлекэмп-Мессис, вычисляющего полином локаторов ошибок $\Lambda^{(2t_C)}(x)$ и вспомогательный полином $B^{(2t_C)}(x)$, еще на 2τ итерации.

Аналитическое продолжение алгоритма на две итерации описывается формулой ниже:

$$\begin{aligned} \Lambda^{(2t_C+2)}(x) = & (\Lambda^{(2t_C)}(x) - \Delta_{2t_C+1} \cdot x \cdot B^{(2t_C)}(x)) - \Delta_{2t_C+2} \cdot x \cdot \\ & (\Delta_{2t_C+1}^{-1} \cdot \delta_{2t_C+1} \cdot \Lambda^{(2t_C)}(x) + x \cdot (1 - \delta_{2t_C+1}) \cdot B^{(2t_C)}(x)) \quad , \end{aligned} \quad (1)$$

где $\Delta_{2t_C+1}, \Delta_{2t_C+2}$ – неизвестные невязки, $\delta_k = 1$, если $\Delta_k \neq 0$ и $2L_{k-1} \leq k-1$, или $\delta_k = 0$ в противном случае, L_k – формальная степень многочлена локаторов ошибок $\Lambda^{(k)}(x)$.

Аналитическое продолжение на 2τ итерации может быть получено применением формулы (1) τ раз:

$$\Lambda^{(2t_C+2\tau)}(x) = f_{\Lambda^{(2t_C)}(x), B^{(2t_C)}(x)}^{\delta, \Delta}(x), \quad (2)$$

где $\delta = \{\delta_k, k = 2t_C + 1, \dots, 2t_C + 2\tau\}$, $\Delta = \{\Delta_k, k = 2t_C + 1, \dots, 2t_C + 2\tau\}$.

Каждому множеству неизвестных невязок соответствует многочлен $\Lambda^{(2t_C+2\tau)}(x)$. По определению многочлен локаторов ошибок $\Lambda(x)$ обращается в нуль, когда x принимает значения обратные к локаторам ошибочных символов. Представляют интерес такие многочлены $\Lambda^{(2t_C+2\tau)}(x)$, которые имеют степень $t_C + \tau$ и имеют точно $t_C + \tau$ корень в $GF(q)$. Множество допустимых корней многочлена $\Lambda^{(2t_C+2\tau)}(x)$ составляет $\{\alpha^0, \alpha^{-1}, \alpha^{-2}, \dots, \alpha^{-(n-2)}, \alpha^{-(n-1)}\}$ (α – примитивный элемент поля $GF(q)$). После подстановки элементов этого множества в уравнение (2), получается следующее:

$$\Lambda^{(2t_C+2\tau)}(\alpha^{-i}) = f_{\Lambda^{(2t_C)}(\alpha^{-i}), B^{(2t_C)}(\alpha^{-i})}^{\delta, \Delta}(\alpha^{-i}), \quad i = 0, 1, \dots, n-1 \quad (3)$$

где $\Lambda^{(2t_C)}(\alpha^{-i})$ и $B^{(2t_C)}(\alpha^{-i})$ – компоненты дискретного преобразования Фурье (DFT) многочленов $\Lambda^{(2t_C)}(x)$ и $B^{(2t_C)}(x)$.

В случае наличия в кодовом слове конфигурации ошибок веса $t_C + \tau$ ровно $t_C + \tau$ правых частей уравнений системы (3) для некоторого набора значений невязок $\Delta_{2t_C+1}, \Delta_{2t_C+2}, \dots, \Delta_{2t_C+2\tau}$ обращаются в 0. Для определения позиций этих ошибочных символов необходимо найти соответствующе невязки и зафиксировать номера i уравнений системы (3) с правыми частями, обратившимися в нуль.

Предлагаемый метод мягкого декодирования предполагает выбор подмножества уравнений из системы (3), соответствующих наиболее вероятным позициям ошибочных символов в кодовом слове.

Поиск неизвестных позиций ошибок (выбор уравнений) осуществляется в порядке возрастания надежностей символов принятого кодового слова. Последовательность номеров позиций символов, упорядоченных по надежности, хранится в таблице $L[]$.

В процессе поиска осуществляется перебор $(2\tau - 1)$ позиций ошибок наименее надежных символов в области ETS (ErrorTrappingScore). Для каждого набора из $(2\tau-1)$ позиций вычисляется последовательность $S_{L[i_1], L[i_2], \dots, L[i_{2\tau-1}]}$ возможных значений невязки $\Delta^{L[i_1], L[i_2], \dots, L[i_{2\tau}]}$ и осуществляется поиск значений Δ , которые встречаются в последовательности точно $(t_C+1-\tau)$:

$$\begin{aligned} & S_{L[i_1], L[i_2], \dots, L[i_{2\tau-1}]} \\ &= \{\Delta^{L[i_1], L[i_2], \dots, L[i_{2\tau}]} = \frac{F^{o1}(L[i_1], L[i_2], \dots, L[i_{2\tau}], R_{L[i_1]}, R_{L[i_2]}, \dots, R_{L[i_{2\tau}]})}{F^{o2}(L[i_1], L[i_2], \dots, L[i_{2\tau}], R_{L[i_1]}, R_{L[i_2]}, \dots, R_{L[i_{2\tau}]})}, \\ & \quad i_{2\tau} = \overline{i_{2\tau-1} + 1, n-1}\}, \end{aligned} \quad (4)$$

где

$$F^o(L[i_1], L[i_2], \dots, L[i_{2\tau}], R_{L[i_1]}, R_{L[i_2]}, \dots, R_{L[i_{2\tau}]}) =$$

$$= \sum_{\substack{\forall \{j_1, j_2, \dots, j_o\} = J \\ j_1 < j_2 < \dots < j_o \\ j_1, j_2, \dots, j_o \in \{1, 2, \dots, 2\tau\}}} \left[\prod_{\substack{\forall \{k_1, k_2\} \\ k_1 < k_2 \\ k_1, k_2 \in J}} (\alpha^{L[i_{k_1}]} + \alpha^{L[i_{k_2}]}) \prod_{\substack{\forall \{k_1, k_2\} \\ k_1 < k_2 \\ k_1, k_2 \in \{1, 2, \dots, l\} \setminus J}} (\alpha^{L[i_{k_1}]} + \alpha^{L[i_{k_2}]}) \prod_{k=1}^o R_{L[i_{j_k}]} \right],$$

$$R_i = \begin{cases} \alpha^{(2s+1)i} B^{(2t_c)}(\alpha^{-i}) / \Lambda^{(2t_c)}(\alpha^{-i}) & \text{если } s \geq 0, \\ \alpha^{(-2s-1)i} \Lambda^{(2t_c)}(\alpha^{-i}) / B^{(2t_c)}(\alpha^{-i}) & \text{в противном случае,} \end{cases}$$

$$s = t_c - L_{2t_c}, \quad o1 = \tau - |s + 1|, \quad o2 = \tau - |s|.$$

Если такие значения Δ найдены, то значениями позиций ошибок являются значения индексов $L[i_1], L[i_2], \dots, L[i_{2\tau-1}]$ этой последовательности и множество значений индекса $L[i_{2\tau}]$, соответствующего Δ . Значения ошибочных символов могут быть найдены с использованием формулы Форни.

На основе предложенного метода разработано семейство алгоритмов мягкого декодирования кодов Рида-Соломона, отличающихся реализацией перебора индексов $i_1, i_2, \dots, i_{l-1}$ и различными условиями выхода из циклов.

Наиболее очевидный алгоритм перебора индексов $i_1, i_2, \dots, i_{l-1}$, допускающий простую аппаратную реализацию, приведен ниже в виде псевдокода.

```

for i = l - 2 downto 0 do
  i_massiv[i] = i_massiv[i] + 1
  if i_massiv[i] ≠ CycleLimits[CycleLimitsPointer + i] then
    for j = 1 to l - i - 2 do
      i_massiv[i + j] = i_massiv[i] + j
    endfor
    goto MainCycle
  endif
endfor

```

Массив $i\_massiv[l]$ содержит набор индексов $i_1, i_2, \dots, i_l$, он инициализируется следующим образом:

for i = 0 to i-1 do i_massiv[i] = i.

Диапазон перебора индексов $i_1, i_2, \dots, i_{l-1}$ ограничен первыми наименее надежными n_T позициями символов из таблицы L . Для этого используется массив $CycleLimits$.

Эффективность коррекции ошибок получившегося алгоритма падает с увеличением размера списка кодовых слов, попадающих в сферу радиуса $t_c + \tau$, из-за немоного роста надежности наборов перебираемых позиций ошибок.

Более эффективный алгоритм мягкого декодирования может быть получен на основе следующего алгоритма перебора индексов $i_1, i_2, \dots, i_{l-1}$.

```

if IndSerLen > 1 then
  IndSerLen = IndSerLen - 1
  for i = 0 to IndSerLen - 1 do
    i_massiv[i] = i
  endfor

```

```

i_massiv[IndSerLen] = IndSerPos + 1
IndSerPos = IndSerLen - 1
else
IndSerPos = IndSerPos + 1
i_massiv[0] = IndSerPos
i = 1
while IndSerPos + 1 = i_massiv[i] do
    i = i + 1
    IndSerLen = IndSerLen + 1
    IndSerPos = IndSerPos + 1
    if i = l - 1 then
        break
    endif
endwhile
end if
if i_massiv[l - 2] < CycleLimits[CycleLimitsPointer + l - 2] then
goto MainCycle
end if

```

Значения управляющих переменных задаются следующим образом:

```

IndSerLen = l - 1
IndSerPos = IndSerLen - 1.

```

Разработанные алгоритмы декодирования позволяют значительно повысить эффективность применения высокоскоростных кодов Рида-Соломона в архивной оптической памяти за счет исправления дополнительных ошибок и обладают приемлемой вычислительной сложностью при небольшом числе дополнительно исправляемых ошибок ($\tau = 1, 2$).

Список литературы

1. Егоров, С. И. Коррекция ошибок в информационных каналах периферийных устройств ЭВМ : моногр. / С. И. Егоров. – Курск : Курск. гос. техн. ун-т, 2008. – 252 с.

С. И. Егоров, А. В. Кривонос

Россия, Курск, Юго-Западный государственный университет

АЛГОРИТМЫ КОРРЕКЦИИ ОШИБОК ДЛЯ ОПТИЧЕСКИХ ДИСКОВ DVD

Предложены новые алгоритмы коррекции ошибок, возникающих при воспроизведении данных с оптических дисков DVD. С помощью имитационного моделирования показана высокая эффективность исправления независимых ошибок предложенными алгоритмами.

Оптические DVD-диски остаются одними из наиболее распространенных носителей информации среди пользователей персональных компьютеров. Достоинствами этих дисков являются их дешевизна и возможность хранения достаточно большого объема информации. Однако для оптических дисков характерен относи-

тельно высокий уровень ошибок в считанной с них информации, основной причиной которых являются дефекты носителя [1].

Для коррекции ошибок, возникающих при чтении информации с дисков DVD, предусмотрено использование произведения кодов Рида-Соломона (Reed-Solomon Product Code, RSPC) [2].

Коды Рида-Соломона (РС-коды) характеризуются параметрами (n, k, d) [3], где n – длина кодового слова; k – количество информационных символов в кодовом слове и d – минимальное кодовое расстояние. При этом количество проверочных символов в слове $r = (n - k)$, и $d = r + 1$. Символы кодового слова представляют собой элементы поля Галуа $GF(q)$.

Число гарантированно исправляемых кодом ошибочных символов ограничено половиной минимального кодового расстояния $t_c = \lfloor (d - 1) / 2 \rfloor$. Для РС-кода (208, 192, 17) с символами из поля Галуа $GF(2^8)$, используемого в DVD, максимальное число гарантированно исправляемых ошибочных символов равно 8. Для РС-кода (182, 172, 11), также используемого в DVD, максимальное число гарантированно исправляемых ошибок равняется 5.

Число гарантированно исправляемых ошибочных символов (байт) произведением кодов Рида-Соломона, используемым в DVD, равно $t_c = \lfloor (d_h \cdot d_v - 1) / 2 \rfloor = 98$.

Закодированный блок данных, записанный в одном секторе диска DVD, приведен на рисунке 1. Информационные символы блока сначала кодируются РС-кодом (208, 192, 17) по вертикали, затем осуществляется кодирование РС-кодом (182, 172, 11) по горизонтали.

Используемая в контроллерах накопителей DVD процедура коррекции ошибок (процедура декодирования произведения кодов Рида-Соломона) предусматривает выполнение двух этапов.



Рис. 1. Блок данных сектора DVD, закодированный произведением кодов Рида-Соломона

Сначала осуществляется коррекция ошибок по строкам блока данных (горизонтальным кодовым словам). В каждой строке можно скорректировать до 5 байтовых ошибок. При этом исправляются случайные ошибки малой кратности. Если строки искажены протяженными пакетами ошибок, содержащими более 5 ошибок-

ных байт, коррекция ошибок в строке завершается либо ошибочным исправлением, либо обнаружением шести и более неисправимых ошибок. В последнем случае отмечается отказ от декодирования, и соответствующее горизонтальное кодовое слово помечается с помощью специального флага.

На втором этапе процедуры коррекции ошибок выполняется исправление ошибок по столбцам блока данных. При этом все символы горизонтальных кодовых слов, в которых зафиксированы неисправимые ошибки, в вертикальных кодовых словах объявляются стертыми. Если столбец имеет e стираний (из-за отказов декодирования) и t ошибок (из-за ошибочных исправлений), то этот столбец корректируем тогда и только тогда, когда $2t + e \leq d - 1$. Поскольку РС-код столбца имеет минимальное расстояние 17, то в каждом столбце можно исправить до 8 байтовых ошибок или до 16 байтовых стираний.

Для исправления ошибок и стираний в горизонтальных и вертикальных кодовых словах используются стандартные алгоритмы алгебраического декодирования кодов Рида-Соломона с жесткими решениями: алгоритм Берлекэмп-Мессе или алгоритм Евклида [2, 3].

Достоинством приведенной процедуры декодирования является эффективность исправления протяженных пакетов ошибок, небольшая задержка декодирования и относительно простая аппаратная реализация. Недостаток процедуры заключается в невысокой эффективности исправления небольших пакетов ошибок, вызванных дефектами оптического диска, что объясняется большим числом ложных стираний символов на втором этапе этой процедуры.

С учетом этого рассмотренная процедура коррекции ошибок может быть усовершенствована следующим образом.

1) На втором этапе процедуры коррекции ошибок при исправлении ошибок в вертикальных кодовых словах стирания не используются. В этом случае в вертикальных кодовых словах возможно исправление до восьми независимых ошибок. При исправлении ошибочных символов модифицируются синдромы горизонтальных кодовых слов, к которым эти символы относятся. Если число ошибок больше 8, фиксируется отказ от декодирования (при отсутствии ложной коррекции, вероятность которой для РС-кода (208, 192, 17) весьма мала). Соответствующие вертикальные кодовые слова маркируются флагами в специальном массиве.

2) В процедуру коррекции ошибок добавляется третий этап, заключающийся в исправлении ошибок в решетчатой конфигурации горизонтальных и вертикальных кодовых слов, помеченных флагами отказов от декодирования, с использованием стираний. Далее в статье этот этап будет называться финальным исправлением стираний.

Фрагмент алгоритма финального исправления стираний следует ниже.

1) $i = 0$.

2) Если $\text{DecFailHor}[i] = 1$, выполняются пп. 3–7, в противном случае осуществляется переход к п. 8.

3) $\text{NumErasures} = \text{CntDecFailVer}$. Формируются позиции стираний для i -го горизонтального кодового слова в соответствии с содержимым массива DecFailVer . Находятся позиции и значения ошибок в этом слове с использованием алгоритма Берлекэмп-Мессе с учетом NumErasures стираний.

4) Если $\text{NumErrors} > 0$, выполняются пп. 5–7, в противном случае осуществляется переход к п. 8.

5) Исправляются NumErrors ошибок в i -м горизонтальном кодовом слове. $\text{DecFailHor}[i] = 0$, значение CntDecFailHor декрементируется.

6) Если $\text{CntDecFailHor} = 0$, осуществляется переход к п. 10, в противном случае выполняется следующий пункт.

7) Выполняется модификация синдромов вертикальных кодовых слов, в которые входят исправленные символы i -го горизонтального кодового слова. Проверяются значения модифицированных синдромов, и если они становятся нулевыми, соответствующие ячейки массива DecFailVer сбрасываются в 0. Одновременно для каждого нулевого значения синдрома декрементируется счетчик CntDecFailVer .

8) $i = i + 1$.

9) Если $i = 208$, процедура коррекции ошибок завершается неудачей, в противном случае осуществляется переход к п. 2.

10) Конец: ошибки в блоке данных исправлены.

В алгоритме используются следующие переменные:

i – счетчик горизонтальных кодовых слов в блоке данных;

$\text{DecFailHor}[0:207]$ и $\text{DecFailVer}[0:171]$ – массивы флагов неудачного декодирования горизонтальных и вертикальных кодовых слов, соответственно, 1 в ячейке массива показывает неудачное декодирование слова;

CntDecFailHor и CntDecFailVer – счетчики числа неудачно декодированных горизонтальных и вертикальных кодовых слов, соответственно;

NumErasures – число стираний на входе алгоритма Берлекэмпа-Мессе;

NumErrors – число найденных ошибочных символов на выходе алгоритма Берлекэмпа-Мессе.

Сложность финального исправления стираний на порядки меньше сложности первого и второго этапов модифицированной процедуры коррекции ошибок. Это определяется двумя факторами: 1) количество декодируемых горизонтальных и вертикальных кодовых слов (с отказами от декодирования) меньше на порядок количества кодовых слов декодируемых на первых двух этапах процедуры, 2) вместо вычисления синдромов используется их модификация.

В то же время финальное исправление стираний позволяет эффективно исправлять протяженные пакеты ошибок в закодированном блоке данных.

Эффективность процедур коррекции ошибок в оптических дисках DVD исследовалась путем имитационного моделирования на ЭВМ. При этом использовалась модель канала с Гауссовым шумом (AWGN) и модуляцией BPSK.

Среднее число исправляемых ошибочных байт достигает величины 550,2 для обычной процедуры коррекции ошибок ($E_b/N_o = 6,85$) и величины 1147,1 (вдвое большей) для модифицированной процедуры ($E_b/N_o = 6,1$). Для модифицированной процедуры количество исправляемых ошибочных байт в блоке примерно в 10 раз больше величины $t_C = 98$, гарантируемой минимальный кодовым расстоянием используемого произведения кодов.

С использованием программной модели были получены зависимости BER и BLER на выходе RSPC-декодера от значений E_b/N_o (отношения энергии сигнала на информационный бит к односторонней спектральной плотности шума), показанные на рис. 2.

Цифрой 1 обозначена кривая зависимости, соответствующая обычной процедуре коррекции ошибок, цифра 2 соответствует модифицированной процедуре (только первый и второй этапы, без финального исправления стираний), цифра 3 – модифицированной процедуре (с финальным исправлением стираний), цифра 4 – «вертикально-настойчивой» процедуре коррекции ошибок, которая будет рассмотрена ниже.

Из рисунка видно, что энергетический выигрыш от кодирования, обеспечиваемый модифицированной процедурой коррекции ошибок, составляет примерно 0,8 db в сравнении с обычной процедурой коррекции ошибок.

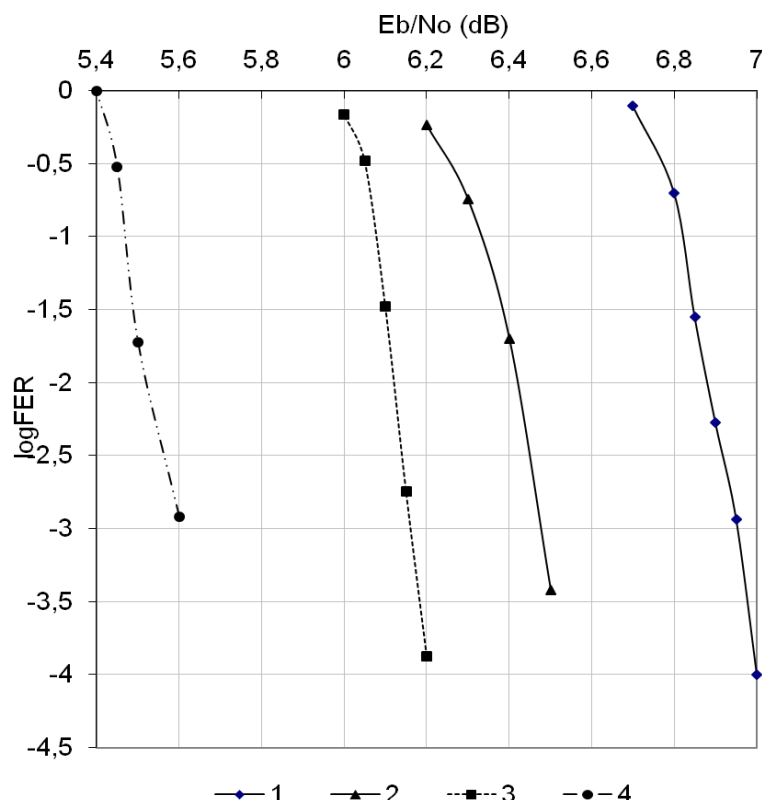


Рис. 2. Эффективность исправления ошибок в блоке данных DVD

В процессе отладки программной модели коррекции ошибок для дисков DVD было отмечено следующее.

1) Большое количество горизонтальных кодовых слов было невозможно исправить, для них фиксировались отказы от декодирования. В блоке данных наблюдались десятки таких слов ($E_b/N_o = 6,1$), что на порядок больше числа вертикальных слов с неисправимыми ошибками.

2) Заметное число горизонтальных кодовых слов исправлялось ошибочно.

Объясняется это значительно меньшим минимальным расстоянием РС-кода ($d_H = 11$), используемым для кодирования горизонтальных слов, по сравнению с минимальным расстоянием РС-кода ($d_V = 17$), используемым для вертикальных слов.

Исходя из более высокой корректирующей способности РС-кода, используемого для вертикальных слов, предлагается новая процедура коррекции ошибок в блоке данных DVD, в которой упор сделан на декодирование вертикальных кодовых слов. При этом коррекция ошибок в горизонтальных кодовых словах осуществляется только в случае невозможности декодирования вертикальных слов. Эту процедура ниже будет называться «вертикально-настойчивой».

В начале процедуры коррекции ошибок производится вычисление синдромов и исправление ошибок в вертикальных кодовых словах. Если все ошибки исправлены, то декодирование завершается. В противном случае вычисляется синдром, и исправляются ошибки в i -м горизонтальном слове ($i = 0-207$). Если ошибки произошли в информационной части слова, производятся модификации синдромов со-

ответствующих вертикальных кодовых слов и выполняются повторные попытки исправления в этих словах ошибок. Если все ошибки в блоке не исправлены, декодируется следующее горизонтальное слово. Если после перебора всех горизонтальных слов останутся ошибки, коррекция блока данных считается неудачной.

Среднее число исправляемых ошибочных байт для процедуры достигает величины 1889,1, что примерно в три раза больше чем для обычной процедуры коррекции ошибок и в два раза больше чем для модифицированной процедуры.

На рис. 2 приведена кривая зависимости BIER от E_b/N_o (кривая 4), полученная для «вертикально-настойчивой» процедурой коррекции ошибок. Из рисунка видно, что энергетический выигрыш от кодирования, обеспечиваемый этой процедурой, составляет примерно 1,4 db в сравнении с обычной процедурой коррекции ошибок и 0,6 db в сравнении с модифицированной процедурой коррекции ошибок.

Предлагаемые новые алгоритмы коррекции ошибок для оптических дисков DVD позволяют значительно повысить эффективность исправления независимых ошибок. Модифицированная и «вертикально-настойчивая» процедуры коррекции ошибок обеспечивают дополнительный энергетический выигрыш от кодирования соответственно 0,8 db и 1,4 db в сравнении с обычной процедурой коррекции ошибок.

Представленная работа поддержана грантом Президента Российской Федерации для государственной поддержки ведущих научных школ НШ-2357.2014.8.

Список литературы

1. Егоров, С. И. Коррекция ошибок в информационных каналах периферийных устройств ЭВМ / С. И. Егоров. – Курск : Курск. гос. техн. ун-т., 2008. – 252 с.
2. Chang, H. C. A Reed-Solomon Product-Code (RS-PC) Decoder Chip for DVD Applications / H. C. Chang, C. B. Shung, C. Y. Lee // IEEE Journal of Solid-State Circuits. – 2001. – Vol. 36, № 2. – С. 229–238.
3. Кларк, Д. Кодирование с исправлением ошибок в системах цифровой связи : пер. с англ. / Д. Кларк, Д. Кейн. – М.: Радио и связь, 1987. – 392 с.

С. А. Жерновой

Россия, Пенза, Пензенский государственный университет

ПРОТОКОЛЫ МАРШРУТИЗАЦИИ БЕСПРОВОДНЫХ МОБИЛЬНЫХ СЕТЕЙ (MANET)

Приводится классификация существующих протоколов маршрутизации беспроводных мобильных сетей (MANET).

На сегодняшний день одним из динамично развивающихся направлений в беспроводных вычислительных сетях является создание телекоммуникационных систем с переменной топологией сети, такие сети получили наименование ad-hoc сети (от лат. – для данного случая). Одним из видов ad-hoc сетей являются мобильные ad-hoc сети (MANET- mobile ad-hoc networks)

MANET – это одноранговые сети, состоящие из множества мобильных узлов, соединенных беспроводными каналами связи, объединение которых формирует

произвольную топологию. Узлы могут свободно передвигаться в любом направлении и, поэтому, такие сети являются самоорганизующимися, поскольку их узлы являются не только конечными пользовательскими терминалами, но и являются ретрансляторами-маршрутизаторами, ретранслируя пакеты других абонентов и участвуя в нахождении маршрутов к ним. В виду такой динамичности структуры сети и отсутствия централизованного управления, каждый беспроводной узел в этой сети должен самостоятельно находить наилучший маршрут для передачи данных другим узлам и вопрос выбора маршрута доставки данных в таких сетях становится первостепенным.

Для успешного применения в ad-hoc сетях протоколы маршрутизации должны обладать следующими качествами:

1. Обеспечивать распределенность узлов сети. Все узлы в сети должны быть способны осуществлять маршрутизацию;
2. Обеспечивать высокую надежность доставки пакетов в условиях постоянно изменяющейся топологии сети;
3. Обеспечивать малое время построения маршрута и его быстрой сходимостью в условиях постоянно изменяющейся топологии сети.
4. Обладать механизмами оперативного обнаружения разрыва маршрута и его восстановления, а также не допускать формирования петель в маршрутах.
5. Обладать высокой масштабируемостью, т.е. обеспечивать высокую производительность сети при различных ее размерах.
6. Обеспечивать минимальную загрузку сети служебной информацией;
7. Поддерживать качество обслуживания (QoS).

Существующие на данный момент протоколы маршрутизации можно классифицировать по **принципу работы** (рис. 1):

1) Проактивные или табличные (англ. *proactive, table-driven*). Периодически рассылают по сети служебные сообщения с информацией обо всех изменениях в ее топологии. В результате каждый узел сети на основе данной информации строит маршруты до всех остальных узлов и сохраняет их в таблицу маршрутизации, откуда они считываются при необходимости передачи сообщения какому-либо адресату.

2) Реактивные или работающие по запросу (англ. *reactive, on-demand*). Составляют маршруты до конкретных узлов лишь при возникновении необходимости в передаче им информации. Для этого узел-отправитель широковещательно рассылает по сети сообщение-запрос, которое должно дойти до узла-адресата. В ответ адресат высылает сообщение-подтверждение, из которого отправитель узнает необходимый маршрут и записывает его в свою таблицу маршрутизации. Для повторных отправок сообщений данному адресату маршрут просто считывается из таблицы. Если обнаруживается его разрушение, то запускается так называемая процедура поддержания маршрута, которая фактически заключается в поиске нового маршрута до адресата.

3) Гибридные (англ. *hybrid*). Данные протоколы комбинируют механизмы проактивных и реактивных протоколов. Как правило, они разбивают сеть на множество подсетей, внутри которых функционирует проактивный протокол, а взаимодействие между ними осуществляется реактивными методами. В крупных сетях это позволяет сократить размеры таблиц маршрутизации, которые ведут узлы сети, т.к. им необходимо знать точные маршруты лишь для узлов подсети, к которой они принадлежат. Также сокращается и объем рассылаемой по сети служебной информации, так как основная ее часть распространяется лишь в пределах подсетей.

По критерию **определения оптимальности маршрута**:

1) Протоколы вектора расстояния (англ. distance-vector, hop-count). Всегда считают оптимальным маршрут, содержащий наименьшее число хопов (ретрансляций пакета) между отправителем и адресатом.

2) Протоколы со сложной метрикой маршрутов или протоколы состояния каналов (англ. link-state). Используют комплексную оценку маршрутов по нескольким параметрам, в которые, помимо числа хопов, обычно входят задержка на доставку пакета, пропускная способность канала и др.

По **типу используемых для маршрутизации данных**:

1) Топологические. Используют информацию о существующих сетевых соединениях между узлами сети.

2) Географические. Используют данные о географическом положении узлов, получаемые обычно посредством спутниковой навигации, для прогнозирования возможности или невозможности связи между отдельными узлами и о возможных в данной сети маршрутах.

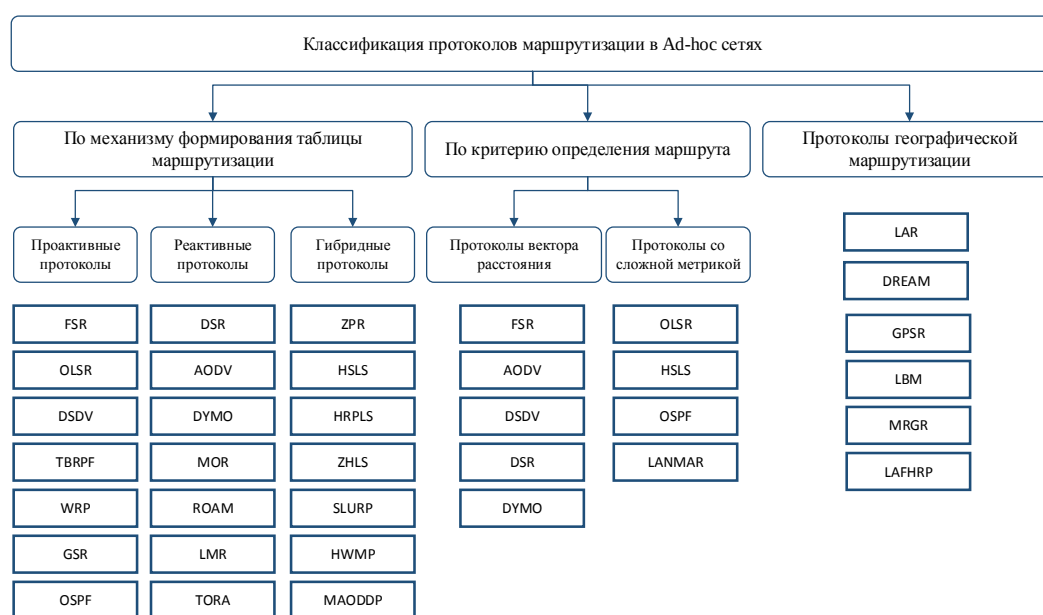


Рис. 1. Классификация протоколов маршрутизации

В настоящее время наиболее часто используемыми протоколами в ad-hoc сети являются:

1) **AODV** (Ad hoc On-Demand Distance Vector). Реактивный протокол маршрутизации, разработанный для использования в сетях различного размера. Как следует из названия, для вычисления маршрутов используется дистанционно-векторный алгоритм маршрутизации.

2) **OLSR** (Optimized Link-State Routing). – Проактивный протокол маршрутизации, использующий обмен сообщениями приветствия и контроля для получения информации о топологии сети. Узлы используют эту информацию для определения следующего прыжка в пути маршрутизируемого пакета.

3) **HWMP** (Hybrid Wireless Mesh Protocol) – гибридный протокол, созданный на основе хорошо известного протокола дистанционно-векторной маршрутизации по запросу (AODV). Однако в HWMP механизмы маршрутизации работают на MAC-уровне, где доступна информация о соседних узлах и условиях беспроводной передачи, что делает алгоритмы маршрутизации более эффективными. HWMP объединяет в себе два режима построения путей: реактивный и проактивный, которые

могут быть использованы как отдельно, так и одновременно в одной сети. При этом используются широковещательные пакеты.

Основной решаемой задачей для данного класса сетей является поиск оптимального пути передачи информации (маршрута) в условиях непрерывно изменяющейся топологии связей (графа связи сети) с использованием распределенного алгоритма при отсутствии единого центра управления. Каждый класс протоколов потенциально имеет свои преимущества и недостатки при использовании в условиях мобильных Ad hoc-сетей. Например, проактивные протоколы обладают преимуществом над реактивными во времени построения маршрута и минимальной задержкой отправки пакетов узлам, но загружают значительную часть пропускной способности сети для поддержания текущей маршрутной информации. В свою очередь реактивные требуют существенно меньшей пропускной способности сети, чем проактивные, но вносят значительную задержку на стадии определения маршрутов между конкретными узлами, так как необходимо разослать широковещательный запрос и дождаться подтверждения от адресата.

Список литературы

1. Маршрутизация в беспроводных мобильных Ad hoc-сетях / В. М. Винокуров, А. В. Пуговкин, А. А. Пшенников, Д. Н. Ушарова, А. С. Филатов // Доклады ТУСУРа, 2010. – № 2 (22). – Ч. 1. – С. 288–292.
2. Орлов, В. Г. Протоколы маршрутизации в мобильных AD-HOC-сетях / В. Г. Орлов, А. Н. Фадеев // Материалы Международной научно-технической конференции, 3–7 декабря 2012 г.

Н. С. Карамышева

Россия, Пенза, Пензенский государственный университет

ОРГАНИЗАЦИЯ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ В ВЫЧИСЛИТЕЛЬНОЙ СЕТИ С ТОПОЛОГИЕЙ ТИПА «ЛОГИЧЕСКОЕ КОЛЬЦО»

Даны предложения по организации межагентных взаимодействий в сетевой среде при реализации систем управления распределенными реляционными базами данных с горизонтально фрагментированными отношениями, что повышает эффективность метакомпьютерной реализации процессов обработки структурированных данных. Продемонстрирована возможность применения модели асинхронных предикатных сетей для согласования процессов и агентов при организации обработки структурированных данных. Предложенные методы и алгоритмы распределенной обработки данных также могут быть реализованы и на многоядерных процессорах, в системах кластерного типа, ориентированных не только на взаимодействия агентов, но и на передачу сообщений.

Введение

В связи с интенсивным развитием облачных и грид-технологий, включая и технологии сетевого хранения данных (storage grid), наблюдается быстрое количественное и качественное развитие сетевых систем управления базами данных

(СУБД). Поэтому актуальна разработка СУБД, работающих на всем спектре параллельных архитектур от многоядерных процессоров до грид-сетей. Современные языки параллельного и распределенного программирования, например, C#, MS#, технология CUDA [1, 2, 3] и другие позволяют проектировать распределенные приложения и программные системы обработки баз данных. Известно, что при проектировании конкретной архитектуры сетевой или многомашинной СУБД может быть предусмотрено выполнение следующих параллельных операций:

- параллельная обработка данных в центральной (клиентской) ЭВМ и в облачном или грид-сервисе, реализующем функции машины базы данных (МБД);
- анализ запроса и обработка данных;
- доступ к данным и обработка;
- передача данных и обработка;
- параллельный доступ к данным и параллельная обработка данных;
- параллельная обработка данных.

Последние пять пунктов соответствуют работе [4], а первый несколько модифицирован нами в соответствии с современными тенденциями развития новых информационных технологий. В этой работе полагается, что первые четыре операции скорее следует считать конвейерной обработкой данных.

Реализация конвейерной обработки возможна в различных проектах, например, в проекте МБД университета Карнеги-Меллона (обработка данных при помощи систолических матриц), в МБД DIRECT Висконсинского университета, в МБД DBC (Database Computer for Very Large Database) университета штата Огайо и особенно эффективно реализована в машине GRACE Токийского университета [4]. Многосерверная сетевая база данных с циркулирующей информацией рассмотрена также в работе [5], где предложена организация локальной вычислительной сети, работающей по принципу «клиент-сервер» при удовлетворении запросов от рабочих станций к сегментированной базе данных, находящейся на нескольких серверах.

Целью настоящей работы является описание реализации кольцевой обработки сегментов реляционной базы данных в сетевой агентно-ориентированной среде. При этом без ограничения общности предложенные методы и алгоритмы распределенной обработки данных могут быть реализованы и на многоядерных процессорах, а также в системах кластерного типа, ориентированных не на коллектив взаимодействующих агентов, а на передачу сообщений.

Общая схема выполнения операций над сегментами отношений

На рис. 1 представлена общая схема организации выполнения операций реляционной алгебры над фрагментированными отношениями реляционной базы данных.

Агенты A, B, C, D, E и F выполняют глобальные операции реляционной алгебры, осуществляя при необходимости конвейерный или другие разновидности коллективного обмена данными друг с другом. Основной обмен данными происходит в двунаправленном логическом кольце агентов. В тексте данной работы использован прямой шрифт для имен агентов A, B, C, D, E, F для того, чтобы отличать эти имена от имен A, B и C отношений.

Масштабируемость агентно-ориентированной схемы реализации кольца для другого числа агентов (логических узлов метакomпьютера) очевидна. Поэтому для компактности изложения ограничимся шестью агентами и, соответственно, шестью фрагментами каждого из исходных и промежуточных отношений.

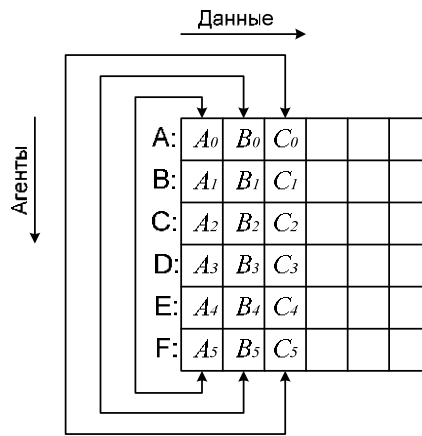


Рис. 1. Общая схема логической организации агентно-ориентированной метакomпьютерной системы для распределенных баз данных

Выполнение операций реляционной алгебры над сегментами отношений в логическом кольце

Предложим основные схемы выполнения бинарных и унарных операций реляционной алгебры, определенных над фрагментами $A_0, A_1, A_2, A_3, A_4, A_5$ и $B_0, B_1, B_2, B_3, B_4, B_5$ – непересекающимися подмножествами кортежей отношений A и B соответственно. Определения и алгоритмы выполнения операций реляционной алгебры для нефрагментированных отношений хорошо известны [6, 7, 8, 9] и являются основой для описанной ниже агентно-ориентированной реализации. Ее отличие и новизна состоит в распределенной реализации и организации межагентных взаимодействий в процессе обработки фрагментов отношений. Реализация операций в подобной системе является, по существу, расширением интерфейса передачи сообщений MPI [10, 11]. В работе [11] рассмотрена сетевая реализация интерфейса передачи сообщений MPI как облачного сервиса в агентно-ориентированных метакomпьютерных системах. Основные варианты обмена сообщениями между процессами описаны асинхронными предикатными сетями, ранее предложенными в работе [12].

Бинарная операция объединения « $\cup$ » выполняется над отношениями A и B с одинаковой схемой. Результат C имеет такую же схему. Агенты A, B, C, D, E и F метакomпьютерной системы при реализации данной операции выполняют следующие действия над фрагментами отношений A и B :

– формирование частичных результатов:

$$A: C_0 = A_0 \cup B_0;$$

$$B: C_1 = A_1 \cup B_1;$$

$$C: C_2 = A_2 \cup B_2;$$

$$D: C_3 = A_3 \cup B_3;$$

$$E: C_4 = A_4 \cup B_4;$$

$$F: C_5 = A_5 \cup B_5;$$

– формирование окончательного результата:

$$A: C = (((C_0 \cup C_1) \cup C_2) \cup C_3) \cup C_4 \cup C_5.$$

В формуле для окончательного результата C расстановка скобок указывает на порядок выполнения операции объединения частичных результатов агентом A . При этом необходимые фрагменты C_1, C_2, C_3, C_4 и C_5 передаются агенту A либо в про-

цессе кольцевого сдвига данных, либо агент A поочередно запрашивает частичные результаты C_1, C_2, C_3, C_4 и C_5 и затем их получает последовательно после очередного запроса от агентов B, C, D, E и F соответственно. На этом выполнение операции $C=A \cup B$ завершается.

Бинарная операция пересечения « $\cap$ » аналогично выполняется над двумя операндами A и B с одинаковой схемой и результат C имеет такую же схему. На основании известного в теории множеств свойства дистрибутивности операции пересечения относительно операции объединения получим следующую схему выполнения операции пересечения над фрагментами исходных отношений:

– формирование частичных результатов в кольце агентов:

$$A: C_0 = (((((A_0 \cap B_0) \cup (A_0 \cap B_1)) \cup (A_0 \cap B_2)) \cup (A_0 \cap B_3)) \cup (A_0 \cap B_4)) \cup (A_0 \cap B_5);$$

$$B: C_1 = (((((A_1 \cap B_1) \cup (A_1 \cap B_2)) \cup (A_1 \cap B_3)) \cup (A_1 \cap B_4)) \cup (A_1 \cap B_5)) \cup (A_1 \cap B_0);$$

$$C: C_2 = (((((A_2 \cap B_2) \cup (A_2 \cap B_3)) \cup (A_2 \cap B_4)) \cup (A_2 \cap B_5)) \cup (A_2 \cap B_0)) \cup (A_2 \cap B_1);$$

$$D: C_3 = (((((A_3 \cap B_3) \cup (A_3 \cap B_4)) \cup (A_3 \cap B_5)) \cup (A_3 \cap B_0)) \cup (A_3 \cap B_1)) \cup (A_3 \cap B_2);$$

$$E: C_4 = (((((A_4 \cap B_4) \cup (A_4 \cap B_5)) \cup (A_4 \cap B_0)) \cup (A_4 \cap B_1)) \cup (A_4 \cap B_2)) \cup (A_4 \cap B_3);$$

$$F: C_5 = (((((A_5 \cap B_5) \cup (A_5 \cap B_0)) \cup (A_5 \cap B_1)) \cup (A_5 \cap B_2)) \cup (A_5 \cap B_3)) \cup (A_5 \cap B_4);$$

– формирование окончательного результата:

$$A: C = (((C_0 \cup C_1) \cup C_2) \cup C_3) \cup C_4) \cup C_5.$$

Здесь при формировании частичных результатов C_0, C_1, C_2, C_3, C_4 и C_5 фрагменты B_0, B_1, B_2, B_3, B_4 и B_5 отношения B сдвигаются по логическому кольцу, передаваясь от одного агента к другому, причем порядок выполнения операций задается расстановкой скобок на основе известного в теории множеств свойства ассоциативности операции объединения в выражениях C_0, C_1, C_2, C_3, C_4 и C_5 для частичных результатов. Окончательный результат $C=A \cap B$ формируется агентом A так же, как и для предыдущей операции объединения.

Бинарная операция разности « $-$ » позволяет получить множество $C=A-B$ кортежей, принадлежащих отношению A и не принадлежащих отношению B . Здесь требуется, чтобы отношения имели одну и ту же схему. Для фрагментированных отношений в агентно-ориентированном метакомпьютере выполнение данной операции может быть организовано в соответствии со следующими выражениями:

– формирование частичных результатов в кольце агентов:

$$A: C_0 = (((((A_0 - B_0) - B_1) - B_2) - B_3) - B_4) - B_5;$$

$$B: C_1 = (((((A_1 - B_1) - B_2) - B_3) - B_4) - B_5) - B_0;$$

$$C: C_2 = (((((A_2 - B_2) - B_3) - B_4) - B_5) - B_0) - B_1;$$

$$D: C_3 = (((((A_3 - B_3) - B_4) - B_5) - B_0) - B_1) - B_2;$$

$$E: C_4 = (((((A_4 - B_4) - B_5) - B_0) - B_1) - B_2) - B_3;$$

$$F: C_5 = (((((A_5 - B_5) - B_0) - B_1) - B_2) - B_3) - B_4;$$

– формирование окончательного результата:

$$A: C = (((C_0 \cup C_1) \cup C_2) \cup C_3) \cup C_4) \cup C_5.$$

Например, в результате выполнения агентом A действий, предписанных формулой для частичного результата C_0 , из фрагмента A_0 отношения A удаляются поочередно кортежи, принадлежащие еще и фрагментам B_0, B_1, B_2, B_3, B_4 и B_5 отношения B . Аналогичные действия после каждого кольцевого сдвига выполняются остальными агентами. В результате ни один фрагмент из фрагментов отношения A не будет содержать ни одного кортежа, принадлежащего отношению B . После объ-

единения частичных результатов C_0, C_1, C_2, C_3, C_4 и C_5 агентом A результат $C=A \cup B$ будет находиться в зоне ответственности данного агента.

Выбор σ_α , или селекция, – это унарная операция над отношением. Пусть α – допустимая формула, выражающая предикат выборки, т.е. допустимая логическая формула с предикатными символами сравнения, используемыми в качестве логических переменных. Сравняться могут как значения атрибутов с некоторыми константами, так и значения различных атрибутов между собой. Результатом является отношение, содержащее кортежи, для которых формула α истинна. При реализации данной операции агенты A, B, C, D, E и F в метакомпьютерной системе выполняют соответственно следующие действия над фрагментами A_0, A_1, A_2, A_3, A_4 , и A_5 отношения A :

– формирование частичных результатов:

$$A: C_0 = \sigma_\alpha(A_0);$$

$$B: C_1 = \sigma_\alpha(A_1);$$

$$C: C_2 = \sigma_\alpha(A_2);$$

$$D: C_3 = \sigma_\alpha(A_3);$$

$$E: C_4 = \sigma_\alpha(A_4);$$

$$F: C_5 = \sigma_\alpha(A_5);$$

– формирование окончательного результата:

$$A: C = (((C_0 \cup C_1) \cup C_2) \cup C_3) \cup C_4 \cup C_5.$$

В результате выполнения **бинарной операции « $\times$ » декартова произведения** двух отношений A и B со схемами R и S соответственно будет сформировано отношение со схемой, являющейся конкатенацией схем R и S , содержащее кортежи, образованные всеми возможными конкатенациями кортежей отношений A и B . Порядок выполнения операции декартова произведения, определенный над фрагментами отношений A и B , задается следующими выражениями:

– формирование частичных результатов в кольце агентов:

$$A: C_0 = (((((A_0 \times B_0) \cup (A_0 \times B_1)) \cup (A_0 \times B_2)) \cup (A_0 \times B_3)) \cup (A_0 \times B_4)) \cup (A_0 \times B_5);$$

$$B: C_1 = (((((A_1 \times B_1) \cup (A_1 \times B_2)) \cup (A_1 \times B_3)) \cup (A_1 \times B_4)) \cup (A_1 \times B_5)) \cup (A_1 \times B_0);$$

$$C: C_2 = (((((A_2 \times B_2) \cup (A_2 \times B_3)) \cup (A_2 \times B_4)) \cup (A_2 \times B_5)) \cup (A_2 \times B_0)) \cup (A_2 \times B_1);$$

$$D: C_3 = (((((A_3 \times B_3) \cup (A_3 \times B_4)) \cup (A_3 \times B_5)) \cup (A_3 \times B_0)) \cup (A_3 \times B_1)) \cup (A_3 \times B_2);$$

$$E: C_4 = (((((A_4 \times B_4) \cup (A_4 \times B_5)) \cup (A_4 \times B_0)) \cup (A_4 \times B_1)) \cup (A_4 \times B_2)) \cup (A_4 \times B_3);$$

$$F: C_5 = (((((A_5 \times B_5) \cup (A_5 \times B_0)) \cup (A_5 \times B_1)) \cup (A_5 \times B_2)) \cup (A_5 \times B_3)) \cup (A_5 \times B_4);$$

– формирование окончательного результата:

$$A: C = (((C_0 \cup C_1) \cup C_2) \cup C_3) \cup C_4 \cup C_5.$$

Данными выражениями определяются сдвиги и передачи данных в логическом кольце агентов при формировании результата $C=A \times B$.

Бинарная операция соединения определяется на основе операций декартова произведения и выборки как $\sigma_\alpha(A \times B)$ [7, 9], где α – формула, содержащая предикаты сравнения атрибутов из отношений A и B . Схема выполнения операции соединения соответствует схеме выполнения операции декартова произведения при учете условия, согласно которому в результирующее отношение включаются лишь те кортежи, которые удовлетворяют определенному соотношению между атрибутами соединения отношений A и B :

– формирование частичных результатов в кольце агентов:

$$A: C_0 = ((((\sigma_\alpha(A_0 \times B_0) \cup \sigma_\alpha(A_0 \times B_1)) \cup \sigma_\alpha(A_0 \times B_2)) \cup \sigma_\alpha(A_0 \times B_3)) \cup$$

$$\cup \sigma_{\alpha}(A_0 \times B_4)) \cup \sigma_{\alpha}(A_0 \times B_5);$$

$$B: C_1 = (((\sigma_{\alpha}(A_1 \times B_1) \cup \sigma_{\alpha}(A_1 \times B_2)) \cup \sigma_{\alpha}(A_1 \times B_3)) \cup \sigma_{\alpha}(A_1 \times B_4)) \cup \sigma_{\alpha}(A_1 \times B_5) \cup \sigma_{\alpha}(A_1 \times B_0);$$

$$C: C_2 = (((\sigma_{\alpha}(A_2 \times B_2) \cup \sigma_{\alpha}(A_2 \times B_3)) \cup \sigma_{\alpha}(A_2 \times B_4)) \cup \sigma_{\alpha}(A_2 \times B_5)) \cup \sigma_{\alpha}(A_2 \times B_0) \cup \sigma_{\alpha}(A_2 \times B_1);$$

$$D: C_3 = (((\sigma_{\alpha}(A_3 \times B_3) \cup \sigma_{\alpha}(A_3 \times B_4)) \cup \sigma_{\alpha}(A_3 \times B_5)) \cup \sigma_{\alpha}(A_3 \times B_0)) \cup \sigma_{\alpha}(A_3 \times B_1) \cup \sigma_{\alpha}(A_3 \times B_2);$$

$$E: C_4 = (((\sigma_{\alpha}(A_4 \times B_4) \cup \sigma_{\alpha}(A_4 \times B_5)) \cup \sigma_{\alpha}(A_4 \times B_0)) \cup \sigma_{\alpha}(A_4 \times B_1)) \cup \sigma_{\alpha}(A_4 \times B_2) \cup \sigma_{\alpha}(A_4 \times B_3);$$

$$F: C_5 = (((\sigma_{\alpha}(A_5 \times B_5) \cup \sigma_{\alpha}(A_5 \times B_0)) \cup \sigma_{\alpha}(A_5 \times B_1)) \cup \sigma_{\alpha}(A_5 \times B_2)) \cup \sigma_{\alpha}(A_5 \times B_3) \cup \sigma_{\alpha}(A_5 \times B_4);$$

– формирование окончательного результата:

$$A: C = (((C_0 \cup C_1) \cup C_2) \cup C_3) \cup C_4 \cup C_5.$$

Особенности вариантов операции соединения, состоящие в отбрасывании некоторых атрибутов, учитываются при реализации оператора σ_{α} .

Проекция π_X является **унарной операцией** над отношением. Пусть R – схема отношения A , а X – подмножество R . Операция проекции позволяет получить отношение $C = \pi_X(A)$ путем исключения из кортежей исходного отношения значений атрибутов, соответствующих атрибутам в $R \setminus X$ и последующего исключения повторяющихся кортежей. При реализации проекции агенты A , B , C , D , E и F в метакомпьютерной системе выполняют соответственно следующие действия над фрагментами A_0 , A_1 , A_2 , A_3 , A_4 , и A_5 отношения A :

– формирование частичных результатов:

$$A: C_0 = \pi_X(A_0);$$

$$B: C_1 = \pi_X(A_1);$$

$$C: C_2 = \pi_X(A_2);$$

$$D: C_3 = \pi_X(A_3);$$

$$E: C_4 = \pi_X(A_4);$$

$$F: C_5 = \pi_X(A_5);$$

– формирование окончательного результата:

$$A: C = (((C_0 \cup C_1) \cup C_2) \cup C_3) \cup C_4 \cup C_5.$$

На этапе получения окончательного результата при выполнении объединения промежуточных результатов агентом A естественно исключаются оставшиеся дубликаты кортежей.

Заключение

Основным результатом данной работы являются предложения по организации межагентных взаимодействий в сетевой среде при реализации систем управления распределенными реляционными базами данных, что повышает эффективность метакомпьютерной реализации процессов обработки структурированных данных. Предложенные методы и алгоритмы распределенной обработки данных могут быть реализованы и на многоядерных процессорах, а также в системах кластерного типа, ориентированных не на агенты, а на передачу сообщений.

Статья подготовлена в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития НТК России на 2014-2020 г.» (Соглашение № 14.574.21.0045 от 19.06.2014).

Список литературы

1. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5 / Э. Троелсен, – М. : Вильямс, 2013. – 1311 с.
2. Guzev, V. Asynchronous parallel programming language based on the Microsoft .NET platform / V. Guzev, Y. Serdyuk // PaCT-2003, Lecture Notes in Computer Science. – Springer, 2003. – Vol. 2763. – P. 236–243.
3. Параллельные вычисления на GPU. Архитектура и программная модель CUDA / А. Боресков, А. Харламов, Н. Марковский [и др.]. – М. : Изд-во МГУ, 2012. – 336 с.
4. Компьютеры на СБИС : в 2 кн. / Т. Мотоока, Х. Хорикоси, М. Сакауты и др. – М. : Мир, 1988. – Кн. 2. – 366 с.
5. Барский, А. Б. Применение SPMD-технологии при построении сетевых баз данных с циркулирующей информацией / А. Б. Барский // Информационные технологии. – 2004. – № 7. – С. 62–65.
6. Мейер, Д. Теория реляционных баз данных / Д. Мейер. – М. : Мир, 1987. – 608 с.
7. Ульман, Дж. Основы систем баз данных / Дж. Ульман. – М. : Финансы и статистика, 1983. – 334 с.
8. Codd, E. F. A relational model of data for large shared data banks / E. F. Codd // Comm. ACM. – 1970. – № 13(6), – P. 377–387.
9. Чери, С. Логическое программирование и базы данных / С. Чери, Г. Готлоб, Л. Танка. – М. : Мир, 1992. – 352 с.
10. PACX-MPI: The Grid-computing library PACX-MPI extending MPI for computational Grid. – URL: <http://www.hlr.de/organization/av/amt/research/pacx-mpi/>
11. Вашкевич, Н. П. О реализации интерфейса передачи сообщений как облачного сервиса в агентно-ориентированных метакомпьютерных системах / Н. П. Вашкевич, С. А. Зинкин, Н. С. Карамышева // Инфокоммуникационные технологии. – 2013. – № 4. – С. 42–53.
12. Зинкина, Н. С. Методы и модели логического управления дискретными процессами в распределенных вычислительных системах на основе концепции согласования / Н. С. Зинкина // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2011. – № 1. – С. 35–47.

А. Е. Карпов

Россия, Пенза, Пензенский государственный университет

РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ УНИФИКАЦИИ ФУНКЦИЙ НА ОСНОВЕ ЭКВИВАЛЕНТНЫХ КОНСТРУКЦИЙ ЦИКЛОВ

Рассматриваются семантически эквивалентные конструкции циклов как основа для расширения возможностей унификации функции в целях автоматизированного устранения дублирования кода.

Дублирование кода – одна из самых распространенных причин проведения рефакторинга [1]. Исследования показывают, что дублирование кода затрудняет сопровождаемость программных систем и повышает вероятность внесения ошибок [2–3]. В настоящее время разработаны различные методы и реализованы инструментальные средства для обнаружения клонов в исходном коде [4]. Однако проблема устранения дублирования кода путем автоматизированного рефакторинга не была достаточно исследована.

Существующие методы и средства позволяют устранять тривиальные и определенные типы более сложных клонов кода, однако имеют ограничения [5]. В рабо-

те [6] Н. Тсантилис (Nikolaos Tsantilis) и Г. П. Кришнан (Giri Panamoottil Krishnan) представили подход, позволяющий находить оптимальное отображение между функциями, содержащими клоны. Тем не менее, разработанный авторами метод не учитывает семантики некоторых синтаксических конструкций языков программирования, вследствие чего некоторые семантически эквивалентные фрагменты программ рассматриваются как отличающиеся. Данное ограничение приводит к тому, что устранение определенных клонов становится невозможным. В данной работе рассматривается подход, позволяющий расширить пространство потенциально устранимых клонов.

Базовый подход заключается в применении трех последовательных шагов к двум входным функциям. На первом шаге выполняется поиск идентичных управляющих конструкций. На втором шаге строится оптимальное отображение операторов между анализируемыми функциями. На третьем шаге выполняется попытка унифицировать выявленные отличия. Новизна предлагаемого подхода заключается в том, в использовании семантически эквивалентных конструкций для построения оптимального отображения.

Рассмотрим несколько примеров семантически эквивалентных конструкций. Пусть $C : S \rightarrow M$ – функция из синтаксиса в денотационную семантику. В качестве синтаксиса возьмем синтаксис языка C#.

Эквивалентность for-while циклов

Для цикла со счетчиком i , начальным значением счетчика $init$, условием $cond$ и выражением инкрементирования счетчика $incr$ и телом цикла $Body$ выполняется следующее равенство семантик for- и while-циклов:

$$C \left[\begin{array}{l} \text{for } (T \ i = \text{init}; \text{cond}(i); i = \text{incr}(i)) \\ \{ \\ \quad \text{Body} \\ \} \end{array} \right] = C \left[\begin{array}{l} \{ \\ \quad T \ i = \text{init}; \\ \quad \text{while } (\text{cond}(i)) \\ \quad \{ \\ \quad \quad \text{Body} \\ \quad \quad i = \text{incr}(i); \\ \quad \} \\ \} \end{array} \right] \quad (1)$$

Эквивалентность foreach-while циклов

Для циклов по элементам коллекции $collection$ с телом $Body$ справедливо:

$$C \left[\begin{array}{l} \text{foreach } (T \ x \text{ in } xs) \\ \{ \\ \quad \text{Body} \\ \} \end{array} \right] = C \left[\begin{array}{l} \text{var enumerator} = xs.GetEnumerator(); \\ \text{while } (\text{enumerator.MoveNext}()) \\ \{ \\ \quad T \ x = (T) \text{enumerator.Current}; \\ \quad \text{Body} \\ \} \end{array} \right] \quad (2)$$

Рассмотрим частный случай цикла *foreach* для коллекций, поддерживающих получение элемента по индексу и количества элементов (т.е. коллекций реализующих интерфейс *IReadOnlyCollection<T>*). Метод *GetEnumerator* в этом случае возвращает объект, концептуально описываемый классом *IndexedEnumerator<T>* представленным ниже в листинге.

```

1 class IndexedEnumerator<T>(IReadOnlyList<T> list) : IEnumerator<T> {
2     private int index = 0;
3     private IReadOnlyList<T> items = list;
4     private T current;
5     public bool MoveNext() {
6         if (index < items.Count) {
7             current = items[index];
8             index = index + 1;
9             return true;
10        }
11        return false;
12    }
13    public T Current => current;
14 }

```

Листинг – Концептуальный код итератора IReadOnlyList<T>

Принимая во внимание, что *MoveNext* возвращает *true* только в ветке на строках 6–10, применение уравнения (2) совместно с рефакторингом встраивания метода (Inline Method) [7] дает:

$$C \left[\begin{array}{l} \text{foreach } (T \ x \text{ in } xs) \\ \{ \\ \quad \text{Body} \\ \} \end{array} \right] = C \left[\begin{array}{l} \text{int index} = 0; \\ \text{while } (\text{index} < xs.Count) \\ \{ \\ \quad T \ x = xs[\text{index}]; \\ \quad \text{Body} \\ \quad \text{index} = \text{index} + 1; \\ \} \end{array} \right] = C \left[\begin{array}{l} \text{for } (\text{int } i = 0; i < xs.Count; i = i + 1) \\ \{ \\ \quad [x := xs[i]] \text{Body} \\ \} \end{array} \right]$$

Запись $[x := xs[i]] \text{Body}$ означает, что все вхождения x в *Body* заменяются на $xs[i]$. Таким образом, для коллекций с индексатором цикл *foreach* всегда может быть преобразован в цикл *for*, а цикл *for* может быть заменен на цикл *foreach*, если тело цикла соответствует правилам тела цикла *foreach* (в теле цикла текущий элемент используется только для чтения, нет обращений к другим элементам коллекции, тело цикла не изменяет счетчик).

Более того в формуле выше выражение инкрементации счетчика может быть заменено на операции пре- или постинкремента, часто используемые при написании циклов со счетчиком.

Для преинкремента очевидно: $C[+ i] = C[i = i + 1]$

Для постинкремента:

$$C[i ++] = C[\text{postIncr}(\text{ref } i)], \text{ где } C[\text{postIncr}(\text{ref } i)] = C \left[\begin{array}{l} \text{int postIncr}(\text{ref int } i) \\ \{ \\ \quad \text{var temp} = i; \\ \quad i = i + 1; \\ \quad \text{return temp;} \\ \} \end{array} \right]$$

$$C \left[\begin{array}{l} \text{for}(\text{int } i = v; \text{cond}(i); i++) \\ \{ \\ \text{Body} \\ \} \end{array} \right] = C \left[\begin{array}{l} \{ \\ \text{int } i = v; \\ \text{while}(\text{cond}(i)) \\ \{ \\ \text{Body} \\ \text{var temp} = i; \\ i = i + 1; \\ \} \\ \} \end{array} \right] = C \left[\begin{array}{l} \text{for}(\text{int } i = v; \text{cond}(i); i = i + 1) \\ \{ \\ \text{Body} \\ \} \end{array} \right]$$

Выше были рассмотрены лишь несколько примеров эквивалентных конструкций циклов. Множество уравнений можно получить используя конструкций язык интегрированных запросов LINQ (Language Integrated Query). Эти и другие эквивалентные преобразования могут быть применены на шаге построения оптимального отображения между анализируемыми функциями. Таким образом, становится возможным устранение отличий в синтаксисе, при сохранении исходной семантики. Несмотря на простоту представленных уравнений, программная реализация инструмента выполняющего унификацию требует дополнительной алгоритмической проработки, которая является целью дальнейших научных исследований.

Список литературы

1. Макконнелл, С. Совершенный код / С. Макконнелл. – М. : Русская редакция, 2005.
2. Lozano, A. Assessing the effect of clones on changeability / A. Lozano, M. Wermelinger // Proceedings of the 24th IEEE International Conference on Software Maintenance. – 2008. – P. 227–236.
3. Comparative Stability of Cloned and Non-cloned Code: An Empirical Study / M. Mondal, C. K. Roy, S. Rahman, R. K. Saha, J. Krinke, K. A. Schneider // Proceedings of the 27th ACM Symposium on Applied Computing. – 2012. – P. 1227–1234.
4. Roy, C. K. Comparison and evaluation of code clone detection techniques and tools: A qualitative approach / C. K. Roy, J. R. Cordy, R. Koschke // Science of Computer Programming – SCP. – 2009. – Vol. 74, № 7. – P. 470–495.
5. Tsantalis, N. Refactoring clones: A new perspective / N. Tsantalis, G. P. Krishnan // Proceedings of the 7th International Workshop on Software Clones. – 2013. – P. 12–13.
6. Krishnan, G. P. Unification and Refactoring of Clones / G. P. Krishnan, N. Tsantalis // Proceedings of the European Conference on Software Maintenance and Reengineering. – 2014. – P. 104–113.
7. Фаулер, М. Рефакторинг: улучшение существующего кода / М. Фаулер. – СПб. : Символ-Плюс, 2003.

Н. Г. Федотов, А. А. Сёмов

Россия, Пенза, Пензенский государственный университет

К ВОПРОСУ ИЗВЛЕЧЕНИЯ ПАРАМЕТРОВ 3D ОБЪЕКТОВ ИЗ РЕЗУЛЬТАТОВ СКАНИРОВАНИЯ

Статья посвящена проблеме распознавания 3D изображений. Подробно рассматривается такой аспект, как извлечение параметров преобразования 3D модели из результатов сканирования (например, определение угла разворота 3D объекта). Данная проблема особенно актуальна в робототехнике и машинном зрении.

Вопрос, рассматриваемый в статье, примыкает к проблеме распознавания образов [1].

В робототехнике распознавание образов (в частности, 3D изображений) необходимо, например, при решении таких задач, как распознавание целей 3D объектов на местности, навигация на местности и в пространстве, захват 3D объекта с параллельным применением маневров уклонения (для ракетносителя). Промышленные и транспортные роботы при некачественной системе машинного зрения не смогут обеспечить надежную манипуляцию предметом при движении. Для этого роботу необходимо знать не только информацию о наличии препятствий, но и об углах поворота, путях обхода этих препятствий, поворотов и манипуляций самим предметом и т.п. [1, 2].

Таким образом, одной из важных задач развития компьютерного зрения является не только правильное распознавание 3D изображений, но и параллельное извлечение множества дополнительной информации о движении пространственных объектов, в частности, параметрах их ориентации.

Для решения поставленных задач авторами был разработан новый метод сканирования 3D объектов – 3D трейс преобразование, который позволяет извлекать различную информацию и параметры 3D модели из результатов сканирования без дополнительных преобразований.

Математическая модель 3D трейс преобразования

Обозначим через $B(\eta, r)$ плоскость на расстоянии r от начала координат и вектором нормалью $\eta = [\cos \varphi \cdot \sin \omega, \sin \varphi \cdot \sin \omega, \cos \omega]$, где ω – угол между $B(\eta, r)$ и Oxz , φ – угол между $B(\eta, r)$ и Oxy .

Исходный 3D объект F сканируется сеткой параллельных плоскостей (рис. 1). Результат пересечения каждой сканирующей плоскости $B(\eta(\omega, \varphi), r)$ с трехмерной моделью F характеризуется числом: $G = \text{HyperT}(F \cap B(\eta(\omega, \varphi), r))$. Объект сканируется под различными углами ω и φ с шагом $\Delta\omega$ и $\Delta\varphi$ до завершения оборота обхода объекта F в 2π радиан по каждому углу с дискретным шагом Δ между сканирующими плоскостями. В результате вычислений формируется гипертрейс матрица ЗТМ, у которой ось 0ω направлена горизонтально, ось 0φ – вертикально, ось $0r$ – вглубь.

Далее, используя строковые и столбцовые обработки матрицы ЗТМ при помощи гиперфункционалов HyperP , $\text{Hyper}\Omega$ и $\text{Hyper}\Theta$, вычисляется признак $\text{Res}(F)$ исходной модели [3]:

$$Res(F) = Hyper\Theta \circ Hyper\Omega \circ HyperP \circ HyperT(F_{sect})$$

Признаки получившихся в сечении фигур извлекаются при помощи 2D трейс преобразования, введенного Н. Г. Федотовым и развитого в последующих работах [1, 4–6]. Так, изображение фигуры сечения F_{sect} сканируется решеткой параллельных прямых $l(\rho, \theta)$ с дискретным шагом $\Delta\rho$ между линиями под различными углами θ , до завершения полного оборота обхода в 2π радиан. Признак 2D изображения имеет структуру в виде композиции трех функционалов:

$$\Pi(F_{sect}) = HyperT(F_{sect}) = \Theta \circ P \circ T(F_{sect} \cap l(\theta, \rho))$$

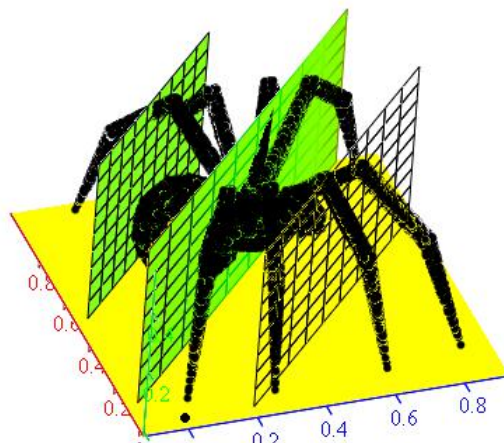


Рис. 1. Процесс сканирования 3D объекта плоскостями

3D трейс преобразование позволяет конструировать признаки как инвариантные, так и чувствительные к группе движений 3D объектов и к масштабированию [7, 8].

Извлечение дополнительной информации об 3D объекте

3D трейс преобразование является удобным инструментом исследования движущихся объектов и объектов, подвергающихся линейным искажениям. Оно является не только основой распознавания, инвариантного по отношению к этим преобразованиям, но и позволяет определить параметры этих преобразований [9]. 3D трейс преобразование является источником формирования конструктивных признаков распознавания нового класса – гипертриплетных признаков, характерной особенностью которых является их структура в виде композиции четырех функционалов.

3D трейс преобразование способно извлекать помимо множества основных типов признаков, в частности метрических и геометрических признаков, дополнительную информацию об объекте. При этом не требуется производить дополнительных измерений, т.к. извлечение информации происходит в той же технике сканирования.

Так, например, возможно получение информации об удаленности объекта от начала координат, его пространственной протяженности, выделения его главных осей, о масштабе объекта, кривизне поверхности, обобщенной информации об линейных искажениях объекта.

Опишем, например, как можно получить информацию о пространственной ориентации объекта в той же технике сканирования, что и при формировании признаков. Чтобы узнать, на сколько повернут исследуемый объект, нужно определить, относительно чего будет измеряться этот разворот. В качестве эталона возьмем

первоначальную неизмененную модель, хранимую в базе данных. Данный выбор оправдан, так как анализ эталона очень часто применяется при решении задачи идентификации или классификации.

Предлагаемый способ достаточно прост. Необходимо определить какие-либо опорные точки, опорные линии и/или опорные поверхности, которые жестко фиксированы по отношению к своим 3D объектам (пусть для определенности это будут объекты A и B). Зная угол между данными геометрическими фигурами можно узнать угол поворота между исходными объектами.

Для достижения однотипности техники сканирования в качестве таких опорных поверхностей возьмем сканирующую плоскость. Так, построив соответствующие плоскости для каждой из моделей A и B , получим геометрические объекты, одинаково закрепленные в своих моделях (т.е. преобразования, переводящие объект A в B , переведут и опорную плоскость для A в опорную плоскость для B). Угол между этими опорными плоскостями даст информацию об угле поворота, при котором один объект перейдет в другой.

Определить из всего множества сеток сканирующих параллельных плоскостей уникальные плоскости можно с помощью гиперфункционалов $HyperP$, $Hyper\Omega$ и $Hyper\Theta$, использующихся при формировании гипертриплетных признаков. Получающиеся в результате обработки совокупностей сеток плоскостей уникальные сечения могут сами выступать в качестве гипертриплетных признаков $Res(F)$ исходного 3D объекта. Например, признак сечения максимальной площади, признак сечения минимальной площади среди максимальных площадей срезов сеткой параллельных плоскостей под одним углом и т.п.

Для получения результатов более устойчивых к шумам и искажениям необходимо формировать несколько таких признаков и выбирать их усредненное значение [9, 10].

На языке гипертрейс матриц выбор уникальных сечений эквивалентен выбору конкретного элемента матрицы ЗТМ с номером (i, j, k) , которому соответствует тройка $(\omega_i, \varphi_j, r_k)$. Таким образом, зная сферические координаты плоскости можно определить углы поворота, переводящие одну плоскость в другую плоскость.

Работа выполнена при поддержке гранта РФФИ проект №12-07-00501.

Список литературы

1. Федотов, Н. Г. Теория признаков распознавания образов на основе стохастической геометрии и функционального анализа / Н. Г. Федотов. – М. : Физматлит, 2009. – 304 с.
2. Fedotov, N. G. The Theory of Image-Recognition Features Based on Stochastic Geometry / N. G. Fedotov // Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications. – 1998. – Vol. 8, № 2. – P. 264–266.
3. Fedotov, N. G. Trace transform of spatial images / N. G. Fedotov, S. V. Ryndina, A. A. Syemov // 11th International conference on Pattern Recognition and Image Analysis: New Information technologies (PRIA-11-2013) : Conference Proceedings (V. I–II). – Samara : IPSI RAS, 2013. – V. 1. – P. 186–188.
4. Fedotov, N. G. Recognition of halftone textures from the standpoint of stochastic geometry and functional analysis / N. G. Fedotov, D. A. Mokshanina // Pattern Recognition and Image Analysis, Advances in Mathematical Theory and Applications. – 2010. – Vol. 20, № 4. – P. 551–556.
5. Fedotov, N. G. Recognition of images with complex half-tone texture / N. G. Fedotov, D. A. Mokshanina // Measurement Techniques. – 2011. – Vol. 53, № 11. – P. 1226–1232.

6. Fedotov, N. Application of triple features theory to the analysis of half-tone images and colored textures. Feature construction along stochastic geometry and functional analysis / N. Fedotov, S. Romanov, D. Goldueva // Computer and Information Science / Canadian Center of Science and Education Canada. – 2013. – Vol. 6, № 4. – P. 17–24.

7. Федотов, Н. Г. Идеи построения равномерной сетки на сфере и 3d трейс- преобразование / Н. Г. Федотов, А. А. Сёмов // Проблемы информатики в образовании, управлении, экономике и технике : сб. ст. XIII Междунар. науч.-техн. конф. – Пенза : Изд-во АННОО «Приволжский Дом знаний». – 2013. – С. 23–26.

8. Федотов, Н. Г. Основные преимущества и дополнительные возможности 3D трейс-преобразования / Н. Г. Федотов, А. А. Сёмов // XXI век: итоги прошлого и проблемы настоящего Плюс. Серия: технические науки. Информационные технологии. – Пенза : Изд-во Пенз. гос. технол. ун-та, 2014. – № 03 (19). – С. 77–83.

9. Сёмов, А. А. Об одном подходе к распознаванию 3D-изображений / А. А. Сёмов // Надежность и качество : тр. Междунар. симп. / под ред. Н. К. Юркова. – Пенза : Изд-во ПГУ, 2013. – Т. 1. – С. 350–351.

10. Сёмов, А. А. Повышение надежности распознавания 3D-объектов на основе методов стохастической геометрии / А. А. Сёмов // Надежность и качество : тр. Междунар. симп. / под ред. Н. К. Юркова. – Пенза : Изд-во ПГУ. – 2014. – Т. 1. – С. 393–396.

Т. Т. Стас, А. Ю. Козлов

**Россия, Тверь, ЗАО «НПО Российские инновационные технологии»,
Пенза, Пензенский государственный университет**

ПРИМЕНЕНИЯ ФУНКЦИИ МНОГОМЕРНОГО НОРМАЛЬНОГО РАСПРЕДЕЛЕНИЯ В ПОСТРОЕНИИ АДАПТИВНОГО АЛГОРИТМА РАСПОЗНАВАНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ФРАЗЫ

Представлен математический алгоритм, полученный с помощью применения функции многомерного нормального распределения, для адаптации распознавателя к окружающей среде, не требуя при этом, использования большого количества обучаемых данных.

Введение

Имеются области применения систем распознавания речи, где, несмотря на растущую актуальность, существует проблема достижения высокой точности распознавания. К таким областям относится бортовая авиационная среда летательного аппарата при подаче экипажем отдельных речевых команд на вход авиационной электронной системы [1], справочные и поисковые системы, системы автоматической идентификации пользователей по голосу и т.д. Одним из наиболее эффективных методов повышения точности распознавания является метод адаптации распознавателя к условиям окружающей среды. Точность распознавания отдельных фраз, может понижаться в случае применения недостаточного количества обучаемых данных. Понижение точности распознавания может быть вызвано влиянием фактора наличия шумовых воздействий в окружающей среде, а также может быть вызвано такими факторами как варьирование от одного носителя к другому носителю одного и того же языка или варьирования одного и того же носителя языка. Для решения проблемы применения недостаточного количества обучаемых данных применяется метод адаптации к окружающей среде, который, в отличие от метода обучения с помощью сбора и подготовки новых речевых данных, не требует боль-

шого количества данных. Адаптация требует малых количеств обучаемых данных для приспособления существующих математических моделей распознавателя к новым характеристикам диктора или к новому шумовым воздействием окружающей среды. Наиболее известными методами адаптации являются: метод максимальной вероятностной линейной регрессии [1] и метод максимальной апостериорной вероятности [1]. В статье представлен математический вывод формулы, вычисляющий коэффициент, который можно использовать в составе математического алгоритма для обновления параметров функции многомерного нормального распределения при адаптации распознавателя к окружающей среде.

Применения функции многомерного нормального распределения в адаптивном алгоритме распознавания последовательности фразы

Для вычисления параметров ФМНР сначала необходимо получить некоторые параметры СММ распознавания отдельных фраз речи в процессе обучения распознавателя [3], для этого в первоначальной стадии обучения необходимо получить последовательность ВАП, затем вычислить первоначальные значения параметров СММ распознавания отдельных фраз речи. Далее с помощью применения алгоритма Баума – Велша можно получить параметры ФМНР [3].

Для получения последовательности ВАП применяется метод мел-частотных кепстральных коэффициентов [5]. Для получения первоначальных значений параметров СММ распознавания отдельных фраз речи применяется метод преобразования последовательности ВАП в последовательность дискретных символов, где каждый такой символ соответствует определенному вектору центра кластера, построенного с помощью отрезка речевого сигнала, входящего в состав состояния СММ отдельный фразы речи [3]. После определения первоначальных состояний СММ отдельный фразы речи с помощью применения алгоритма Баума – Велша вычисляются параметры ФМНР:

$$f_{x|S}(x(t) | s(t) = i) = \frac{1}{2\pi^{P/2} |\Sigma_i|^{1/2}} \exp \left\{ -[x(t) - \mu_i]^T \Sigma_i^{-1} [x(t) - \mu_i] \right\},$$

где $x(t)=[x(1), x(2), \dots, x(k), \dots, x(N)]^T$ – ВАП; N – общее количество признаков в ВАП; $f_{x|S}(x(t) | s(t) = i)$ – вероятность принадлежности ВАП $x(t)$ состоянию $s(t) = i$; $|\Sigma_i|$ – определитель матрицы Σ_i для состояния i ; Σ_i^{-1} – обратная матрица к матрицы Σ_i ; μ_i – вектор средних значений для состояния i ; Σ_i – квадратная матрица вектора средних значений.

Вектор средних значений вычисляется по следующей формуле [3]:

$$\overline{\mu}_i = \frac{\sum_{t=0}^{T-1} \gamma_t(i) x(t)}{\sum_{t=0}^{T-1} \gamma_t(i)},$$

где $\gamma_t(i)$ – апостериорная вероятность нахождения СММ с последовательностью акустических признаков $x(i)$ в состоянии i во время t , полученная в предыдущий стадии обучения, а квадратная матрица вектора средних значений вычисляется по следующий формуле [3]:

$$\overline{\Sigma}_i = \frac{\sum_{t=0}^{T-1} \gamma_t(i) (x(t) - \overline{\mu}_i)(x(t) - \overline{\mu}_i)^T}{\sum_{t=0}^{T-1} \gamma_t(i)}.$$

Для адаптации распознавателя к новом шумовым воздействием окружающей среды или к новым характеристикам диктора применяется алгоритм адаптации максимальной вероятности распознавания ФМНР. В соответствии с данным алгоритмом параметры ФМНР перенастраиваются и происходит повышение вероятности принадлежности ВАП состоянию СММ фразы.

Для адаптации распознавателя к окружающей среде предположим, что вектор средних значений для состояния i можно получить следующим образом:

$$\bar{\mu}_i = \mu_i \beta_i,$$

где $\mu_i = [\mu(1), \mu(2), \dots, \mu(k), \dots, \mu(N)]^T$; N – общее количество признаков в ВАП $x(t)$.

Логарифм функции ФМНР можно записать следующим образом:

$$P(\mu_i) = \log(f_{x|s}(x(t) | s(t) = i)) = C_i - [x_t - \mu_i \beta_i]^T \Sigma_i^{-1} [x_t - \mu_i \beta_i].$$

Поиск максимального значения целевой функции $P(\mu_i)$ осуществляется с помощью ее градиента, представляющего вектор $\nabla P(\mu_i)$, где справедливо следующее соотношение [2, 4]:

$$\nabla P(\mu_i) = \left(\frac{\partial P}{\partial \mu(1)}, \frac{\partial P}{\partial \mu(2)}, \dots, \frac{\partial P}{\partial \mu(K)}, \dots, \frac{\partial P}{\partial \mu(N)} \right).$$

Градиент целевой функции $P(\mu_i)$ можно записать следующим образом:

$$\nabla P(\mu_i) = \frac{\partial P}{\partial \mu(k)} = \frac{\partial (-[x_t - \mu_i \beta_i]^T \Sigma_i^{-1} [x_t - \mu_i \beta_i])}{\partial \mu(k)}, \quad (1)$$

где $k = 1, 2, \dots, N$.

При достижении целевой функции $P(\mu_i)$ максимального значения $-\frac{\partial P}{\partial \mu(k)} = 0$.

В таком случае, вычислив производную (1), можно получить следующее выражение:

$$\beta_i = (\mu_i^T \mu_i)^{-1} \mu_i^T x_t.$$

Заключение

В статье представлен математический вывод формулы, вычисляющий коэффициент, который можно использовать в составе математического алгоритма для обновления параметров ФМНР при адаптации распознавателя к окружающей среде, не требуя при этом использования большого количества обучаемых данных. Дальнейшие направление по обоснованию предложенного алгоритма заключаются в проведении сравнительных экспериментальных исследований, подтверждающих эффективность распознавания отдельных фраз речи с помощью адаптации распознавателя к окружающей среде, не требуя при этом использования большого количества обучаемых данных.

Список литературы

1. Englund, C. Speech recognition in the JAS 39 Gripen aircraft – Adaptation to speech at different G-loads /C. Englund // Master Thesis in Speech Technology / Department of Speech, Music and Hearing, Royal Institute of Technology. S-100 44 Stockholm. – 2004. – 11th March.

2. Тихомиров, В. А. Математическая модель системы распознавания акустической последовательности фраз речи / В. А. Тихомиров, Т. Т. Стас // Сборник Шестой всероссийской конференции, 26– 28 января 2011. – М. : МГТУ им. Баумана. – № 11. – С. 140–144.
3. Advanced digital signal processing and noise reduction fourth edition / Professor Saeed V. Vaseghi. – London, UK, 2008.
4. Рассел, С. Искусственный интеллект – современный подход / С. Рассел, П. Норвиг. – М. ; СПб. ; Киев, 2007.
5. Giampiero, Salvi. Mining Speech Sounds, Machine Learning Methods for Automatic Speech Recognition and Analysis : doctoral Thesis / Giampiero Salvi. – Stockholm, 2006.

Е. Г. Бердичевский, Д. В. Вознюк

Россия, Великий Новгород, Новгородский государственный университет имени Ярослава Мудрого

ПЕРСПЕКТИВЫ ИНФОРМАЦИОННО-ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ В ТЕХНИЧЕСКОЙ ЭСТЕТИКЕ

Рассмотрены особенности информационно-интеллектуальных систем для разработки аудио-визуальных композиций в мультимедийном рекламном дизайне. Предложен алгоритм гармонизации звуковых и визуальных составляющих рекламных видеороликов.

Информационно-интеллектуальные системы (ИИС) стали в последние годы объектом пристального внимания проектантов в области технической эстетики, к которой относятся архитектура, дизайн, реклама, промышленное искусство и др. Однако известные разработки и концепции ИИС нацелены в основном на визуальную составляющую продуктов технической эстетики. Известны разработки ряда структурных схем ИИС, принципов формирования баз знаний и баз шаблонов, правила формирования экспертных групп и многое другое [1]. Значительно меньше внимание уделено ИИС для продуктов технической эстетики, содержащих не только визуальную, но и ауди (звуковую) компоненту. Характерным представителем такой продукции является мультимедийная реклама.

Учитывая все возрастающие объемы разработок рекламных мультимедийных роликов и все возрастающую значимость рекламы в современной экономике, представляется актуальным рассмотреть принципы создания ИИС, содержащих дополнительно блоки (или алгоритмы), позволяющие автоматически гармонизировать в процессе проектирования формообразование, движение (анимацию) и музыкально-звуковое сопровождение. Сравнительно молодая научная дисциплина «киматика», занимающаяся изучением влияния внешних звуковых волн на формообразование материальных объектов, позволила выявить ряд физических объектов и качественных закономерностей. В частности, швейцарский ученый Хане Йенни, предложивший термин «киматика» в 1920 г., доказал, что в определенных ситуациях звук может творить форму. В соответствии с теорией доктора Йенни между звуковыми гармониками и материальными гармоническими структурами существует взаимосвязь [2]. Различные субстанции обретали форму под воздействием звуковых гармоник с различной частотой колебаний. Еще дальше пошел Ганс Кайзер, разрабо-

тавший теорию мировых гармоник. Кайзер полагал, что исследование принципов, лежащих в основе взаимосвязи между музыкой и математикой, позволяет вывести законы взаимосвязи между тонами и числами. Теория Кайзера предполагала, что принцип соотношений целых чисел лежит не только в учении о гармониках, но и в множестве других проявлений живой и неживой природы. В частности, Кайзер разработал теорию взаимодействия законов гармонии и архитектуры. По мнению ученого далеко не все формы, встречающиеся в геометрии и природе, подчинены закону гармонических соотношений, но именно формы, соотносящиеся с гармоническими рядами, являются наиболее эстетичными.

Ближе всех к решению проблемы «интеллектуализации» процесса создания электронных аудио-визуальных концепций подошла Демидова М. В. [3], предложившая четыре основных принципа проектирования:

- Принцип целостности формообразования объекта, определяющий структуру издания и общий характер взаимодействия основных частей концепции.
- Принцип системной информативности, который предусматривает проектное воплощение графического интерфейса, задающего стиль мультимедийному изданию. Это реализуется через согласование звуковой и визуальной системы подачи контента. Принцип отражает тип кодирования информации, динамический прием осуществления интерактивности элементов, а также динамику смены информации на экране. При этом согласуются отношения транслируемого количества информации по времени трансляции.
- Принцип формализованного содержания. Согласно принципа проектирование ведется от внутренней структуры к внешней оболочке.
- Принцип динамического развития, который основан на особенности динамики предъявления и восприятия информации. Возможность динамического предъявления информации влияет на композицию каждого отдельного экрана и определяет особенности композиционного формообразования мультимедийного издания в целом.

Вышеизложенные принципы явились теоретической базой для разработки алгоритма подбора и гармонизации визуальных и звуковых составляющих композиции рекламного видеоролика (рис. 1).

Исследование значительного количества видеороликов позволило выявить пять основных составляющих, определяющих в значительной степени эстетические и функциональные параметры изделия. К таким составляющим отнесены стиль/жанр; ритм; динамика/статика; взаимодействие главного и второстепенного (фона и доминанты) и эмоции.

Исходя из тезиса, что звуковое сопровождение рекламного видеоролика практически не влияет на его информативность, акцент был сделан на составляющих, отвечающих за эмоциональную насыщенность рекламы. В алгоритм не включены колористические характеристики визуальной составляющей композиции (цвет, свет), так как, с одной стороны, вопросы светомузыкальной акустики достаточно хорошо разработаны [5], с другой стороны, использование цветовых характеристик значительно усложнило бы применимость алгоритма

Из вышеуказанных основных составляющих особый интерес представляет тембр, который можно определить как колористическую окраску звука. Именно тембр позволяет различить звуки одной высоты и громкости, и именно он вызывает наибольший эмоциональный отклик у слушателя, формирует в воображении образы, вызывает ассоциации.

Тембр до сих пор остается самой неизученной и неклассифицированной составляющей звука. Поэтому ИИС должна содержать базу характеристик тембра,

позволяющую дизайнеру искать соответствие тембра визуальной среде. Тембральную специфику звука при наличии базы характеристик можно сравнить с органолептическими ощущениями от тех или предметов или явлений визуальной среды. К специфическим базам данных, включаемых в ИИС, относятся базы музыкальных композиций разных эпох и стилей, базы ритмических отношений, базы визуальных фонов (паттернов) и др.

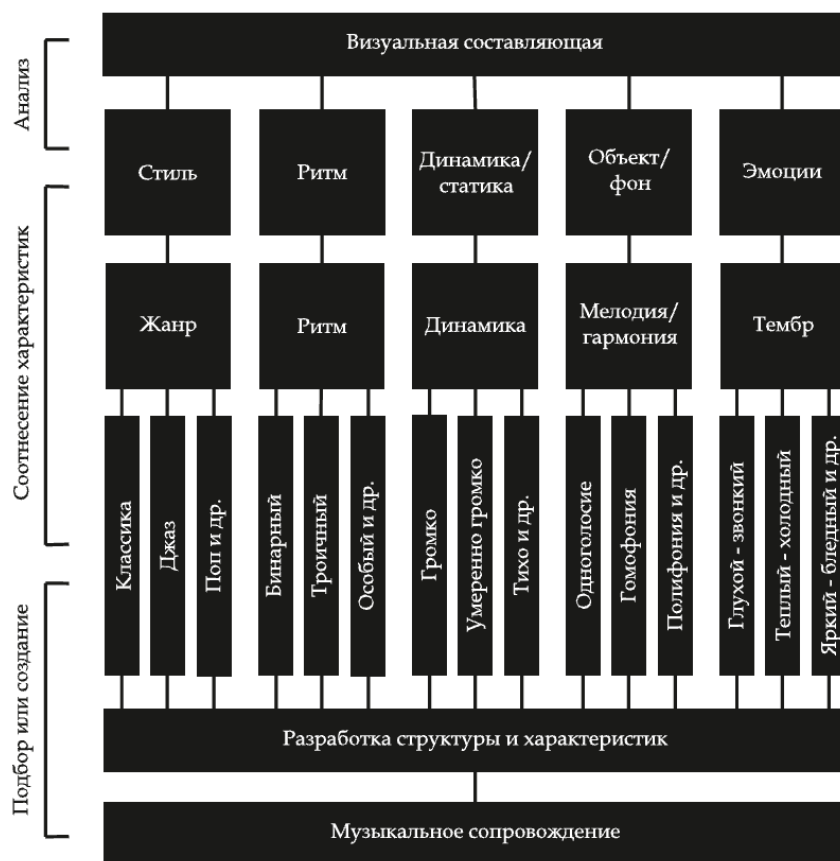


Рис. 1. Стилизованный алгоритм подбора музыкального сопровождения к рекламному видеоролику

При разработке баз основных показателей звуко-визуальной композиции возникли терминологические трудности. Нужно было в качестве основы для терминов принять формулировки, понятные дизайнерам и художникам. Эти формулировки отличаются от узкопрофильных акустических и физических определений.

Для апробации предложенного алгоритма был произведен редизайн двух популярных рекламных видеороликов. На основании анализа пяти важнейших показателей визуальной составляющей были сформулированы оптимальные значения показателей музыкального (звукового) сопровождения. Из соответствующей базы были отобраны музыкальные композиции, основные показатели которых близки к установленным. На видеоряд рекламного ролика были наложены новые музыкальные композиции, скорректированные с основными показателями визуального изображения. После синхронизации и гармонизации элементов композиции получился новый рекламный продукт. Экспертная оценка показала, что после наложения нового музыкального сопровождения на прежний видеоряд эффективность рекламного и эмоционально-эстетического воздействия роликов на потребителя существенно возросла.

Предлагаемый алгоритм подбора музыкального (звукового) сопровождения видеоряда позволит автоматизировать основные процедуры и может служить основой для разработки информационно-интеллектуальных систем в рекламном дизайне. Алгоритм вполне адаптирован для использования в системах виртуальной компьютерной реальности.

Список литературы

1. Бердичевский, Е. Г. Информационно-коммуникационные технологии как инструмент принятия решений в дизайне / Е. Г. Бердичевский // Информация и связь. – 2012. – Вып. 5. – С. 33–36.
2. URL: <http://cymatica.ru>
3. Демидова, М. В. Проектирование визуально-звуковой композиции в дизайне мультимедиа : автореф. дис. ... / Демидова М. В. – СПб., 2006.
4. Шлыкова, О. В. Культура мультимедиа / О. В. Шлыкова. – М. : ФАИР ПР ЕСЕ, 2004.
5. Шлыкова, О. В. Светомузыка в системе искусств / О. В. Шлыкова. – Казань : КГК, 1991.

К. А. Орешкин

Россия, Пенза, Пензенский государственный университет

ИСПОЛЬЗОВАНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ БАЗЫ ДАННЫХ INTERSYSTEMS CACHE В КАЧЕСТВЕ ПЛАТФОРМЫ ДЛЯ РЕАЛИЗАЦИИ АКТИВНОЙ БАЗЫ ДАННЫХ

Описаны преимущества использования объектно-ориентированной базы данных InterSystems Cache в качестве платформы для реализации активной базы данных, описаны средства языка Cache Object Script для реализации основных структурных моментов.

Традиционно базы данных (БД) рассматривались как репозитории, которые хранят необходимую для приложений информацию и доступны прикладным программам или пользователям через интерактивные интерфейсы. Системы управления базами данных (СУБД) пассивны в том смысле, что действия, выполняемые СУБД, инициируются пользователями или программными приложениями. В таком контексте, целый ряд различных инструментов и систем используются вместе, чтобы поддержать требования приложения. Тем не менее, СУБД начинают применяться к целому ряду областей, связанных с очень сложной обработкой информации все более значительных объемов данных, а также высокими требованиями к производительности, в котором традиционная многокомпонентная среда уже оказывается неудовлетворительной. Это привело к тенденции в области исследований баз данных, которые содержат в себе функционал, необходимый приложению, внутри самой СУБД. Активные базы данных (АБД) – это базы данных, позволяющие следующее:

1. Содержать логику обработки в самой базе данных так, чтобы она управлялась на уровне системы управления активными базами данных (СУАБД), а не прикладным программным обеспечением приложений.

2. Обеспечивать некоторую форму мониторинга событий и условий, которые воздействуют на данные и могут инициировать обработку, управляемую базой данных.

3. Включать в систему базы данных некоторое средство, с помощью которого эти события и условия могли бы запускать логику внутри базы данных.

Объектно-ориентированные базы данных (ООБД), в отличие от реляционных, изначально поддерживают близкую связь данных и поведения. Это выражается в объединении данных и методов в одну сущность. Пользовательские методы позволяют определить требуемую реакцию ООБД на внешние события (например, для изменения состояния объекта пользователь вызывает метод `setValue`, который помимо установки нового значения поля объекта может выполнить дополнительные действия).

В ООБД схема данных строится в соответствии с объектно-ориентированной концепцией. База данных представляется пользователю в виде набора классов, каждый из которых обладает индивидуальностью (набором атрибутов) и поведением (набором методов). Работа с базой данных ведется не в терминах запросов, как в реляционных БД, а в терминах работы с объектами (чтение атрибутов объекта и вызов его методов). В связи с этим слой активности ООБД должен поддерживать обнаружение другого набора событий, таких как извлечение объекта из БД, вызов метода объекта, сохранение объекта из БД и т.п.

InterSystems Cache обеспечивает полную поддержку ОО-методов проектирования и разработки, включая инкапсуляцию, множественное наследование и полиморфизм, а также реализацию встраиваемых объектов, ссылок, коллекций и отношений. В ООБД InterSystems Cache существует два варианта передачи ссылок на произвольный объект. Для идентификации объекта в оперативной памяти используется ссылка OREF (object reference), если же объект сохраняется в базе данных, ему назначается долговременный объектный идентификатор – OID (object ID). Объект получает OID в момент первой записи в базу данных. OID не меняется, пока существует объект. OREF – назначается объекту, когда он попадает в оперативную память. При каждой новой загрузке в оперативную память объект может получить новую объектную ссылку OREF.

АБД в состоянии контролировать и реагировать на конкретные обстоятельства, имеющие отношение к приложению. АБД должна обеспечить модель знаний (т.е. есть описание механизма) и модель исполнения (т.е. стратегии выполнения) для поддержки данного реактивного поведения.

Общий подход для модели знаний использует правила. Наиболее популярной формой активных правил являются так называемые ЕСА-правила (ЕСА-rules, «Event-Condition-Action», правила «Событие Условие Действие»). Каждое такое правило описывает действие (action), которой должно быть выполнено при возникновении одного или более событий (events), при условии выполнения некоторого требования или группы требований (conditions). Когда нужное событие происходит, говорят, что правило срабатывает, после срабатывания правило рассматривается на предмет выполнения необходимых требований, и если эти требования выполняются, то правило исполняет предписанные ему действия. Наиболее подходящим средством реализации правила является комбинация триггеров и хранимых процедур.

Функции обратного вызова запускаются СУБД при выполнении определенных действий с объектом. Они являются аналогией механизма триггеров в реляционных СУБД. ООБД Cache поддерживает следующие функции обратного вызова:

1. `%OnNew` – вызывается при создании нового объекта в оперативной памяти в тот момент, когда выделена память под объект и инициализированы его свойства.

2. %OnOpen – функция вызывается перед извлечением существующего объекта из базы данных.

3. %OnDelete – данная функция вызывается перед удалением объекта.

4. %OnBeforeSave и %OnAfterSave – вызываются непосредственно перед и сразу после сохранения объекта в базу данных соответственно.

Для обнаружения возникновения ошибочных ситуаций в Cache используется механизм обработчика ошибок (Error handler). Cache позволяет при возникновении ошибки передать управление пользовательской функции-обработчику. Для этого устанавливается значение системной переменной \$ETRAP.

```
SET $ETRAP="GOTO LogError^ErrRoutine"
```

Переменная \$ETRAP содержит код на Cache Object Script, который выполняется при возникновении ошибки. В приведенном выше примере управление передается функции LogError (которая является частью функции ErrRoutine). Функции-обработчику доступна следующая информация:

1. Тип контекста (DO, EXCUTE или функция пользователя).
2. Номер строки в исходном тексте, где произошла ошибка.
3. Код ошибки.

Таким образом, обнаружение событий типа Exception сводится к установке обработчика, который будет информировать слой активности о всех возникающих ошибках.

Из событий управления транзакциями Cache поддерживает обнаружение только событие отката транзакции. Обнаружить событие можно с помощью функции обратного вызова %OnRollBack. Функция вызывается перед началом отката транзакции. Откат транзакции может быть остановлен (для этого необходимо вернуть код ошибки, отличный от \$\$\$OK). Начало транзакции может быть косвенно определено как вызов метода %Save объекта (вызов %Save автоматически начинает транзакцию). Однако этот метод не является надежным, т.к. Cache позволяет сохранить несколько объектов в одной транзакции с помощью специальных конструкций Cache Object Script (TSTART, TCOMMIT, TROLLBACK).

Данные особенности ООБД Cache показывают, что в системе реализован большой функционал по определению событий, которые является одним из основных компонентов активной базы данных. Данный функционал значительно шире, чем в традиционных реляционных базах данных, что является следствием объектно-ориентированной структурой данных. Таким образом, Cache находится ближе к понятию активных баз и предоставляет значительно шире набор инструментов по реализации ЕСА-модели для пользователя.

ОСНОВНЫЕ ПОЛОЖЕНИЯ ПО ФОРМАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ДИНАМИЧЕСКИХ ПРОЦЕССОВ НА ОСНОВЕ СОБЫТИЙНО-ИЕРАРХИЧЕСКОГО ФРЕЙМОВОГО МЕТОДА ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

В настоящей статье приводятся основные положения по формализации действий информационных динамических процессов описываемой предметной области посредством фреймов-действий и событийно-иерархических фреймов.

Одним из возможных способов формализации элементарных действий в ходе проектирования программного обеспечения является способ, в ходе которого описание предметных областей осуществляется посредством фреймов-действий.

Типовая структура фрейма-действия изображена на рис. 1 и представляет собой граф, где X_{cd} – субъект действия; X_d – действие; X_{od} – объект действия; $X_{kd1}, X_{kd2}, \dots, X_{kdn}$ – компоненты действия; C_{cd} – класс субъекта действия; C_{od} – класс объекта действия; $C_{kd1}, C_{kd2}, \dots, C_{kdn}$ – классы компонентов действия. Субъект действия, объект действия и все компоненты действия связаны с самим действием X_d бинарными структурными отношениями R_{str} :

- $r_{cd}(X_{cd}, X_d)$ – отношение между субъектом действия и действием;
- $r_{od}(X_d, X_{od})$ – отношение между действием и объектом действия;
- $r_{kd1}(X_d, X_{kd1}), r_{kd2}(X_d, X_{kd2}), \dots, r_{kdn}(X_d, X_{kdn})$ – отношение между действием и компонентами действия.

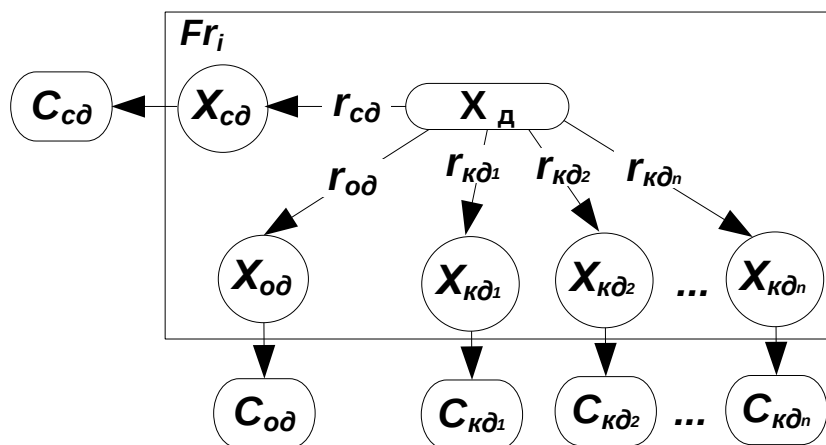


Рис. 1. Типовая структура фрейма-действия

Структуру фрейма-действия можно представить следующими частями:

- событийная (субъект – действие – объект);
- вспомогательная (субъект – действие – (компонента 1, компонента 2, ..., компонента n)).

Событийная и вспомогательная части фрейма-действия образуют единую структуру.

Представить событийную часть фрейма-действия можно составным высказыванием S, а вспомогательную часть – составным высказыванием H.

Высказывания образуются посредством конъюнкций бинарных предикатов.

$$S = r_{c\partial}(X_{\partial}, X_{c\partial}) \& r_{o\partial}(X_{\partial}, X_{o\partial}),$$

$$H = r_{k\partial 1}(X_{\partial}, X_{k\partial 1}) \& r_{k\partial 2}(X_{\partial}, X_{k\partial 2}) \& \dots \& r_{k\partial n}(X_{\partial}, X_{k\partial n})$$

Таким образом, итоговый фрейм-действие можно представить конъюнкциями составных высказываний S и H.

$$Fr = S \& H$$

где $S = r_{c\partial}(X_{\partial}, X_{c\partial}) \cup r_{o\partial}(X_{\partial}, X_{o\partial})$,

$$H = r_{k\partial 1}(X_{\partial}, X_{k\partial 1}) \cup r_{k\partial 2}(X_{\partial}, X_{k\partial 2}) \cup \dots \cup r_{k\partial n}(X_{\partial}, X_{k\partial n})$$

Итоговый фрейм-действие также можно представить в виде:

$$Fr = X_{\partial}((r_{c\partial}, X_{c\partial}), (r_{o\partial}, X_{o\partial}), (r_{k\partial 1}, X_{k\partial 1}), (r_{k\partial 2}, X_{k\partial 2}), \dots (r_{k\partial n}, X_{k\partial n})).$$

Учитывая функциональный характер отношений $r_{c\partial}, r_{o\partial}, r_{k\partial 1} \dots r_{k\partial n}$ фрейм-действие Fr можно преобразовать посредством представления конъюнкций следующих предикатных равенств:

$$Fr^f = [f_{c\partial}(X_{\partial}) = X_{c\partial}] \& [f_{o\partial}(X_{\partial}) = X_{o\partial}] \& [f_{k\partial 1}(X_{\partial}) = X_{k\partial 1}] \& \dots \& [f_{k\partial n}(X_{\partial}) = X_{k\partial n}]$$

Граф преобразованного фрейма-действия изображен на рис. 2.

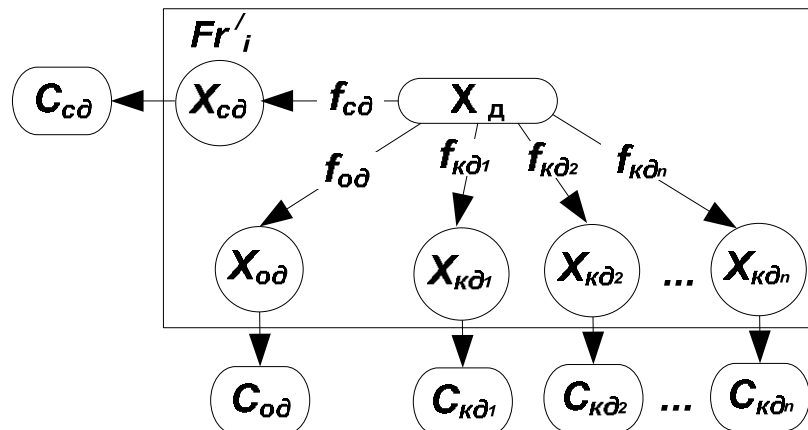


Рис. 2. Преобразованный фрейм-действие

В целях обеспечения целостности описания модель информационных динамических процессов описываемой предметной области, кроме знаний о действиях должна отражать и знания о событиях, возникающих в результате совершения описываемых действий.

В зависимости от различных условий протекания информационного динамического процесса формируются определенные состояния объектов, связанные с наступлением событий. Законы развития событий образующие информационные динамические процессы имеют иерархический характер. События вызывают выполнение функций, которые, в свою очередь, изменяют состояния объектов и формируют новые события, и т.д., пока не будет завершен некоторый процесс.

Для формализации свойства иерархичности была предложена модель событийно-иерархического фрейма, который представляет собой совокупность иерархически вложенных фреймов-действий и (или) событийно-иерархических фреймов

низшего порядка, связанных между собой алгебраическими операциями, определяющими последовательность, порядок возникновения и следования событий в ходе динамических процессов.

Свойства фреймов-действий в совокупности с событийно-иерархическими фреймами могут являться одним из способов формализации информационных динамических процессов моделей предметных областей при проектировании программного обеспечения.

Итоговая модель информационных динамических процессов должна представлять собой объединение всех фреймов-действий ($\mathcal{A}$) и событийно-иерархических фреймов ($\mathcal{B}$) в единую модель информационно динамических процессов описываемой предметной области ($\mathcal{C}$):

$$\mathcal{C} = \mathcal{A} \cup \mathcal{B}$$

Таким образом, структуризация событийно-иерархических фреймов с введением операционной семантики в структуру фреймов позволяет расширить функциональные возможности существующих средств моделирования информационных динамических процессов в части формализации условий возникновения событий в ходе последовательных и параллельных информационных динамических процессов.

И. А. Лезин, Д. Е. Маркелов

Россия, Самара, Самарский государственный аэрокосмический университет

ИЗВЛЕЧЕНИЕ ЗНАНИЙ ИЗ БОЛЬШИХ МАССИВОВ ДАННЫХ

В условиях резкого повышения объема информации является целесообразным разработка методов и инструментов по извлечению знаний из больших массивов данных. Главное отличие знаний от статических данных, хранящихся в базах данных, состоит в их структурности и активности. Появление в базе знаний новых фактов или установление новых связей может стать источником изменений в принятии решений.

Введение

Каждый момент работы гидроэлектростанции (ГЭС) фиксируется огромное количество физических параметров (около 10 тыс.), характеризующих режим работы гидроагрегатов, электрических сетей, трансформаторов, насосных станций и т.д. Дискретность многих из них составляет 1 с. Такие измерения приводят к накоплению массивов числовых характеристик значительной размерности, что затрудняет хранение, анализ и интерпретацию полученных результатов.

Если исходить из предположения о том, что данная выборка отображает какие-либо функциональные зависимости между параметрами, пусть даже неизвестные, то в таком случае можно перейти к созданию набора логических правил, описывающих эти зависимости, т.е. к базе знаний.

Полученные правила позволят внести интеллектуальную составляющую в процесс управления составом гидроагрегатов в соответствии с плановыми значениями нагрузки и резерва с целью обеспечения наибольшей экономичности работы ГЭС.

Важно, что создаваемая автоматизированная система (АС) является прежде всего рекомендательной, предоставляющей информацию о более рациональном состоянии агрегатов, она не принимает непосредственного участия в управлении и потому никак не влияет на устойчивость системы. Работой ГЭС управляет системная автоматика.

Входные данные

Входными данными системы является массивы параметров, накопленные в центре сбора и обработки данных (ЦСОД), имеющемся на ГЭС и располагающий архивными данными за несколько лет. Новые поступающие данные также обрабатываются системой. В этом случае база знаний будет расширена и в дальнейшем в режиме проактивной работы система пересмотрит и оптимизирует существующую базу знаний. Обновленная база знаний будет доступна для следующих сеансов определения состава гидроагрегатов.

Например, план по заданной мощности определен всего на 2 ч вперед, и каждое новое задание приводит к новому анализу текущего состояния базы знаний.

Также к входной информации относятся следующие характеристики:

- 1) эксплуатационные характеристики гидроагрегатов (зависимость КПД от активной мощности гидроагрегата и напора воды);
- 2) расходные характеристики (зависимость расхода воды через гидроагрегатов от активной мощности гидроагрегата и напора воды);
- 3) различные зоны запрещенной и нерекомендованной работы гидроагрегатов;
- 4) остаточный моторесурс (время до капитального ремонта, время работы в граничных условиях, предельное количество пусков и остановов);
- 5) пределы по верхнему и нижнему уровням водохранилища.

Зависимости КПД и расхода воды от мощности агрегатов при постоянном напоре представляются в АС сплайнами третьего порядка. Для промежуточных значений напоров величины КПД вычисляются линейной интерполяцией с использованием данных при ближайших значениях напоров. Такие операции необходимы, чтобы получить достоверные и непрерывные значения КПД на всех диапазонах значений мощностей и напоров.

Выходные данные

Результатом работы являются рекомендация по составу работающих и резервированных гидроагрегатов и также их активной мощности.

Алгоритм работы

После загрузки массивов данных система начинает выполнять этап статистической обработки, заключающийся в нахождении оценки взаимосвязи переменных. Целью данного этапа работы алгоритма является исключение из рассмотрения переменных, которые не вносят существенный вклад в значение результирующей переменной.

Коэффициент корреляции рангов, предложенный К. Спирменом, относится к непараметрическим показателям связи между переменными, измеренными в ранго-

вой шкале. При расчете этого коэффициента не требуется никаких предположений о характере распределений признаков в генеральной совокупности. Этот коэффициент определяет степень тесноты связи порядковых признаков, которые в этом случае представляют собой ранги сравниваемых величин.

При использовании коэффициента ранговой корреляции условно оценивается теснота связи между признаками, считая значения коэффициента равные 0,3 и менее, показателями слабой тесноты связи; значения более 0,4, но менее 0,7 показателями умеренной тесноты связи, а значения 0,7 и более показателями высокой тесноты связи. Таким образом, переменные с низким показателем можно исключить из рассмотрения.

После сокращения объема обрабатываемых данных за счет исключения незначущих переменных система переходит к этапу кластеризации. На данном этапе определяются группы параметров. Результатом кластеризации будет являться то, что близкие по своему влиянию параметры будут принадлежать одной группе.

Следующим шагом в построении базы правил является определение принадлежности значения параметра к тому или иному кластеру.

Для этого каждому примеру из сокращенной выборки ставится в соответствие отдельное правило. Для этого для каждого правила определяются степени принадлежности заданных значений переменных к соответствующим кластерам. После чего каждому обучающему примеру ставятся в соответствие те кластеры, степени принадлежности к которым у соответствующих значений переменных из этого примера являются максимальными.

На этапе определения коэффициента принадлежности было принято допущение, что границы значений функции принадлежности лежат на срединных значениях соседних с рассматриваемым кластером. Точная подстройка вида функции принадлежности осуществляется изменением параметра кривой функции. Вид функции, а также нечеткая алгебра могут быть заданы вручную пользователем АС. После определения принадлежности система уже обладает первичной базой знаний. До применения базы знаний должна быть проведена ее оптимизация.

Этап оптимизация рассматривает правила во взаимодействии между собой. Рассматриваются такие качества базы знаний как непротиворечивость и полнота. Исключение противоречивых правил проводится на основе подсчета рейтинга правил. Достоинством предложенного решения является то, что поиск групп противоречивых правил осуществляется с помощью синтаксического анализа, что позволяет избавиться от рассмотрения конкретных значений, связей между ними и привязки к конкретной предметной области. Определение полноты базы знаний заключается в определении некоего «предела» совпадения различных характеристик истинности. Для этого определяется супремум всех выводов возможных посылок и инфимум всех возможных заключений по каждому правилу.

Результирующая база может быть представлена пользователю для просмотра как набор правил вида «ЕСЛИ...ТО».

Результаты работы АС

Результатом работы автоматизированной системы построения и оптимизации баз знаний является набор правил, отражающий закономерности работы гидроагрегатов.

Рекомендации системы должны способствовать обеспечению суммарного минимального расхода воды при необходимой суммарной выработке электроэнергии, т.е. позволит повысить общее КПД ГЭС. Таким образом, разработка данного ПО

является экономически целесообразной и отвечает требованиям государственной программы Российской Федерации повышения энергоэффективности до 2020 г.

Список литературы

1. Извлечение знаний из баз данных при помощи нейронной сети и нечеткого интерпретатора : сб. науч. тр. / А. Г. Мурашко, И. В. Шевченко, В. А. Самоляк, С. Ю. Лабзов // Нові технології : науковий вісник. – 2008. – № 4 (22). – С. 41–44. – URL: http://n-tech.kueitu.com/view.html_689
2. Заде, Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л. А. Заде ; пер с англ. под ред. А. Н. Аверкина. – М. : Горячая линия – Телеком : ФИЗМАТЛИТ, 2009 – 252 с.
3. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилинский, Л. Рутковский. – М. : Горячая линия – Телеком, 2006. – 452 с.
4. Борисов, В. В. Нечеткие модели и сети / В. В. Борисов, В. В. Круглов, А. С. Федурлов. – М. : Горячая линия – Телеком, 2007. – 484 с.
5. Мандель, И. Д. Кластерный анализ / И. Д. Мандель. – М. : Финансы и статистика, 2010. – 176 с.
6. Новак, В. Математические принципы нечеткой логики / В. Новак, И. Перфильева, И. Мочкорж ; пер с англ. под ред. А. Н. Аверкина. – М. : Горячая линия – Телеком : ФИЗМАТЛИТ, 2006. – 252 с.

Е. С. Мангалова

Россия, Красноярск, Сибирский государственный аэрокосмический университет им. М. Ф. Решетнева

ПОСТРОЕНИЕ АНСАМБЛЯ НЕПАРАМЕТРИЧЕСКИХ ОЦЕНОК С ИЗМЕНЯЮЩИМИСЯ ПАРАМЕТРАМИ РАЗМЫТОСТИ ДЛЯ РЕШЕНИЯ ЗАДАЧ ВОССТАНОВЛЕНИЯ РЕГРЕССИИ И РАСПОЗНАВАНИЯ ОБРАЗОВ

В работе предложен алгоритм построения ансамбля непараметрических оценок последовательного обучения. В процессе построения ансамбля определяются различные наилучшие параметры размытости для различных подобластей исходного признакового пространства.

На сегодняшний день существует широкий спектр алгоритмов восстановления регрессии и распознавания образов: от методов статистического анализа до современных алгоритмов машинного обучения. Это многообразие алгоритмов не случайно и свидетельствует об отсутствии универсального метода для решения всех задач. Более того, при решении практических задач классификации и восстановления регрессии может оказаться, что ни один из алгоритмов не имеет необходимой точности. В таких случаях бывает целесообразно строить ансамбли моделей.

Понятие последовательного обучения ансамблей было введено Р. Е. Шапиром [1]. Основная идея последовательного обучения заключается в построении цепочки моделей, при этом каждая следующая обучается на ошибках текущего ансамбля.

Пусть имеется множество наблюдений $G = \{g_1, g_2, \dots, g_n\}$. Каждое наблюдение характеризуется набором переменных $g_i = \{x_i^1, x_i^2, \dots, x_i^m, y_i\}$, где x^j – независимые переменные, значения которых известны и на основании которых определяется значение зависимой переменной y . Значения переменной y имеет категориальный тип в случае задачи распознавания образов, непрерывный тип – в случае задачи восстановления регрессии [2]. Далее будем полагать, что если мы имеем дело с задачей классификации, то это задача бинарной классификации. Так как в случае наличия K классов, задача может быть сведена к K задачам бинарной классификации.

Распространенным методом решения данной задачи является построение непараметрической оценки следующего вида[3]:

$$h(\bar{x}) = \frac{\sum_{i=1}^n \prod_{j=1}^m K\left(\frac{x^j - x_i^j}{c^j}\right) y_i}{\sum_{i=1}^n \prod_{j=1}^m K\left(\frac{x^j - x_i^j}{c^j}\right)},$$

где K – ядерная функция; $\bar{c}$ – вектор параметров размытости.

Одним из возможных недостатков данного подхода является то, что параметры размытости одинаковы при различных значениях входной переменной, хотя характер зависимости для разных подобластей пространства признаков может быть различным. На рис. 1 приведен пример такой зависимости. Заметим, что непараметрическая оценка с большим параметром размытости хорошо ведет себя на сильно зашумленном участке и сглаживает истинную зависимость в подобласти, где она сильно меняется, в то время как непараметрическая оценка с малым параметром размытости переобучается на зашумленном участке.

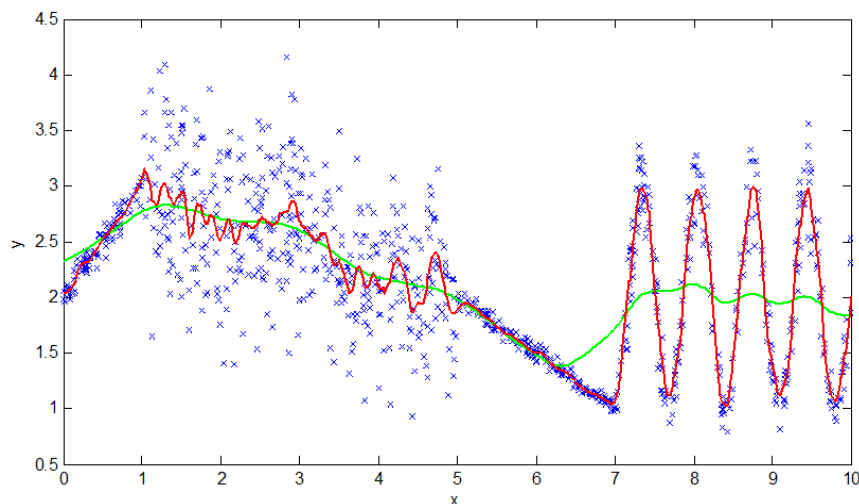


Рис. 1. Тестовый пример. Оценка с большим параметром размытости изображена зеленым, с малым параметром размытости – красным

Предложим алгоритм, позволяющий с помощью построения ансамбля определять подобласти, для которых целесообразно выбирать различные параметры размытости.

Строим первичный ансамбль, состоящий из одной модели:

$$H_1(\bar{x}) = h_1(\bar{x}),$$

$$h_1(\bar{x}) = \frac{\sum_{i=1}^n \prod_{j=1}^m K\left(\frac{x^j - x_i^j}{c_1^j}\right) y_i}{\sum_{i=1}^n \prod_{j=1}^m K\left(\frac{x^j - x_i^j}{c_1^j}\right)},$$

параметры размытости на первом шаге выбираются заведомо большими.

Затем строим модель невязки ансамбля и добавляем эту модель в ансамбль:

$$h_2(\bar{x}) = \frac{\sum_{i=1}^n \prod_{j=1}^m K\left(\frac{x^j - x_i^j}{c_2^j}\right) (y_i - H_1(\bar{x}_i))}{\sum_{i=1}^n \prod_{j=1}^m K\left(\frac{x^j - x_i^j}{c_2^j}\right)},$$

$$H_2(\bar{x}) = H_1(\bar{x}) + h_2(\bar{x}).$$

Подобный процесс продолжается пока добавление новой модели уменьшает ошибку $\sum_{i=1}^v (y'_i - H_k(\bar{x}'_i))^2$, вычисленную по проверочной выборке $\{\bar{x}'_i, y'_i\}$, $i = 1, 2, \dots, v$:

$$h_k(\bar{x}) = \frac{\sum_{i=1}^n \prod_{j=1}^m K\left(\frac{x^j - x_i^j}{c_k^j}\right) (y_i - H_{k-1}(\bar{x}_i))}{\sum_{i=1}^n \prod_{j=1}^m K\left(\frac{x^j - x_i^j}{c_k^j}\right)},$$

$$H_{k+1}(\bar{x}) = H_k(\bar{x}) + h_{k+1}(\bar{x}).$$

Параметры размытости уменьшаются на каждом шаге:

$$\bar{c}_k = \left\{ \bar{c}_k^r = A \bar{c}_{k-1}^r : r = \arg \min (E_k^r); \bar{c}_k^b = \bar{c}_{k-1}^b : b \neq r \right\}$$

где A – некоторый коэффициент (меньше 1); E_k^r – ошибка, вычисленная по проверочной выборке при уменьшении r -го параметра размытости в $1/A$ раз и фиксированных остальных параметров.

В результате получаем последовательность ансамблей $H_k(\bar{x})$, $k = 1, 2, \dots$ (рис. 2).

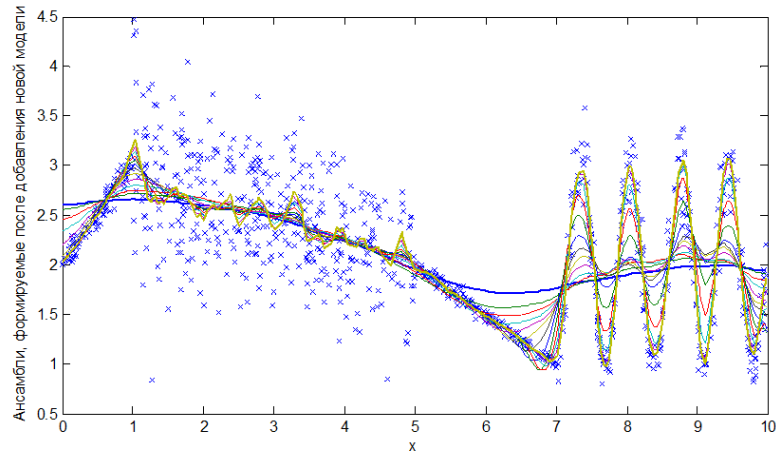


Рис. 2. Последовательность ансамблей

Далее вычисляем разнообразие ансамблей. Одной из мер разнообразия является дисперсия выходов ансамблей (рис. 3). Когда добавление новой модели $h_k(\bar{x})$ в ансамбль приводит к уменьшению разнообразия ансамблей, это значит, что для некоторого $\bar{x}$ выходы ансамблей $H_k(\bar{x})$ и $H_{k+1}(\bar{x})$ близки, а следовательно, рост количества моделей в ансамбле в окрестности $\bar{x}$ стоит прервать. Наибольший такой участок для рассматриваемого примера – это отрезок (1.5, 5), для которого рост ансамбля прекращается после 3 шага.

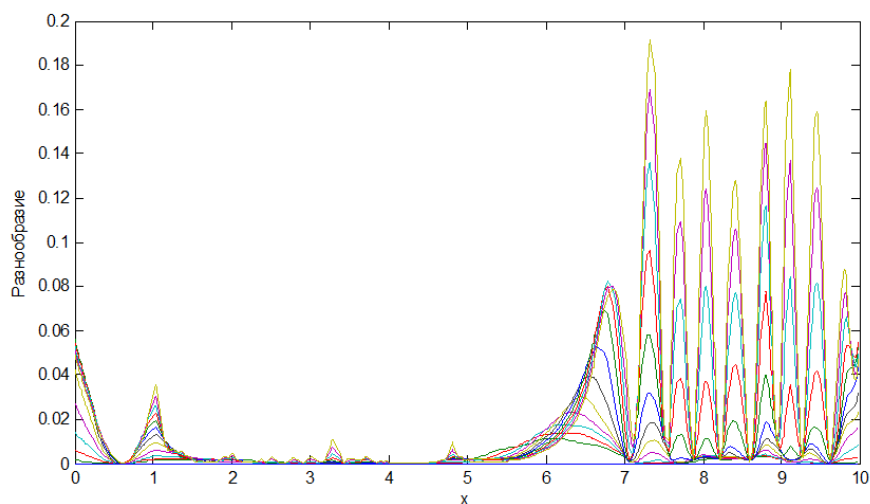


Рис. 3. Разнообразие ансамблей

В результате получаем ансамбль, который для разных областей пространства содержит разное количество моделей (рис. 4). Полученный ансамбль хорошо сглаживает сильно зашумленный участок и отслеживает зависимость на сильно изменяющемся отрезке.

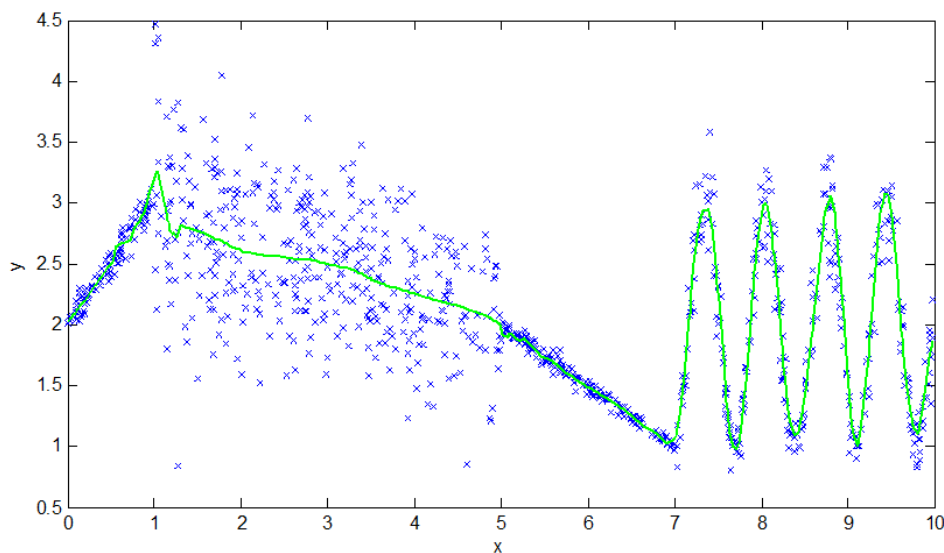


Рис. 4. Итоговый ансамбль

В работе был предложен алгоритм построения ансамблевой системы непараметрических оценок последовательного обучения, позволяющий в процессе построения ансамбля определять различные наилучшие параметры размытости для различных подобластей исходного признакового пространства.

Список литературы

1. Schapire, R. E. The strength of weak learnability / R. E. Schapire // Machine Learning. – 1990. – Vol. 5, no. 2. – P. 197–227.
2. Барсегян, А. А. Анализ данных и процессов / А. А. Барсегян, М. С. Куприянов, И. И. Холод. – BHV, 2009. – 512 с.
3. Хардле, В. Прикладная непараметрическая регрессия / В. Хардле. – М.: Мир, 1993. – 349 с.

М. В. Бобырь, С. А. Кулабухов

Россия, Курск, Юго-Западный государственный университет

ПРОГНОЗИРОВАНИЕ РЕЗУЛЬТАТА ЧИСЛЕННЫХ ВЫЧИСЛЕНИЙ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ

В статье рассмотрены способы прогнозирования результата вычислений на основании регрессии и нейро-нечеткой системы. Оценка вычислений проводилась на основании показателя средне-квадратического отклонения (RMSE). Анализ показал предпочтительность использования нейро-нечетких систем для прогнозирования результатов численного вычисления.

На сегодняшний день в приборо- и машиностроении для прогнозирования результатов вычислений традиционно используются методы регрессионного анализа. Эти методы применимы, когда на объекты управления действуют различные внешние возмущающие факторы, но не в случае если исследуемая область не может быть описана линейными функциями. Тогда для прогнозирования выходного результата рекомендовано использовать методы мягких вычислений, основанные на теории нейро- и нейро-нечетких адаптивных сетей.

Традиционные методы прогнозирования основываются на синтезе уравнения регрессии вида:

$$y_{reg} = y + a_1 * X_1 + a_2 * X_2 + a_3 * X_3 + a_4 * X_4, \quad (1)$$

где a_1, a_2, a_3, a_4 – коэффициенты регрессии; X_1, X_2, X_3, X_4 – входные значения, y – выходное значение.

Однако их использование не всегда позволяет получить максимально приближенный результат к эталонному значению.

С целью повышения точности прогнозирования в условиях неопределенности, авторами предложено использовать нейронечеткую систему на основе трехслойной нейронной сети. Перед использованием нейронной сети ее необходимо обучить по алгоритму, представленному на рис. 1.

Для оценки оптимальности методов прогнозирования предложено проводить на основе минимизации коэффициента среднеквадратического отклонения (RMSE) по формуле:

$$RMSE = \sqrt{\frac{1}{N} * \sum_{i=1}^N (y_{reg_i} - y)^2}, \quad (2)$$

где N – количество рассматриваемых вариантов входа и выхода.

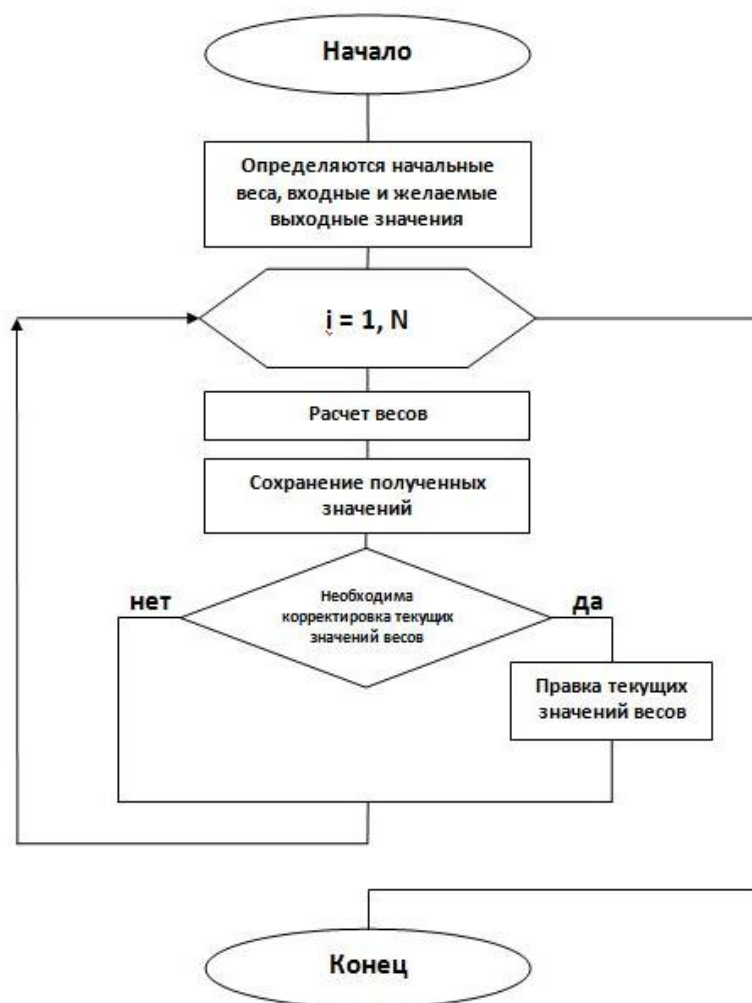


Рис. 1. Алгоритм обучения нейронечеткой системы

Анализ моделирования показал, что использование нейронечеткой системы дает больше точные значения при прогнозировании по отношению к методам регрессии (табл. 1).

Таблица 1

Выходные значения уравнения регрессии и нейронечеткой системы

№	Входные значения				Эталонный выход	Выход нейро- нечеткой системы	Выход метода регрессии
	x_1	x_2	x_3	x_4			
					y	y_{fuzzy_net}	y_{reg}
1	2	2	4	4	2	1,88	2,06
2	2	2	3	4	2	1,88	1,79
3	1	2	1	3	1	1,22	0,64
4	4	3	2	2	4	4,13	3,69
5	4	4	4	3	5	4,83	5,47
6	3	2	3	3	3	2,93	2,60
7	3	5	4	4	5	4,49	4,99
8	1	2	2	1	1	1,15	1,31
9	3	2	2	4	2	2,22	2,24
10	2	2	2	3	1	1,39	1,62

По результатам прогнозирования с помощью регрессии и нейронечеткой системы получены значения RMSE равные 0,69 и 0,42 соответственно, что позволяет сделать вывод, что использование нейронечетких систем в условиях неопределенности, позволяет получать максимально точные результаты вычислений, а следовательно подтверждает предпочтительность использования мягких вычислений.

Список литературы

1. Титов, В. С. Нечетко-логическая система управления стабилизацией процесса резания деталей / В. С. Титов, М. В. Бобырь, С. А. Кулабухов // Известия Юго-Западного государственного университета. – 2013. – № 9. – С. 54–58.
2. Титов, В. С. Учет инерционности в нечетко-логических системах управления / В. С. Титов, М. В. Бобырь, С. А. Кулабухов // Промышленные АСУ и контроллеры. – 2014. – № 10.
3. Титов, В. С. Автоматическая компенсация тепловых деформаций шпиндельных узлов прецизионного оборудования с ЧПУ / В. С. Титов, М. В. Бобырь, Н. А. Милостная // Промышленные АСУ и контроллеры. – 2006. – № 11. – С. 31–35.
4. Бобырь, М. В. Моделирование нечетко-логических систем управления на основе мягких арифметических операций / М. В. Бобырь, В. С. Титов, Г. Ю. Акульшин // Вестник компьютерных и информационных технологий. – 2013. – № 3. – С. 29–35.
5. Титов, В. С. Оценка влияния числа обучаемых точек на аддитивность нечетких систем / В. С. Титов, М. В. Бобырь, С. А. Кулабухов // Промышленные АСУ и контроллеры. – 2014. – № 10. – С. 30–35.
6. Емельянов, С. Г. Адаптивные нечетко-логические системы управления / С. Г. Емельянов, В. С. Титов, М. В. Бобырь. – М. : АРГАМАК-МЕДИА, 2013. – 184 с.
7. Бобырь, М. В. Интеллектуальная система управления подачей при обработке деталей резанием / М. В. Бобырь, В. С. Титов, Н. А. Беломестная // Промышленные АСУ и Контроллеры. – 2010. – № 8. – С. 17–20.
8. Бобырь, М. В. Интеллектуальная система управления температурными деформациями при резании / М. В. Бобырь, В. С. Титов // Автоматизация и современные технологии. – 2011. – № 5. – С. 3–7.

Э. В. Михайлов, Р. П. Богуш

Республика Беларусь, Полоцк, Полоцкий государственный университет

ОБРАБОТКА ИЗОБРАЖЕНИЙ ПЛАТЕЖНЫХ КВИТАНЦИЙ ПРИ АВТОМАТИЗИРОВАННОМ ВВОДЕ ДАННЫХ С ПРИМЕНЕНИЕМ МОБИЛЬНЫХ УСТРОЙСТВ

Рассмотрен подход к обработке изображений платежных квитанций, полученных с помощью фотокамер мобильных устройств. Представлены основные шаги алгоритма и результаты экспериментов.

Современные специализированные решения оптического распознавания текстов (OCR), разработанные ABBYY или Nuance позволяют организовать комплексный оптический ввод финансовых документов с помощью специальных сканеров или повседневных средств получения цифровых изображений (бытовой сканер, фо-

тоаппарат и т.д.). Однако, несмотря на то, что практически все финансовые институты стараются избавиться от бумажного документооборота, внедряя системы штрих кодирования и дистанционного обслуживания, для соблюдения легитимности при работе с клиентами все же необходимо отражать часть информации на бумажных носителях (чеках, счетах и т.д.). Поэтому, для оптического ввода финансовой информации самим клиентом, актуальным является развитие систем OCR на базе мобильных устройств, оснащенных фотокамерами. Изображения, полученные фотокамерами мобильных устройств, имеют ряд особенностей: высокий уровень шума, невысокая резкость, неточность цветопередачи и др. [1].

На этапе получения изображения, фотографирование квитанции целиком, как правило, приводит к получению изображения не пригодного для автоматизированной обработки. Поэтому целесообразно формирование изображения квитанции из его частей. На данном этапе также выполняется проверка качества изображения с использованием методов предварительной оценки качества [2].

Как правило, съемка изображений мобильным телефоном выполняется при недостаточном освещении. В этом случае, для достижения приемлемой яркости изображения необходимо увеличивать светочувствительность матрицы, что приводит к появлению яркостного и хроматического шума на изображении. Отсутствие достаточной освещенности бумажного документа и специфические для мобильного устройства условия съемки (дрожание фотокамеры и т.д.) делают изображение размытым и неконтрастным. Кроме этого, для сохранения изображения подвергается сжатию в специальные форматы с потерей качества, например JPEG, что зачастую приводит к появлению артефактов. Исходный фрагмент платежной квитанции, полученной с помощью мобильного телефона, представлен на рис. 1.



Рис. 1. Фрагмент изображения квитанции, полученный с помощью встроенной фотокамеры мобильного телефона

Таким образом, необходима предварительная обработка изображения квитанции, которая направлена на улучшения качества изображения и включает фильтрацию изображения от шумов, повышение резкости и контрастности изображения.

Для подавления неравномерного освещения в условиях мобильной съемки, а так же нормализации локальных уровней освещенности используется алгоритм SSR (SingleScaleRetinex) [3]:

$$R_i(x, y) = \log I_i(x, y) - \log [F_i(x, y) * I_i(x, y)],$$

где $R_i(x, y)$ – выходное изображение; $I_i(x, y)$ – исходное изображение; $F_i(x, y)$ – функция Гаусса.

Для устранения геометрических и перспективных искажений применяется метод, основанный на использовании патча Безье [4], который с учетом реализации на мобильных устройствах обеспечивает высокую скорость работы и простоту реализации.

В качестве алгоритма адаптивной бинаризации следует применять модифицированный, робастный к шумам, алгоритм Кристиана. Пример применения алгоритма к изображению, показанному на рис. 1, приведен на рис. 2. Порог бинаризации T для скользящего окна вычисляется как [5]:

$$T = (1 - \alpha_1) * \mu + \alpha_2 * \left(\frac{\sigma}{R_\sigma} * [\mu - M] + \alpha * M \right),$$

где $\alpha_1 = k_1 * \left(\frac{\sigma}{R_\sigma} \right)^\gamma$, $\alpha_2 = k_2 * \left(\frac{\sigma}{R_\sigma} \right)^\gamma$, k_1 и k_2 – положительные константы, при $\gamma = 2$ k_1 и k_2 рекомендуется брать в диапазонах 0,1–0,2; 0,15–0,25 и 0,01–0,05; σ – среднее квадратическое отклонение яркости в заданном окне; R_σ – максимальное значение среднее квадратического отклонения в заданном окне; μ – математическое ожидание значения яркости в заданном окне; M – минимальное значение яркости в заданном окне.

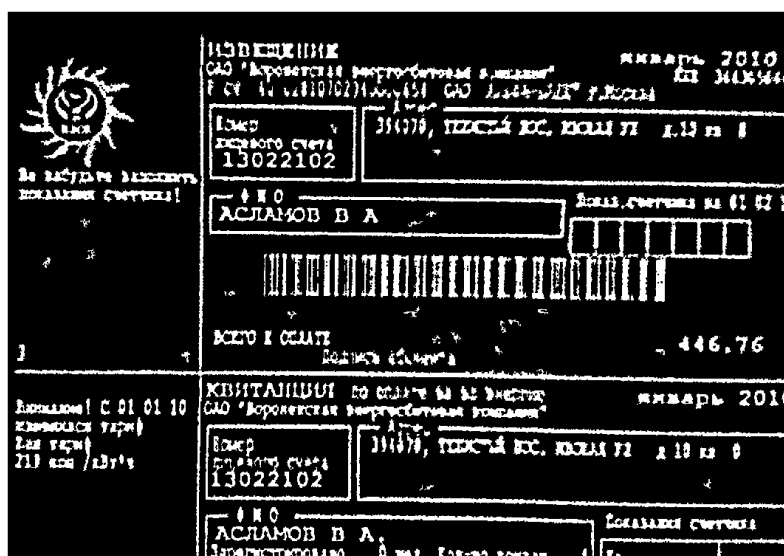


Рис. 2. Пример адаптивной бинаризации фрагмента изображения квитанции по алгоритму Кристиана

Далее выполняется автоматическое склеивание фрагментов изображений из уже обработанных частей путем сопоставления SIFT (Scale Invariant Feature Transform) дескрипторов. SIFT является одним из самых эффективных алгоритмов поиска характерных точек на двух или более изображениях, инвариантный к мас-

штабу изображений и изменениям яркости, а также достаточно устойчив к изменениям ракурса изображения [6]. В результате работы алгоритма получается множество особенных точек на каждой паре изображений с набором своих дескрипторов. Эти точки сравниваются попарно на соседних изображениях, и если дескрипторы двух выбранных точек совпадают в пределах заданной точности, то две точки принимаются как сопряженные. Можно считать, что эти точки были получены как проекция одной и той же трехмерной точки на плоскости двух камер. Пример работы алгоритма представлен на рис. 3.

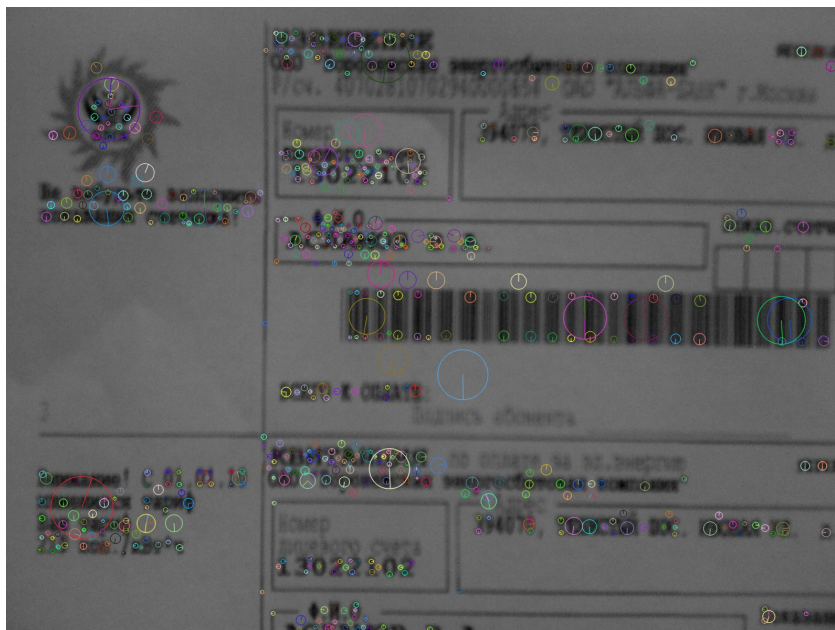


Рис. 3. Фрагмент изображения квитанции, с найденными на ней характерными точками по алгоритму SIFT

Выделение и поиск маркеров происходит по найденному множеству SIFT дескрипторов и по бинарному изображению документа. Основные типы маркеров разделены на три основные группы относительно их значимости для идентификации документа:

1) к самым значимым признакам можно отнести: печати, логотипы, штрих коды, стилизованные надписи;

2) ко второй группе относят: линии, пересечения (углы), текстовые блоки и отдельные надписи;

3) наименее значимыми признаками являются отдельные символы

Проведенные эксперименты показали, что представленный алгоритм обработки изображения квитанции позволяет представить документ в виде упорядоченного множества хорошо различимых признаков, а промежуточный результат работы алгоритма может быть использован системами оптического распознавания текста без предварительной обработки.

Список литературы

1. Улучшение качества фото с мобильного телефона. – 2014. – URL: <http://www.photoshop-master.ru/lessons.php?rub=3&id=587> (дата обращения: 11.08.2014).
2. ABBYY Mobile Imaging SDK. – 2014. – URL: <http://www.abbyy.ru/mobile-imaging-sdk/description> (дата обращения: 23.08.2014).

3. Bian, Z. Retinex Image Enhancement Techniques / Z. Bian, Y. Zhang // Final Project Report for EE264
4. Масалович, А. Использование патча Безье для аппроксимации искажения изображений текстовых документов / А. Масалович, Л. Местецкий // Proc. Graphicon–2007. – 2007. – P. 171–174.
5. Адаптивная бинаризация. – 2014. – URL: <http://www.heclub.ru/2013/12/03-yy.html> (дата обращения: 01.09.2014).
6. Lowe, David, G. Object recognition from local scale-invariant features / David G. Lowe // Proceedings of the International Conference on Computer. – 1999. – Vol. 2. – P. 1150–1157.

И. С. Вершинин

Россия, Казань, Казанский национальный исследовательский технический университет им. А. Н. Туполева

ПОМЕХОУСТОЙЧИВОСТЬ АНАЛИЗА АССОЦИАТИВНО-ЗАЩИЩЕННЫХ НЕСЖАТЫХ БИНАРНЫХ СЦЕН

Анализируется влияние случайных и детерминированных помех на правильность анализа и стойкость ассоциативной защиты бинарных сцен как таковых, без их представления в виде таблиц-отношений, атрибутами которых являются коды объектов и их координат.

Рассматривается случай представления ассоциативно-защищенных бинарных сцен как таковых, т.е. без сжатия (в терминах «объекты-координаты»).

Задача анализа стилизованных бинарных сцен формулируется следующим образом. Пусть γ – число типов объектов анализируемой сцены, $t \in \{1, \gamma\}$ – номер типа объекта. По условию состояние любого узла (бита) сетки с координатами (i, j) – 1 либо 0. Каждый объект занимает прямоугольную область размерами $m \times n$ бит. Его идентификация происходит, например, по правой нижней границе матричного представления объекта.

Для данного троичного эталона (троичной матрицы)

$$X^t = ||x_{pq}^t||; x_{pq}^t \in \{0, 1, -\}, p = 1 \dots m^t, q = 1 \dots n^t;$$

необходимо найти координаты всех покрываемых им объектов двоичного изображения (булевой матрицы)

$$A = ||a_{ij}||; a_{ij} \in \{0, 1\}, i = 1 \dots k, j = 1 \dots \ell; k \geq m, \ell \geq n.$$

Запись $x_{pq}^t = (-)$ означает безразличное (замаскированное) значение соответствующего элемента эталона. Оно может быть как нулем, так и единицей. Множество замаскированных элементов отображается единичными элементами матрицы масок

$$M^t = ||\mu_{pq}^t||; \mu_{pq}^t = \begin{cases} 0, & x_{pq}^t \in \{0, 1\}; \\ 1, & x_{pq}^t \in \{-\}. \end{cases}$$

Постановка задачи. Далее рассматриваются стилизованные бинарные изображения – картографические сцены, ассоциативно-защищенные применением специального АЛГОРИТМа маскирования [1].

В качестве исходной информации, подлежащей хранению или передаче, используется бинарная сцена, содержащая множество объектов размером 11×18 байт. Объекты разбиваются на фрагменты, представленные бинарными матрицами 11×6 в алфавите почтовых символов. По условию, объекты не пересекаются, поэтому для обозначения пробелов между ними вводится пустой символ тех же размеров.

Для всех одиннадцати фрагментов (включая пустой) по АЛГОРИТМу случайно генерируется набор масок. Объекты и пустые символы погружаются по маскам в т.н. стегоконтейнеры, первоначально заполненные отрезками практически случайных последовательностей двоичных бит. При этом значения незамаскированных бит исходных матриц сохраняются. Тем самым осуществляется необходимая рандомизация. Известный набор масок является ключом распознавания.

Задачей является сравнительная оценка критичности анализа ассоциативно-замаскированных картографических сцен к действию помехи анализ стойкости.

Для повышения помехоустойчивости предлагается генерировать для защиты сцены несколько наборов масок k_M , дизъюнктивно объединяемых при погружении в стегоконтейнеры описания сцены (избыточность). Процедура распознавания принимаемого стегосообщения проводится по всем наборам масок. Для каждого стегосимвола за результат распознавания берется эталон, число распознаваний которого $k_r \geq \lfloor (k_M + 1)/2 \rfloor$. Если это условие не выполняется, то фиксируется факт искажения символа (отказ от распознавания).

Экспериментальные исследования. Исходное изображение состоит из 100 существенных объектов (10 строк по 10 объектов, промежутки между соседними объектами заполнены пробелами, строки существенных объектов разделяются строками пустых символов). Пример исходного изображения представлен на рис. 1.

579	471	001	937	038	431	744	477	423	694
931	351	228	369	958	823	936	963	202	296
256	816	297	933	610	151	086	509	119	874
043	438	029	706	371	814	820	122	570	724
584	628	891	944	378	019	675	509	175	370
090	937	353	630	478	531	660	222	142	334
887	109	899	618	032	415	030	842	776	000
210	740	561	069	900	708	119	002	661	129
836	565	891	471	905	122	794	874	002	770
072	469	565	502	157	364	422	361	400	195

Рис. 1. Пример исходного изображения

1 случай. Для каждой области сцены размером 198 байт (всего таких областей для рассматриваемой сцены – 247) случайные помехи локализованы в пределах 16 байт.

Эксперимент проводился следующим образом.

1. Генерируется исходная сцена, погружаемая по маске в случайную последовательность.

2. Случайным образом выбирается 16 байтов для каждой области сцены, в пределах которых будет действовать помеха.

3. В каждом из 16 вышеуказанных байтов всех областей сцены случайным образом выбирается один искажаемый (инвертируемый) бит.

4. Проводится распознавание искаженной сцены.

5. Определяется количество неверных распознаваний.

6. Пункты 3–5 повторяются 8 раз (до полной инверсии всех битов в шестнадцати выбранных байтах).

Обобщенные результаты проведенного эксперимента на множестве 10^3 сцен для существенных объектов (не пробелов) свидетельствуют о том, что метод *без введения избыточности* эффективен лишь при малом количестве помех.

Однако при анализе искаженной сцены возможна ситуация, когда в результате неверного распознавания пустых символов будет получен некий существенный объект, отсутствующий в исходной сцене. Тогда, даже без внедрения избыточности, полученные результаты позволят сделать вывод об искажении сцены.

Далее эксперимент был продолжен с внедрением избыточности при $k_M = 7$. Последовательность проведения нового эксперимента такова:

1. Генерируется исходная сцена, погружаемая по совокупности масок в случайную последовательность.

2. Случайным образом выбирается 16 байтов для каждой области сцены, в пределах которых будет действовать помеха.

3. В каждом из 16 вышеуказанных байтов всех областей сцены случайным образом выбирается один искажаемый (инвертируемый) бит.

4. Проводится распознавание искаженного множества стегоконтейнеров по всем наборам масок.

5. Определяется количество неверных распознаваний и отказов от распознавания.

6. Пункты 3-5 повторяются 8 раз (до полной инверсии всех битов в шестнадцати выбранных байтах).

Результаты экспериментов для существенных объектов говорят о том, что введение избыточности позволяет свести к нулю количество неверных распознаваний.

Отметим, что при внедрении избыточности в рассматриваемом случае фактов неверного распознавания *пустых* символов (пробелов) не выявлено.

2 случай. Для каждой области сцены размером 198 байт случайные помехи искажают более 16 байт.

Были проведены эксперименты, аналогичных рассмотренным в предыдущем случае, с той разницей, что проводилось последовательное искажение *всех* байтов каждой области при $k_M = 1$ и 7. Из полученных результатов при $k_M = 1$ следует, что в отличие от первого случая рассматриваемый метод неэффективен в целом даже при малом количестве помех в искажаемых байтах. Однако нередки случаи, когда в результате неверного распознавания пустых символов будет получен некий существенный объект, отсутствующий в исходной сцене, что позволит установить факт искажения.

Внедрение избыточности эффективно только при инверсии не более 1–2 битов в искажаемых байтах. В остальных случаях получаем отказ от распознавания (для существенных объектов). Однако в рассматриваемом случае при $k_M = 7$ для пустых символов возможны единичные факты их неверного распознавания. Поэтому целе-

сообразен переход к количеству масок $k_M = 9$. Проведенные аналогично описанным выше экспериментальные исследования при $k_M = 9$ установили факт отсутствия неверного распознавания пустых символов.

Оценка стойкости. В работе [2] постулируется безусловная стойкость метода ассоциативной стегозащиты при отсутствии ассоциаций с картой местности в случае сжатого представления сцены в терминах «объекты-координаты». Однако в случае представления сцены как таковой достижение безусловной стойкости невозможно. На рис. 2 представлен результат распознавания защищенной сцены на неверном ключе.

Наличие в результатах распознавания объектов, состоящих более чем из трех фрагментов, позволяет сделать заключение о *неверности* примененного для распознавания ключа. Таким образом, для сцен без сжатия достижима только вычислительная стойкость, связанная с невозможностью полного перебора ключей за приемлемое время.

Заключение. Как и в случае со сжатыми изображениями, возможности предлагаемого подхода по коррекции ошибок эффективны только в случае искажения малого числа байтов в стегоконтейнере. В противном случае, с учетом вводимой избыточности по количеству масок, можно говорить лишь о детектировании (т.е. обнаружении) ошибки, но не о ее исправлении. Тем не менее, предлагаемый подход может быть полезен в чрезвычайно зашумленных каналах связи, когда количество ошибок *превышает* возможности используемого в системе связи кода коррекции и исправления ошибок.

```
33043 50 3 353054 313235132730303595133
535323552239193829534333163599901338349
0334932505536695939195226585599012 2306
335359723921920395283331963593335583331
553392 53015550931532298338525363392 46
57428335334419973253919903693 242117430
535033213243351037136248351905489429 05
823930439321782426233490133839369533155
391 20231925366382693632333864349105345
165325319328333332 8310198232524353368
233988 71823688953994359203323933852668
324 7552239341282532 3016539 182633 329
3443636319584333532303099349369552446 4
03223869176393636 62303655373234353333
2329535390 9349722152993517559313422926
616322322051341850813279923843303312484
396285635313153880997932873789 93225534
2557943 6145964439392542684826523195365
2832433053 1325129023593529002338739442
```

Рис. 2. Пример распознавания на неверном ключе

Дополнительно при отсутствии избыточности, в отличие от сжатых изображений, имеется возможность установления факта искажения сцены, если количество фрагментов в полученном в результате распознавания объекте превышает допустимую.

Обсуждение. Полученные результаты могут быть распространены и на случай произвольных объектов, состоящих из различного числа фрагментов, отличных от почтовых символов. Внедрение избыточности позволит выявить факт искажения сцены. Этому же (даже без внедрения избыточности) способствует и получение цепочки фрагментов, либо не совпадающей ни с одним из существенных объектов, либо с разрядностью, превышающей допустимую.

Список литературы

1. Райхлин, В. А. К решению задачи маскирования стилизованных двоичных изображений / В. А. Райхлин, И. С. Вершинин, Е. Е. Глебов // Вестник КГТУ им. А. Н. Туполева. – 2001. – № 1. – С. 42–47.
2. Вершинин, И. С. Изменение результатов распознавания на множестве замаскированных бинарных матриц при действии аддитивных помех / И. С. Вершинин, Р. Ф. Гибадуллин // Вестник КГТУ им. А. Н. Туполева. – 2012. – № 4–1. – С. 198–206.

В. И. Горбаченко, М. В. Жуков

Россия, Пенза, Пензенский государственный университет

ИСПОЛЬЗОВАНИЕ СЕТЕЙ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ ДЛЯ ИДЕНТИФИКАЦИИ СВОЙСТВ СРЕДЫ ПО РЕЗУЛЬТАТАМ ДИСКРЕТНЫХ ИЗМЕРЕНИЙ¹

Предложен бессточный подход к решению задачи идентификации свойств среды по результатам дискретных измерений. Подход основан на использовании метода параметрической идентификации и сетей радиальных базисных функций. Регуляризация решения осуществляется с помощью итерационного метода, в котором роль регуляризатора выполняет количество итераций, определяемое по невязке. Эффективность подхода подтверждена на примере коэффициентной задачи для эллиптического уравнения.

Задача идентификации свойств среды по результатам дискретных измерений является коэффициентной обратной задачей. Рассмотрим следующую коэффициентную обратную задачу, заданную в операторной форме:

$$L(k(x, u))u(x) = f(x), x \in \Omega, \quad (1)$$

$$Bu(x) = p(x), x \in \partial\Omega, \quad (2)$$

где u – искомое решение; k – неизвестный коэффициент; L – дифференциальный оператор, зависящий от коэффициента k ; оператор B задает граничные условия; Ω – область решения; $\partial\Omega$ – граница области; f и p – известные функции и дополнительного условия:

$$Du(z) = \varphi(z), z \in Z, \quad (3)$$

где D – оператор, задающий дополнительное условие; $Z \subset \Omega \cup \partial\Omega$; φ – известная функция. В прикладных исследованиях типичной является ситуация, когда дополнительные условия заданы с погрешностью, поэтому вместо условия (3) будем использовать условие

$$Du(z) \approx \varphi^\delta(z), z \in Z. \quad (4)$$

¹ Работа поддержана грантом РФФИ 14-01-00660 «Методы построения нейросетевых и гибридных математических моделей процессов и явлений в сложных технических системах».

Кроме того, будем считать, что точность задания функции φ^δ определяется величиной δ .

При приближенном решении коэффициентных обратных задач особого внимания заслуживает метод параметрической идентификации [1]. Его идея заключается в том, что неизвестный коэффициент представляется в параметрическом виде, параметры которого необходимо найти. В качестве такого представления можно использовать сеть радиальных базисных функций (РБФ-сеть) [2]:

$$k_{RBF}(\mathbf{x}, u) = \sum_{m=1}^M w_m G(\mathbf{x}, u; \mathbf{p}_m)$$

где G – радиальные базисные функции (РБ-функции) с параметрами $\mathbf{p}_m$; w_m – веса РБ-функций; M – количество РБ-функций. Неизвестные параметры $\mathbf{p}_m$ и веса РБ-функций w_m выбираются из условия минимизации невязки между левой и правой частями (4), которое можно записать в виде:

$$J(\mathbf{w}^k, \mathbf{p}^k) = \int_Z \left(Du(\mathbf{z}; \mathbf{w}^k, \mathbf{p}^k) - \varphi^\delta(\mathbf{z}) \right)^2 d\mathbf{z} \rightarrow \min \quad (5)$$

где $\mathbf{w}^k = \{w_1, w_2, \dots, w_M\}$, $\mathbf{p}^k = \{p_1, p_2, \dots, p_M\}$.

Задача (5) имеет две особенности: во-первых, поскольку функция φ^δ задана с погрешностью, то существует множество решений задачи (5), которые с точностью δ удовлетворяют (1), (2), (4); во-вторых, функция u в (5) неизвестна.

Чтобы из множества решений выделить наиболее приемлемое, можно воспользоваться регуляризацией, в частности, итерационным методом регуляризации, в котором в роли регуляризатора выступает число итераций: итерационный процесс минимизации (5) продолжается до тех пор, пока $J(\mathbf{w}^k, \mathbf{p}^k) > \int_Z \delta^2 d\mathbf{z}$ (условие Морозова).

Поскольку функция u неизвестна, то предлагается аппроксимировать решение u прямой задачи (1), (2), в которой $k = k_{RBF}$ с помощью РБФ-сети

$$u_{RBF}(\mathbf{x}) = \sum_{n=1}^N w_n G(\mathbf{x}; \mathbf{p}_n)$$

где w_n , $\mathbf{p}_n$ – веса и параметры РБ-функций; N – количество РБ-функций. Неизвестные параметры $\mathbf{p}_n$ и веса РБ-функций w_n можно найти, минимизировав функционал [2]

$$I(\mathbf{w}^u, \mathbf{p}^u) = \int_{\Omega} \left(L(k_{RBF}(\mathbf{x}, u_{RBF})) u_{RBF}(\mathbf{x}; \mathbf{w}^u, \mathbf{p}^u) - f(\mathbf{x}) \right)^2 d\mathbf{x} + \int_{\partial\Omega} \left(Bu_{RBF}(\mathbf{x}; \mathbf{w}^u, \mathbf{p}^u) - p(\mathbf{x}) \right)^2 d\mathbf{x} \quad (6)$$

где $\mathbf{w}^u = \{w_1, w_2, \dots, w_N\}$, $\mathbf{p}^u = \{p_1, p_2, \dots, p_N\}$. Ускорить сходимость процесс решения обратной задачи можно перейдя от задачи поочередной минимизации (5), (6) к задаче минимизации функционала

$$R(\mathbf{w}^u, \mathbf{p}^u, \mathbf{w}^k, \mathbf{p}^k) = I(\mathbf{w}^u, \mathbf{p}^u) + J(\mathbf{w}^k, \mathbf{p}^k).$$

Минимизацию функционала R можно осуществлять с помощью адаптированного авторами метода доверительных областей [3, 4]. Процесс продолжается до тех пор, пока $J(\mathbf{w}^k, \mathbf{p}^k) > \int_Z \delta^2 d\mathbf{z}$.

Экспериментальные исследования предложенного подхода проводились на примере коэффициентной обратной задачи для эллиптического уравнения

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + c(y)u = 0, \quad \mathbf{x} \in \Omega$$

$$u(\mathbf{x}) = 1 + x, \quad \mathbf{x} \in \partial\Omega$$

и дополнительных данных, полученных из измерений на границе расчетной области в виде:

$$\frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) = \psi^\delta(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega,$$

где $\Omega = \{\mathbf{x} | \mathbf{x} = (x, y), 0 < x < 1, 0 < y < 1\}$; $\partial\Omega$ – граница расчетной области, представляющая из себя единичный квадрат; $c(y)$ – неизвестный коэффициент; $\mathbf{n}$ – внешняя по отношению к Ω нормаль, ψ^δ известная функция, заданная с точностью δ .

Погрешность восстановления младшего коэффициента $c(y)$ оценивалась по формуле относительной среднеквадратической погрешности:

$$\sqrt{\sum_{i=1}^M (u_c(y_i) - c(y_i))^2} / \sqrt{\sum_{i=1}^M c(y_i)^2},$$

где M – количество контрольных точек из отрезка $[0;1]$. Сеть u_{RBF} состояла из 14 нейронов, центры которых в начальный момент времени были случайным образом расположены в квадратной области ограниченной точками $(0;0)$ и $(1;1)$, ширина нейронов выбиралась случайным образом из интервала $[0,25;0,45]$, начальные значения весов РБ-функций так же выбирались случайным образом из интервала $[-0,05;0,05]$. Сеть k_{RBF} состояла из 3 нейронов, центры которых в начальный момент времени были случайным образом расположены на отрезке $[0;1]$, ширина нейронов выбиралась случайным образом из интервала $[0,45;0,55]$, начальные значения весов РБ-функций так же выбирались случайным образом из интервала $[-0,05;0,05]$. Для обучения использовались 144 случайно выбранные контрольные точки, 100 из которых располагалось в области решения Ω , 44 – в области $\partial\Omega$. Значение штрафного множителя λ_1 равнялось 1000. В общей сложности были решены три обратные задачи для $\delta = 0$, $\delta = 0,04$ и $\delta = 0,08$. Погрешность восстановления при $\delta = 0$ равнялась 0,031, при $\delta = 0,04$ – 0,037, при $\delta = 0,08$ – 0,042.

Результаты экспериментальных исследований позволяют сделать вывод об эффективности предложенного подхода. Действительной, он универсален, прост в реализации и устойчив к погрешности во входных данных.

Список литературы

1. Вабищевич, П. Н. Численные методы решения задач математической физики : учеб. пособие. / П. Н. Вабищевич, А. А. Самарский. – 3-е. изд. – М. : Изд-во ЛКИ, 2009. – 480 с.
2. Жуков, М. В. Решение коэффициентных обратных задач математической физики с помощью сетей радиальных базисных функций / М. В. Жуков // Нейрокомпьютеры: разработка и применение. – 2014. – № 2. – С. 32–39.
3. Горбаченко, В. И. Подходы и методы обучения сетей радиальных базисных функций для решения задач математической физики / В. И. Горбаченко, М. В. Жуков // Нейрокомпьютеры: разработка и применение. – 2013. – № 9. – С. 12–19.

4. Горбаченко, В. И. Обучение сетей радиальных базисных функций методом доверительных областей для решения уравнения Пуассона / В. И. Горбаченко, М. В. Жуков // Информационные технологии. – 2013. – № 9. – С. 65–70.

Н. В. Слепцов

Россия, Пенза, Пензенский государственный университет

ВЫБОР ЭФФЕКТИВНОЙ СХЕМЫ КОДИРОВАНИЯ ПРИ ЭВОЛЮЦИОННОМ ПОИСКЕ

Рассмотрены особенности двоичного и недвоичного генетического кодирования. Выделены в качестве инструмента повышения результативности генетических преобразований действия с шаблонами строк и переход от операций со строками к операциям с множествами ключевых шаблонов. Обосновывается необходимость универсального представления кодировки.

1. Преобразования в генетических алгоритмах и эффективность поиска

Поиск путей эффективных решений сложных комплексных проблем, например, проблем искусственного интеллекта, обеспечил вовлечение в спектр применяемых методов идей биологической эволюции. Один из подходов состоит в том, чтобы заимствовать идеи эволюции на основе естественного отбора и использовать их для решения проблем.

Класс эволюционных методов вычислений представлен генетическими алгоритмами (ГА), генетическим и эволюционным программированием [1].

ГА работает с множествами особей (элементов) – популяциями. Элементы оцениваются с помощью функции оценки, после чего производится селекция, отбирающая пары элементов из популяции с вероятностью, пропорциональной приспособленности. Пары подвергаются рекомбинации, потомок помещается в новую популяцию. Процесс селекция → рекомбинация повторяется до заполнения новой популяции. Рекомбинация – это процесс объединения двух строк для формирования потомков, он включает операции кроссовера и мутации.

Представление элементов популяции осуществляется строками символов, которые по их естественной биологической аналогии называют хромосомами. Позиции в строке – локусу – соответствует ген – некоторое множество, содержащее символы, которые могут появиться в отдельной позиции строки. Элементы гена называются его аллелями.

Для ГА внешняя среда, в которой особи существуют, интерпретируются как задача, которая должна быть решена, а особи или организмы, приспособленные к среде, являются решениями этой задачи, соответственно, при применении ГА к реальной задаче, должны быть определены: а) генетическая кодировка для задачи и б) функция оценки (пригодности). Кодирование для решения определенной задачи зависит от двух составляющих: а) от длины строки и генов, расположенных в ней, что соответствует представлению задачи, и б) от способа отображения между строками и возможными решениями задачи.

Причины эффективности поиска, основанного на случайном обмене и изменении подстрок совместно со смещенным, но случайным методом селекции строк, полученных рекомбинацией, состоят в следующем.

Рассмотрим способ сохранения внутри строк отображений аллелей. Отображения аллелей – шаблоны – представляют строки с дополнительным отличительным признаком «#», который разрешен в каждой позиции. Признак интерпретируется как «не имеет значения», и является групповым символом. Шаблонами для задачи представления $\langle 3, 2, 2 \rangle$ являются ###, ##0, #1#, 2##, #00, 0#1 и 210.

Шаблоны определены двумя величинами – порядком – числом определенных в нем аллелей (число отличных от # символов) и разрешением (длиной разрешения) – расстоянием между первой и последней определенной аллелью. Эти величины определяют поведение шаблона при рекомбинации: порядок определяет, насколько вероятно шаблон будет разрушен при мутации, разрешение определяет, с какой вероятностью шаблон может быть разрушен кроссовером.

Вероятность выживания шаблона зависит от вероятности мутации и используемого оператора кроссовера.

При обработке строк происходит неявная обработка шаблонов. Игнорируя шаблоны, которые, вероятно, будут разрушены при рекомбинации, для каждой n обработанных строк (популяций размера n) число обработанных шаблонов по оценке составляет $O(n^3)$ [2].

Идея «свободной» обработки такого рода аналогична неявному параллелизму при обработке двоичного дерева, при котором одно текущее сравнение исключает из последующего в среднем $n/2$ элементов, где n – число элементов на рассматриваемой ветви. В ГА свободная обработка и более сложна, и имеет вероятностные характеристики: шаблоны обрабатываются, но с этой обработкой связана определенная степень погрешности.

Можно представить алгоритм явной обработки множества шаблонов, таких, что обработка этих шаблонов вызывает неявную обработку других шаблонов. Их можно определить как *ключевые* шаблоны, они являются точно теми шаблонами, которые *должны* быть обработаны для всех обрабатываемых шаблонов.

Покажем, как обработка множества ключевых шаблонов может привести к свободной обработке всех других шаблонов.

Рассмотрим представление $\langle 2, 2 \rangle$. Для однобитового аргумента при обработке 1# и #1 шаблоны 0# и #0 обрабатывались как побочные эффекты. Однако ни один из двухбитовых шаблонов не будет обработан как побочный эффект. Так, если известно, что именно одно появление каждого из однобитовых шаблонов, это не дает информации о содержании в популяции 00, 11 или 01, 10. Если же обрабатывается также и 11, все остальные двухбитовые шаблоны обрабатываются неявно, поскольку если известно, сколько имеется значений 11, то значения 1#, не связанные с 11, могут появиться только в 10. Оставшееся значение #1 может появиться лишь в 01, и остальная часть популяции должна быть представлена значениями 00. Для этого примера ключевыми шаблонами являются 1#, #1 и 11.

Может быть доказано, что, если есть $n_s - 1$ строк, то для того, чтобы все оставшиеся были обработаны как побочный эффект, должны быть обработаны $n_s - 1$ ключевых шаблонов (при рассмотрении удельного веса строк в популяции используется $n_s - 1$ ключевой шаблон) [2]. Таким образом, можно ограничиться рассмотрением ключевых шаблонов с обработкой остальных ситуаций автоматически как побочный эффект.

2. Представление данных для ГА.

Сравнительный анализ двоичного и двоично-кодированного представлений

Двоичное представление задачи для ГА традиционно означает, что решение задачи представляется в виде двоичных строк фиксированной длины, а двоично-кодированное предполагает представление задачи строками фиксированной длины, содержащих аллели из дискретных генов, из которых все или часть интерпретируются не как двоичные. Таким образом двоичное представление не является подмножеством двоично-кодированного. Для точной характеристики вводится термин «обобщенное», или «унифицированное представление», оно применяется для описания генов, являющихся конечными множествами величин строк фиксированной длины и задач, основанных на представлении таких генов.

Двоичное кодирование обеспечивает следующие положительные свойства: 1 – количество шаблонов и 2 – охват аллели.

Преимущества двоичного представления в том, что оно использует больше шаблонов, чем эквивалентное двоично-кодированное, например, для представления $\langle 4 \rangle$ используется 5 шаблонов (0, 1, 2, 3, #), а представление $\langle 2, 2 \rangle$ с тем же самым числом строк использует 9 шаблонов (00, 01, 10, 11, 0#, 1#, #0, #1, ##). Большее число шаблонов дает больше информации для ГА и обеспечивает его лучшее функционирование.

Применение шаблонов может вызвать неожиданные ситуации при обработке. Частично это связано с тем, что чем больше разрешающая длина и порядок шаблона, тем больше вероятность разрушения этого шаблона при рекомбинации. Короткие шаблоны низких порядков могут явиться источником искажений из-за высокой степени взаимодействия друг с другом. Такие взаимодействия ослабляются длинными шаблонами высоких порядков, однако последние чаще разрушаются из-за своих размеров. Замена нескольких генов одним геном более высокого порядка «замораживает» шаблон, поскольку он не уничтожается кроссовером и становится менее восприимчивым к мутации.

Таким образом, двоичное кодирование дает больше шаблонов, чем двоично-кодированное, но особых преимуществ это не дает поскольку эти шаблоны могут содержать информацию, препятствующую эффективной работе ГА. Двоично-кодированное представление дает меньше шаблонов, но более высокого порядка.

Альтернативная интерпретация шаблонов. Основным аргументом для применения двоично-кодированного представления является альтернативная интерпретация. С этой точки зрения символ # означает не «без различия», а «0 или 1». Такой подход ничего не меняет для двоичного представления, но для кодированного вносит радикальные изменения. Рассмотрим триарный ген {0,1,2}. Вместо единственного символа «без разницы» теперь для различных подмножеств будет задано множество «мелкомодульных» символов безразличия: #₀₁ – позиция, где в строке могут появиться 0 или 1, #₀₂ – 0 или 2, аналогично #₁₂ и вместо традиционного # используется #₀₁₂.

Для строк длины a и всех генов порядка c^* при традиционной интерпретации имеем $(c^* + 1)^a$ шаблонов, альтернативная интерпретация дает $(2^{c^*} - 1)^a$, т.е. кодированное представление обеспечивает для одинакового числа строк намного больше шаблонов.

Данное обстоятельство говорит в поддержку двоично-кодированного представления, но вопросы увеличения числа шаблонов, взятые сами по себе, обходят проблему качества шаблонов.

Альтернативная интерпретация шаблонов позволяет проводить обработку более свободно и не оказывает влияния на саму идею ключевых шаблонов, за исключением того, что в этом случае можно обобщить понятие шаблона до ключевого шаблона. Размер множества ключевых шаблонов при этом остается неизменным. Например, если для популяции известны удельные веса шаблонов 1 и 2, тогда удельный вес $\#_{12}$ составит их сумму. Это распространяется на любые шаблоны с «мелкомодульными» символами безразличия и показывает, что множества ключевых шаблонов, рассмотренные ранее, таковыми остаются для любой интерпретации.

Отсюда вновь возникает интерпретация двоично-кодированных генов как зафиксированных или замороженных шаблонов. Если ГА обрабатывает множества недвоичных аллелей, то в альтернативной интерпретации – множество шаблонов. Подход применим как к двоичному, так и к недвоичному кодированию и означает, что ГА обрабатывает одновременно все возможные множества строк – 2^{n_s} шаблонов.

Заключение

В завершение сравнительного рассмотрения представлений, можно прийти к выводу, что нет явных причин предпочесть один способ другому. Поэтому для обеспечения единой основы представления можно предложить введение единого унифицированного способа кодирования задачи для реализации ее средствами ГА.

Список литературы

1. Емельянов, В. В. Теория и практика эволюционного моделирования / В. В. Емельянов, В. В. Курейчик, В. М. Курейчик. – М. : ФМЛ, 2003. – 432 с.
2. Слепцов, Н. В. Методы эволюционной информатики в системах искусственного интеллекта / Н. В. Слепцов. – Пенза : Изд-во ПГУ, 2010. – 316 с.

И. В. Бойков, В. А. Руднев, А. И. Бойкова, О. А. Баулина, Р. В. Столяров

Россия, Пенза, Пензенский государственный университет

ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ НА НЕЙРОННЫХ СЕТЯХ ХОПФИЛДА

Дан обзор работ авторов по решению уравнений математической физики на нейронных сетях Хопфилда. Предложены алгоритмы решения систем линейных и нелинейных алгебраических уравнений, интегральных уравнений Фредгольма, гиперсингулярных интегральных уравнений и уравнений в частных производных на нейронных сетях Хопфилда. В основу предлагаемых алгоритмов положены методы устойчивости решений систем обыкновенных дифференциальных уравнений.

1. Введение

Применение нейронных сетей Хопфилда (НСХ) для решения задач математической физики основано на возможности представления нейрона в виде электронной схемы, описываемой нелинейным обыкновенным дифференциальным уравнением. Согласно этому представлению i -й нейрон, соединенный с N – нейронами сети (включая самого себя), описывается уравнением

$$C_i \frac{du_i}{dt} = -\frac{u_i}{R_i} + \sum_{j=1}^N w_{ij} f(u_j) + I_i, \quad i = 1, 2, \dots, N, \quad (1.1)$$

где w_{ij} – синаптические веса нейронов сети; I_i – ток, представляющий внешнее смещение; u_i – индуцированное локальное поле на входе функции активации $f(u_i)$; $f(u_i)$ – нелинейные функции активации; R_i и C_i – сопротивление утечки и емкость утечки, соответственно.

В 80-х г. прошлого столетия начали активно развиваться методы моделирования численных методов на искусственных нейронных сетях [1]–[3]. Подробная библиография работ, выполненных в этом направлении содержится в книгах [3]–[8].

Наряду с моделированием известных численных методов на искусственных нейронных сетях (ИНС), представляет интерес развитие специальных методов, предназначенных для решения задач математической физики на ИНС и, в первую очередь, на НСХ.

Приведем краткий обзор работ, выполненных в данном направлении. В работе [9]. J. J. Hopfield исследует возможность применения вычислительных свойств биологических организмов к конструированию вычислительных машин. В основу архитектуры этих машин положено очень большое число взаимосвязанных и очень простых однотипных вычислительных узлов (названных нейронами).

В работе [10] показана возможность реализации подобных компьютеров, получивших название НСХ, используя простые цепи составленные из сопротивлений, емкостей и индуктивностей. В [10] была введена энергетическая функция и исследована устойчивость нейронных сетей на основе второго метода Ляпунова.

Начиная с этих работ, началось широкое применение НСХ: к задачам оптимизации [11], аналого-цифровым преобразователям [12], обращению матриц [13], параметрической идентификации динамических систем [14].

В работе [15] НСХ применяются для решения систем нелинейных алгебраических уравнений.

Опишем архитектуру НСХ, используемой в данной работе. Предлагаемая сеть, состоящая из n нейронов, показана на рис. 1.

В нейронную сеть входят нелинейные устройства, реализующие нелинейные функции $f_i(x_1, x_2, \dots, x_n)$, $i = 1, 2, \dots, m$. Эти устройства обозначены на рис. 1 буквой BN (блок нелинейных элементов).

Выходные сигналы $f_i(x_1, x_2, \dots, x_n)$, $i = 1, 2, \dots, n$, суммируются в блоке сумматора с синаптическими коэффициентами w_{ij} , $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$, и после прохождения RC цепочки подаются на вход устройства активации, реализующего функцию $x = \varphi(u)$.

В данной архитектуре используется функция активации $\varphi(u) = au$.

В известных авторам работах численные методы решения задач математической физики на ИНС основаны на методах минимизации функционалов.

В данной работе дан обзор работ авторов, посвященных методам решения уравнений математической физики на НСХ. В этих работах в основу построения алгоритмов положены методы устойчивости решений систем обыкновенных дифференциальных уравнений.

Приведем обозначения, используемые в работе:

$R(a, r) = \{z \in B : \|z - a\| \leq r\}$, $S(a, r) = \{z \in B : \|z - a\| = r\}$, $\operatorname{Re} K = K_R = (K + K^*)/2$, $\Lambda(K) = \lim_{h \downarrow 0} (\|I + hK\| - 1)h^{-1}$. Здесь B – банахово пространство, $a \in B$, K – линейный оператор, действующий из B в B , $\Lambda(K)$ – логарифмическая норма [16] оператора K ; через I обозначен тождественный оператор.

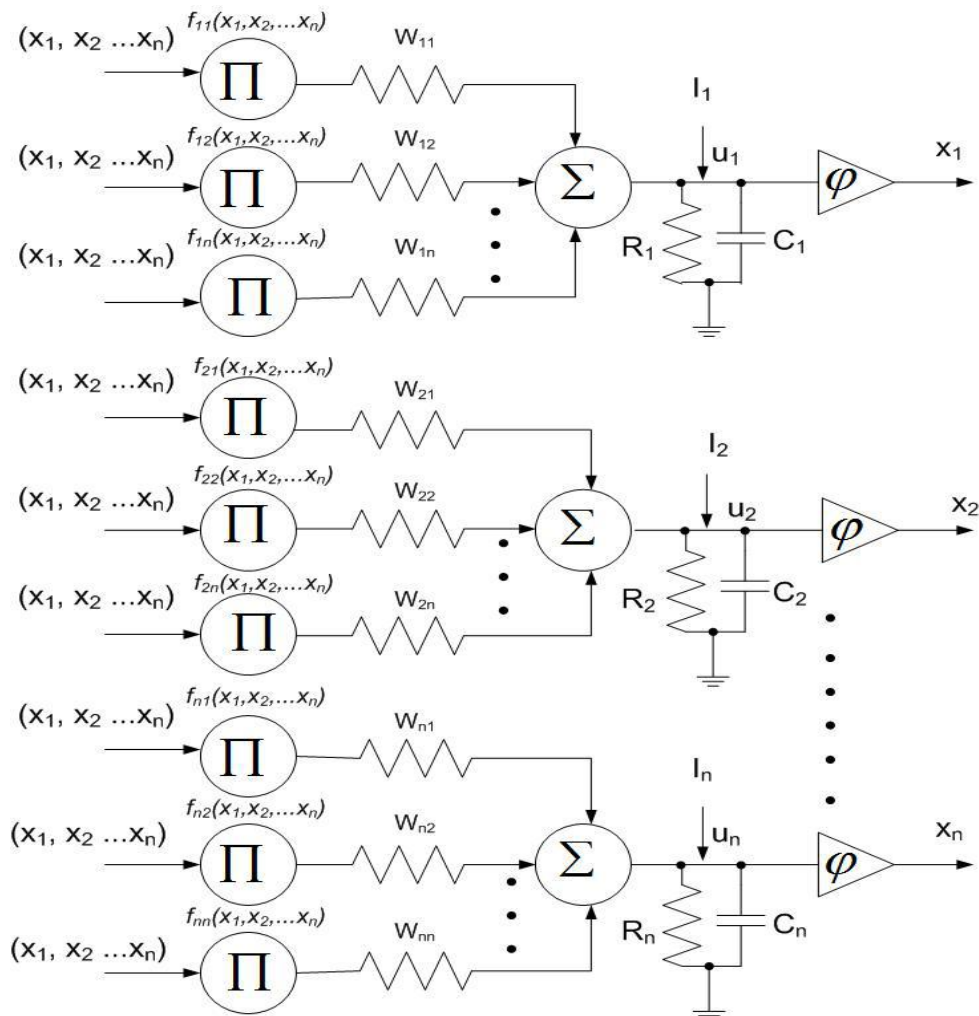


Рис. 1. Схема нейронной сети

2. Непрерывные методы решения операторных уравнений

Приближенным методам решения нелинейных операторных уравнений посвящена обширная литература, подробная библиография которой содержится в книгах [17], [18]. При этом, в основном, рассматривались дискретные методы, среди которых в первую очередь следует отметить методы простой итерации и Ньютона–Канторовича. Исследование непрерывных аналогов метода Ньютона–Канторовича началось, по-видимому, со статьи [19]. Позднее непрерывные аналоги метода Ньютона–Канторовича широко применялись при решении многочисленных задач физики [19].

Приведем несколько утверждений о непрерывных методах решения операторных уравнений, которые ниже будут использованы при обосновании вычислительных методов.

Рассмотрим нелинейное операторное уравнение

$$A(x) = 0, \quad (2.1)$$

действующее из банахова пространства B в B . Здесь $A(x)$ – нелинейный оператор. Рассмотрим в банаховом пространстве B задачу Коши

$$\frac{dx(t)}{dt} = A(x(t)), \quad x(0) = x_0, \quad A(0) = 0. \quad (2.2)$$

Будем считать, что оператор A имеет непрерывную производную Гато.

Теорема 2.1 [20]. Пусть уравнение (2.1) имеет решение x^* . Пусть на любой дифференцируемой кривой $g(t)$, расположенной в банаховом пространстве B , справедливо неравенство

$$\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \Lambda(A'(g(\tau))) d\tau \leq -\alpha, \quad \alpha > 0. \quad (2.3)$$

Тогда решение задачи Коши (2.2) сходится к решению x^* уравнения (2.1) при любом начальном приближении.

Теорема 2.2 [21]. Пусть уравнение (2.1) имеет решение x^* . Пусть на любой дифференцируемой кривой $g(t)$, расположенной в шаре $B(x^*, r)$ выполняются следующие условия:

1) при любом $t(t > 0)$ выполняется неравенство $\int_0^t \Lambda(A'g(\tau)) d\tau \leq 0$;

2) справедливо равенство $\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \Lambda(A'(g(\tau))) d\tau = -\alpha, \alpha > 0$.

Тогда решение задачи Коши (2.2) сходится к решению x^* уравнения (2.1).

Непрерывный метод решения нелинейных уравнений имеет следующие преимущества относительно стандартного метода Ньютона-Канторовича:

1) не требуется существования обратного оператора для производной Фреше нелинейного оператора;

2) в случае выполнения неравенства $\int_0^t \Lambda(C(g(\tau))) d\tau < 0$ на любой дифференцируемой функции $g(t)$, сходимость метода не зависит от начальных условий.

3. Приближенное решение алгебраических уравнений на нейронных сетях Хопфилда

Рассмотрим систему линейных алгебраических уравнений

$$Ax = b, \quad (3.1)$$

где $A = \{a_{ij}\}$, $i, j = 1, 2, \dots, n$, $x = (x_1, \dots, x_n)$, $b = (b_1, \dots, b_n)^T$.

Пусть логарифмическая норма $\Lambda(A)$ матрицы A отрицательна.

Поставим системе алгебраических уравнений (3.1) в соответствие систему обыкновенных дифференциальных уравнений

$$\frac{dx(t)}{dt} = Ax - b \quad (3.2)$$

с начальным условием

$$x(0) = 0. \quad (3.3)$$

Из результатов предыдущего раздела следует, что при $t \rightarrow \infty$ решение системы обыкновенных дифференциальных уравнений (3.2) сходится к решению $x^* = A^{-1}b$ системы алгебраических уравнений (3.1).

Таким образом, моделирование уравнения (3.2) на нейронных сетях Хопфилда позволяет при достаточно больших значениях t получить хорошее приближение к решению системы уравнений (3.1). Можно показать, что решение устойчиво к возмущению коэффициентов и правых частей уравнений.

Рассмотрим нелинейное операторное уравнение

$$A(x) - f = 0, \quad (3.4)$$

действующее из банахова пространства X в X . Здесь $A(x)$ – нелинейный оператор.

Пусть уравнение (3.4) имеет изолированное решение x^* .

Рассмотрим задачу Коши

$$\frac{dx(t)}{dt} = A(x(t)) - f, \quad x(0) = x_0 \quad (3.5)$$

Теорема 3.1 [21]. Пусть уравнение (3.4) имеет решение x^* . Пусть на любой дифференцируемой кривой $g(t)$, расположенной в банаховом пространстве B , справедливо неравенство $\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \Lambda(A'(g(\tau))) d\tau = -\alpha, \alpha > 0$. Тогда решение задачи

Коши (3.2) сходится к решению x^* уравнения (3.2) при любом начальном приближении.

Теорема 3.2. Пусть уравнение (3.4) имеет решение x^* . Пусть на любой дифференцируемой кривой $g(t)$, расположенной в шаре $B(x^*, r)$ выполняются следующие условия:

- 1) при любом $t (t > 0)$ выполняется неравенство $\int_0^t \Lambda(A'g(\tau)) d\tau \leq 0$;
- 2) справедливо неравенство $\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \Lambda(A'(g(\tau))) d\tau = -\alpha, \alpha > 0$.

Тогда решение задачи Коши (3.5) сходится к решению x^* уравнения (3.4).

Замечание. Подробное изложение методов решения алгебраических уравнений на НСХ приведено в [21]. Там же приведены модельные примеры.

4. Приближенное решение интегральных уравнений Фредгольма на нейронных сетях Хопфилда

Решение интегральных уравнений Фредгольма на нейронных сетях Хопфилда изложим на примере одномерного интегрального уравнения

$$x(t) = \int_0^1 h(t, \tau, x(\tau)) d\tau + f(t) \quad (4.1)$$

с непрерывным ядром и непрерывной правой частью. Предположим, что уравнение (4.1) имеет решение $x^*(t)$ в шаре $B(x^*, r)$ пространства $C[0, 1]$.

Приближенное решение $x_N(t)$ уравнения (4.1) определяется из системы уравнений

$$x_N(t_{kN}) = \sum_{l=1}^N \alpha_{lN} h(t_{kN}, t_{lN}, x_N(t_{lN})) + f(t_{kN}), \quad k = 1, 2, \dots, N, \quad (4.2)$$

где α_{lN} и t_{lN} , $l = 1, 2, \dots, N$, коэффициенты и узлы квадратурной формулы

$$\int_0^1 g(t) dt = \sum_{l=1}^N \alpha_{lN} g(t_{lN}) + R_N(g).$$

Условия разрешимости системы (4.2) и сходимости приближенных решений $x_N^*(t)$ системы уравнений (4.2) к точному решению $x^*(t)$ уравнения (4.1) в узлах $t_{lN}, l = 1, 2, \dots, N$, приведены в [17] (теорема 19.5 из главы 4).

Наложим на функцию $h(t, \tau, u)$ следующее условие: во всякой внутренней точке области $B(x^*, r)$ пространства $C[0, 1]$ существует производная $h'_3(t_{kN}, t_{lN}, u), k, l = 1, 2, \dots, N$. Здесь $h'_3(t, \tau, u)$ означает производную по третьей переменной.

Рассмотрим матрицу

$$C(u) = \{c_{ij}(u)\}, i, j = 1, 2, \dots, N,$$

где $c_{ii}(u) = 1 - \alpha_{iN} h(t_{iN}, t_{iN}, u)$, $i = 1, 2, \dots, N$; $c_{ij}(u) = -\alpha_{iN} h(t_{iN}, t_{jN}, u)$, $i, j = 1, 2, \dots, N, i \neq j$.

Из теоремы 2.1 следует, что если $\Lambda(C(u)) < 0$ при $u \in B(x^*, r)$, то решение системы дифференциальных уравнений

$$\frac{dz_{kN}(t)}{dt} = -(z_{kN}(t) - \sum_{l=1}^N \alpha_{lN} h(t_{kN}, t_{lN}, z_{lN}(t)) - f(t_{kN})), k = 1, 2, \dots, N, \quad (4.3)$$

сходится к решению $x^*(t)$ системы уравнений (5.2) в узлах $t_{kN}, k = 1, 2, \dots, N$.

Замечание. Условие $\Lambda(C(u)) < 0$ носит достаточный характер и, как показывают модельные примеры, решение системы (4.3) при $t \rightarrow \infty$ сходится к $x^*(t)$ при более широких условиях.

5. Приближенное решение линейных гиперсингулярных интегральных уравнений на нейронных сетях Хопфилда

Рассмотрим одномерное линейное гиперсингулярное интегральное уравнение

$$Kx \equiv a(t)x(t) + b(t) \int_{-1}^1 \frac{x(\tau) d\tau}{(\tau - t)^p} + \int_{-1}^1 h(t, \tau)x(\tau) d\tau = f(t), \quad p = 2, 4, \dots \quad (5.1)$$

На коэффициенты и правую часть уравнения (5.1) наложим следующие условия:

1) функция $b(t) \neq 0$ на сегменте $[-1, 1]$;

2) функции $a(t), b(t), f(t) \in W^r(1), h(t, \tau) \in W^{r,r}(1), r \geq p$.

3) уравнение (5.1) однозначно разрешимо и его решение $x^*(t) \in W^r(M), M = \text{const}$.

Введем узлы $t_k = -1 + 2k/N, k = 0, 1, \dots, N$, и $\bar{t}_k = t_k + k/N, k = 0, 1, \dots, N-1$.

Обозначим через Δ_k сегменты $\Delta_k = [t_k, t_{k+1}]$, $k = 0, 1, \dots, N-1$.

Приближенное решение уравнения (5.1) будем искать в виде кусочно-постоянной функции

$$x_N(t) = \sum_{k=0}^{N-1} \alpha_k \psi_k(t), \psi_k(t) = \begin{cases} 1, & t \in \Delta_k \\ 0, & t \notin \Delta_k. \end{cases} \quad (5.2)$$

Значения $\{\alpha_k\}$, $k = 0, 1, \dots, N-1$, определяются из системы линейных алгебраических уравнений

$$a(\bar{t}_k) \alpha_k + b(\bar{t}_k) \sum_{l=0}^{N-1} \alpha_l \int_{\Delta_l} \frac{d\tau}{(\tau - \bar{t}_k)^p} + \sum_{l=0}^{N-1} \alpha_l \int_{\Delta_l} h(\bar{t}_k, \tau) d\tau = f(\bar{t}_k), \quad k = 0, 1, \dots, N-1. \quad (5.3)$$

Здесь $\sum'$ означает суммирование по $l \neq k-v, k-v+1, \dots, k-1, k+1, \dots, k+v-1$, где величина $v(v \geq 1)$ зависит от абсолютной величины коэффициентов $a(t)$, $b(t)$ и от значений N . Способ выбора v описан в работе [22].

Теорема 5.1 [22]. Пусть выполнены следующие условия: 1) уравнение (5.1) имеет единственное непрерывно-дифференцируемое до $p-1$ порядка решение $x^*(t)$; 2) справедливо неравенство $|b(t)| \geq b > 0$ при $t \in (-1, 1)$. 3) функция $h(t, \tau)$ удовлетворяет условию Липшица по второй переменной. Тогда при достаточно больших N система уравнений (5.3) имеет единственное решение $x_N^*(t)$ и в метрике пространства R_N справедлива оценка $\|x^* - x_N^*\| \asymp N^{-1}$.

При выполнении условий теоремы 5.1 система уравнений (5.3) может быть решена на нейронных сетях Хопфилда.

Для этого систему (5.3) следует представить в эквивалентном виде

$$(\text{sgnb}(\bar{t}_k)) \left[a(\bar{t}_k) \alpha_k + b(\bar{t}_k) \sum_{l=0}^{N-1} \alpha_l \int_{\Delta_l} \frac{d\tau}{(\tau - \bar{t}_k)^p} + \sum_{l=0}^{N-1} \alpha_l \int_{\Delta_l} h(\bar{t}_k, \tau) d\tau - f(\bar{t}_k) \right] = 0, \quad k = 0, 1, \dots, N-1. \quad (5.4)$$

Из утверждений, приведенных в разделе 2 следует, что решение системы уравнений

$$\frac{d\alpha_k(t)}{dt} = (\text{sgnb}(\bar{t}_k)) \left[a(\bar{t}_k) \alpha_k(t) + b(\bar{t}_k) \sum_{l=0}^{N-1} \alpha_l \int_{\Delta_l} \frac{d\tau}{(\tau - \bar{t}_k)^p} + \sum_{l=0}^{N-1} \alpha_l \int_{\Delta_l} h(\bar{t}_k, \tau) d\tau - f(\bar{t}_k) \right],$$

$k = 0, 1, \dots, N-1$, при $t \rightarrow \infty$ сходится к решению системы уравнений (5.4) при любом начальном приближении.

Исследуем приближенные методы решения нелинейных гиперсингулярных интегральных уравнений

$$a(t, x(t)) + \int_{-1}^1 \frac{h(t, \tau, x(\tau)) d\tau}{(\tau - t)^p} = f(t), \quad p = 2, 4, 6, \dots \quad (5.5)$$

Рассмотрим уравнение (5.5) при $p = 2$. Приближенное решение уравнения (5.5) будем искать в виде кусочно-постоянной функции (5.2), коэффициенты которой определяются из системы нелинейных алгебраических уравнений

$$a(\bar{t}_k, \alpha_k) + \sum_{l=0}^{N-1} \int_{\Delta_l} \frac{h(\bar{t}_k, \bar{t}_l, \alpha_l)}{(\tau - \bar{t}_k)^2} d\tau = f(\bar{t}_k), \quad k = 0, 1, \dots, N-1, \quad (5.6)$$

где $\bar{t}_k = -1 + (2k+1)/N, k = 0, 1, \dots, N-1$.

Запишем уравнение (5.6.) в операторной форме $K_N x_N = F_N$, где $x_N = (\alpha_0, \dots, \alpha_{N-1})^T$, $F_N = (f(\bar{t}_0), \dots, f(\bar{t}_{N-1}))^T$, $K_N - N \times N$ матрица.

Будем считать, что уравнение (5.6) имеет решение x^* в шаре $B(x^*, R)$ пространства R_N и что для любой дифференцируемой кривой $g(t) \in B(x^*, R)$ выполняется неравенство

$$\int_0^t \Lambda(K'_N(g(\tau))) d\tau < 0.$$

Тогда решение системы дифференциальных уравнений

$$\frac{d\alpha_k(t)}{dt} = a(\bar{t}_k, \alpha_k(t)) + \sum_{l=0}^{N-1} \alpha_k(t) \int_{\Delta_l} \frac{h(\bar{t}_k, \bar{t}_l, \alpha_l(\tau))}{(\tau - \bar{t}_k)^2} d\tau - f(\bar{t}_k),$$

$k = 0, 1, \dots, N-1$, сходится к решению $x_N^* = (\alpha_0^*, \dots, \alpha_{N-1}^*)$ системы уравнений (5.6).

Замечание 1. В случае нечетных p в основу построения вычислительных схем следует положить алгоритмы предложенные и обоснованные в работе [23].

Замечание 2. Разделы 6, 7 написаны по материалам работы [24], в которой приведены результаты численных экспериментов.

6. Решение эллиптических уравнений на нейронных сетях Хопфилда

Пусть $D = [0, 1; 0, 1]$. В области D рассмотрим задачу Дирихле:

$$\begin{aligned} & a(x_1, x_2) \Delta u(x_1, x_2) + b(x_1, x_2) \frac{\partial u(x_1, x_2)}{\partial x_1} + \\ & + c(x_1, x_2) \frac{\partial u(x_1, x_2)}{\partial x_2} + d(x_1, x_2) u(x_1, x_2) = f(x_1, x_2) \end{aligned} \quad (6.1)$$

с граничными условиями

$$u(x_1, x_2)|_{\Gamma} = u_0(x_1, x_2), \quad \Gamma = \partial D. \quad (6.2)$$

Введем сетку узлов (v_k, v_l) , $k, l = 0, 1, \dots, N$, где $v_k = k/N$, $k = 0, 1, \dots, N$. Здесь $h = 1/N$.

Задача Дирихле (6.1), (6.2) аппроксимируется разностной схемой

$$\begin{aligned} & a(v_k, v_l) \left(\frac{u(v_{k-1}, v_l) + u(v_{k+1}, v_l) - 2u(v_k, v_l)}{h^2} + \frac{u(v_k, v_{l+1}) + u(v_k, v_{l-1}) - 2u(v_k, v_l)}{h^2} \right) + \\ & + b(v_k, v_l) \frac{u(v_{k+1}, v_l) - u(v_k, v_l)}{h} + c(v_k, v_l) \frac{u(v_k, v_{l+1}) - u(v_k, v_l)}{h} + \\ & + d(v_k, v_l) u(v_k, v_l) - f(v_k, v_l) = 0, \quad k, l = 1, 2, \dots, N-1. \end{aligned} \quad (6.3)$$

Системе уравнений (6.3) поставим в соответствие систему дифференциальных уравнений

$$\begin{aligned} \frac{du_{kl}(t)}{dt} = & \alpha_{kl}(a(v_k, v_l)(u_{k+1,l}(t) + u_{k-1,l}(t) + u_{k,l+1}(t) + u_{k,l-1}(t) - 4u_{k,l}(t)) + \\ & + hb(v_k, v_l)(u_{k+1,l}(t) - u_{k,l}(t)) + hc(v_k, v_l)(u(v_k, v_{l+1}) - u(v_k, v_l)) + \\ & + h^2 d(v_k, v_l)u(v_k, v_l) - h^2 f(v_k, v_l)), \quad k, l = 1, 2, \dots, N-1. \end{aligned} \quad (6.4)$$

Здесь α_{kl} – константы, выбор которых будет описан ниже, $u_{k,l}(t) = u(v_k, v_l, t)$.

Пусть система уравнений (6.3) однозначно разрешима. Обозначим через $u^*(v_k, v_l)$, $k, l = 0, 1, 2, \dots, N$, решение этой системы.

Пусть существует функция $\alpha(t_1, t_2)$ такая, что

$$\begin{aligned} \alpha(t_1, t_2)a(t_1, t_2) &\geq 0, \alpha(t_1, t_2)b(t_1, t_2) \geq 0, \\ \alpha(t_1, t_2)c(t_1, t_2) &\geq 0, \alpha(t_1, t_2)d(t_1, t_2) < 0. \end{aligned} \quad (6.5)$$

при всех $t_1, t_2 \in D$. Отсюда следует, что система уравнений (6.4) устойчива в целом при любых начальных возмущениях, не нарушающих граничных значений.

Поэтому решение системы уравнений (6.4) сходится при $t \rightarrow \infty$ к решению $u^*(v_k, v_l)$, $k, l = 1, 2, \dots, N-1$, системы уравнений (6.3).

Рассмотрим в области $D = [0, 1]^2$ систему нелинейных уравнений

$$\begin{aligned} a(x_1, x_2)g_1(\Delta u(x_1, x_2)) + b(x_1, x_2)g_2\left(\frac{\partial u(x_1, x_2)}{\partial x_1}\right) + \\ + c(x_1, x_2)g_3\left(\frac{\partial u(x_1, x_2)}{\partial x_2}\right) + d(x_1, x_2)g_4(u(x_1, x_2)) = f(x_1, x_2), \end{aligned} \quad (6.6)$$

с граничными условиями

$$u(x_1, x_2)|_{\Gamma} = u_0(x_1, x_2), \quad (6.7)$$

где $\Gamma = \partial D$; $g_i(x)$, $i = 1, 2, 3, 4$ – непрерывно дифференцируемые функции на R_1 .

По аналогии с линейным случаем введем сетку узлов (v_k, v_l) , $k, l = 0, 1, \dots, N$ и поставим граничной задаче (6.6), (6.7) в соответствие систему разностных уравнений

$$\begin{aligned} a(v_k, v_l)g_1\left(\frac{u(v_{k-1}, v_l) + u(v_{k+1}, v_l) - 2u(v_k, v_l)}{h^2} + \frac{u(v_k, v_{l+1}) + u(v_k, v_{l-1}) - 2u(v_k, v_l)}{h^2}\right) + \\ + b(v_k, v_l)g_2\left(\frac{u(v_{k+1}, v_l) - u(v_k, v_l)}{h}\right) + c(v_k, v_l)g_3\left(\frac{u(v_k, v_{l+1}) - u(v_k, v_l)}{h}\right) + \\ + d(v_k, v_l)g_4(u(v_k, v_l)) = f(v_k, v_l), \quad k, l = 1, 2, \dots, N-1. \end{aligned} \quad (6.8)$$

Системе уравнений (6.8) поставим в соответствие систему нелинейных дифференциальных уравнений

$$\begin{aligned} \frac{du_{kl}(t)}{dt} = & \alpha_{kl}(a(v_k, v_l)g_1\left(\frac{u_{k-1,l}(t) + u_{k+1,l}(t) + u_{k,l+1}(t) + u_{k,l-1}(t) - 4u_{kl}(t)}{h^2}\right) + \\ & + b(v_k, v_l)g_2\left(\frac{u_{k+1,l}(t) - u_{k,l}(t)}{h}\right) + c(v_k, v_l)g_3\left(\frac{u_{k,l+1}(t) - u_{k,l}(t)}{h}\right) + \\ & + d(v_k, v_l)g_4(u_{k,l}(t)) - f(v_k, v_l)), \quad k, l = 1, 2, \dots, N-1. \end{aligned} \quad (6.9)$$

Здесь $u_{kl}(t) = u(v_k, v_l, t)$; $k, l = 0, 1, \dots, N$; $u_{0l}(t) = u_0(0, v_l)$, $l = 0, 1, \dots, N$,
 $u_{kN}(t) = u_0(v_k, 1)$, $k = 0, 1, \dots, N$, $u_{Nl}(t) = u_0(1, v_l)$, $l = 0, 1, \dots, N$, $u_{k0}(t) = u_0(v_k, 0)$,
 $k = 0, 1, \dots, N$.

Предположим, что система уравнений (6.8) имеет решение $u^*(v_k, v_l)$, $k, l = 0, 1, \dots, N$.

В работе [25] приведены условия сходимости решения системы дифференциальных уравнений (6.9) к решению системы разностных уравнений (6.8).

Список литературы

1. Dehghan, Mehdi. Numerical solution of Helmholtz equation by the modified Hopfield finite difference technique / Mehdi Dehghan, Mojtaba Nourian, Mohammad B. Menhaj // Numerical Methods for Partial Differential Equations. – 2009. – V. 25, iss. 3. – P. 637–656,
2. Joya, G. Application of high-order Hopfield neural networks to the solution of diophante equations / G. Joya, M. A. Atencia, F. Sandoval // Lecture Notes in Computer Science. – 1991. – V. 540/1991. – P. 395–400.
3. Галушкин А. И. Теория нейронных сетей / А. И. Галушкин. – М. : ИПРЖР, 2000. 416 с.
4. Нейроинформатика / А. Н. Горбань, В. Л. Дунин-Барковский, А. Н. Кирич и др. – Новосибирск : Наука. Сибирское предприятие РАН, 1998. – 296 с.
5. Тархов, Д. А. Нейронные сети как средство математического моделирования / Д. А. Тархов. – М. : Радиотехника. 2006. – 48 с.
6. Горбаченко, В. И. Нейрокомпьютеры в решении краевых задач теории поля / В. И. Горбаченко. – М. : Радиотехника, 2003. – 336 с.
7. Gupta, M. M. Static and Dynamic Neural Networks. From Fundamentals to Advanced Theory / M. M. Gupta, L. Jin, N. Hamma. – New York : CEE Press. Wiley Interscience, 2005. – 722 p.
8. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – 2-е изд. – М. : Вильямс, 2006. – 1104 с.
9. Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities / J. J. Hopfield // Proc. Natl. Acad. Sci. USA. – 1982. – April. – Vol. 79. – P. 2554–2558.
10. Hopfield, J. J. Neural Computation of decision in Optimization problems / J. J. Hopfield, D. W. Tank // Biological Cybernetics. – 1985. – Vol. 52. – P. 141–152.
11. Tank, D. W. Simple Neural Optimization: An A/D Converter, a Single Decision Circuit and Linear Programming Circuit / D. W. Tank, J. J. Hopfield // IEEE Transaction on Circuit and Systems. – 1991. – Vol. 33. – P. 137–142.
12. Jang, J. S. An Optimization Network for Solving a Set of Simultaneous Linear Equations / J. S. Jang, S. Y. Lee, S. Y. Shin // IEEE Proceedings. – 1992. – P. 516–521.
13. Atencia, M. Hopfield Neural Networks for Parametric Identification of Dynamical Systems / M. Atencia, G. Joya, F. Sandoval // Neural Processing Letters. – 2005. – P. 143–152.

14. Mishra, D. Modified Hopfield Neural Network Approach for Solving Nonlinear Algebraic Equations / D. Mishra, P. K. Kalra // Engineering Letters. – 2007. – No 1. – P. 14.
15. Далецкий, Ю. Л. Устойчивость решений дифференциальных уравнений в банаховом пространстве / Ю. Л. Далецкий, М. Г. Крейн. – М. : Наука, 1970. – 534 с.
16. Канторович, Л. В. Функциональный анализ / Л. В. Канторович, Г. П. Акилов. – М. : Наука, 1977. – 750 с.
17. Приближенное решение операторных уравнений / М. А. Красносельский, Г. М. Вайникко [и др.]. – М. : Наука, 1969. – 456 с.
18. Гавурин, М. К. Нелинейные функциональные уравнения и непрерывные аналоги итеративных методов / М. К. Гавурин // Известия вузов. Математика. – 1958. – № 5. – С. 18–31.
19. Пузынина, Т. П. Модифицированные ньютоновские схемы для численного исследования квантово-полевых моделей : автореф. дисс. ... д-ра физ.-мат. наук / Пузынина Т. П. – Тверь : Тверской государственный университет, 2003. – 37 с.
20. Бойков, И. В. Об одном непрерывном методе решения нелинейных операторных уравнений / И. В. Бойков // Дифференциальные уравнения. – 2012. – Т. 48, № 9. – С. 1308–1314.
21. Бойков, И. В. Приближенное решение задач математической физики на нейронных сетях Хопфилда / И. В. Бойков, В. А. Руднев, А. И. Бойкова // Нейрокомпьютеры, разработка, применение. – 2013. – № 10. – С. 13–22.
22. Бойков, И. В. Сплайн – коллокационный метод решения гиперсингулярных интегральных уравнений / И. В. Бойков, А. И. Бойкова // Вестник Харьковского национального университета. Сер. «Математическое моделирование. Информационные технологии. Автоматизированные системы управления». – 2007. – Вып. 7. – С. 36–49.
23. Бойков, И. В. Приближенное решение гиперсингулярных интегральных уравнений с целыми сингулярностями нечетного порядка / И. В. Бойков, А. И. Бойкова // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. Математика. – 2010. – № 3. – С. 15–27.
24. Бойков, И. В. Приближенное решение гиперсингулярных интегральных уравнений на нейронных сетях Хопфилда / И. В. Бойков, О. А. Баулина // Журнал Средневолжского математического общества. – 2013. – Т. 15, № 1. – С. 41–51.
25. Бойков, И. В. Приближенное решение эллиптических уравнений на нейронных сетях Хопфилда / И. В. Бойков, О. А. Баулина // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. Математика. – 2014. – № 1. – С. 39–53.

D. Drozdov^(1, 2), V. Dubinin⁽¹⁾, S. Patil⁽²⁾, V. Vyatkin^(2, 3)

⁽¹⁾ **Russia, Penza, Penza State University**

⁽²⁾ **Sweden, Luleå, Luleå University of Technology**

⁽³⁾ **Finland, Helsinki, Aalto University**

PLANT ABSTRACTION FOR IEC 61499 CLOSED-LOOP MODEL CHECKING

Formal verification of closed-loop plant-controller models allows detection of errors in safety critical control systems which may be omitted at manual or simulation-based testing. In this paper, we propose a way to reduce complexity of the closed-loop SMV models generated from IEC 61499 function blocks by plant abstraction based on Net Condition/Event Systems (NCES). It enables the application of model-checking techniques to verification of IEC 61499 control systems.

1. Introduction

Formal verification is an act of proving or disproving an algorithm with respect to property specifications and model-checking is one such formal verification approach. It enables the unsupervised automatic verification process of a system by generating model's state space and identifies system failure via counterexamples. For safety-critical applications, formal verification is one of the most efficient ways to prove system's correctness. In the last two decades, symbolic model checking technique has proven itself effective on many real projects [1].

IEC 61499 is a new paradigm in distributed automation systems programming that aims at development of highly scalable, reconfigurable and maintainable systems. But for such systems, which may contain hundreds devices with their own controllers operating as a whole, it is not enough to ensure that controllers operate properly, but the whole system is correct. Closed-loop models represent controllers and controlled devices. This modeling pattern can provide additional means to check correctness of complex systems.

2. Closed-loop FB models

State spaces of complicated closed-loop models may be too large for verification, and the main way to reduce model complexity is abstraction. Function block models, based on the FB operational semantics [2, 3, 4], cannot be reduced using abstraction without loose of important execution aspects. In this case, abstraction reduction can be applied only to plant model.

A plant, controlled by IEC 61499 applications can be controlled both by value (level) and event (edge) signals, and Net Condition-Event Systems (NCES) [5, 6] semantics exactly has sufficient expressiveness.

In works [7, 8] modelling and verification of closed-loop automation systems using NCES was described. In work [9] the methods to model Petri net in SMV were proposed. Using first and second methods in combination with event signal transfer rules from FB operational semantics [10], hierarchical NCES models can be transformed to SMV models.

In order to ensure plant model compatibility with automatically generated [11, 12] controller SMV model, reduced plant components are put into function block *wrapper* – SMV module that provides function blocks operational semantics compatible interface in connection with NCES model inside.

This wrapper contains “first-line” transition-place pairs, which provide interface for internal NCES-based model. Each of this pairs can be described in NCES graphical notation as it shown in Fig. 1 (a), but the main difference between FB wrapper and NCES is that all transitions in wrapper’s “first-line” pairs fire simultaneously when the wrapper FB module is active. If an activation event occurs the transition rewrites place marking regard to ‘data input’ value, for simple plant models with Boolean inputs, considered in this paper, the place becomes marked if ‘data input’ value is TRUE, otherwise the place becomes unmarked.

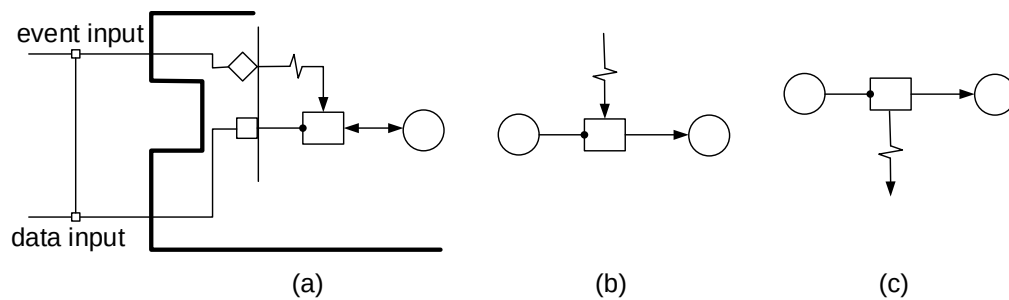


Fig. 1. Wrapper first-line pair (a) and typical NCES transitions (b, c)

Additions to Wiemmel’s model [9]:

- 1) Event activated transition (event consumer (Fig. 1 (b))).

The simple Petri net transition SMV module [4] can be transformed into NCES transition with event trigger by adding event in the ‘enable’ rule and simple unconditional event reset statement. This approach mainly corresponds to events in FB operational semantics.

```

MODULE pnpVacuumOn(event_in, input, output)
DEFINE enabled := input & event_in;
ASSIGN
  next(input) := case
    enabled : FALSE;
    TRUE : input;
  esac;
  next(output) := case
    enabled : TRUE;
    TRUE : output;
  esac;
  next(event_in) := FALSE;

```

- 2) Event emitting transition (Fig. 1 (c)).

Event signal transmission is performed through signal buffers (SMV variables), which are set as ‘true’ when the event emitting module fires and reset to ‘false’ every time when the event consumer module runs.

```

MODULE pnpVacuumOn(input, output, event_out)
DEFINE enabled := input;
ASSIGN
next(input) := case
enabled : FALSE;
TRUE : input;
esac;
next(output) := case
enabled : TRUE;
TRUE : output;
esac;
next(event_out) := case
enabled : TRUE;
TRUE : event_out;
esac;

```

3. Pick and Place manipulator case study

Fig. 2 shows the 3-cylinder *Pick-and-Place (PnP)* manipulator used for case study. It consists of two vertical and one horizontal cylinders and a suction unit. Each cylinder has two sensors indicating two end positions of the piston.

As different to the NCES-based closed-loop model described in works [7, 8], we propose a way to verify full operational semantics-based controller model [3] with plant abstraction.

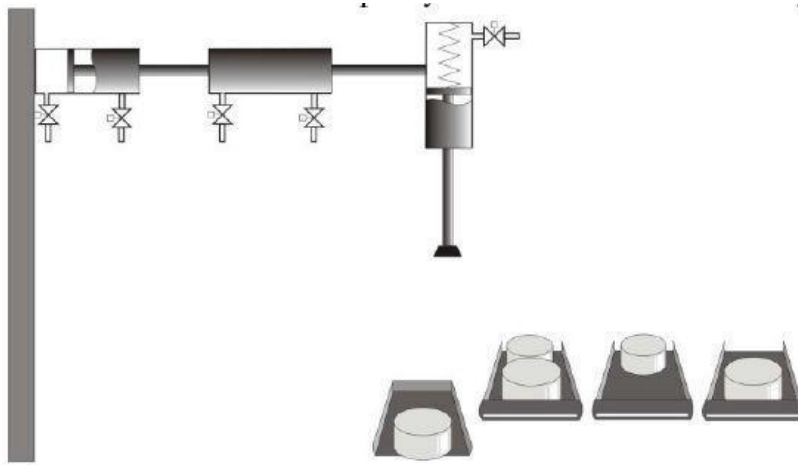


Fig. 2. 3-cylinder PnP manipulator

Fig. 3 shows the closed-loop model schema with *MasterSlave* controller [8] and NCES plant model. At the initial state, the event input *INIT* has value *TRUE*, which corresponds to occurrence of event *INIT* in the beginning of execution. Event output *INITO* connected to input *REQ*. This connection provides entry point for system's main execution path, which may be described as a cycle of occurrence of events

REQ -> CNF -> CYL_IND -> CHG -> REQ ->...

Also, input *pp1* is 'true' and *pp2* and *pp3* are 'false' (workpiece is present in place 1, and places 2 and 3 are empty) for model testing. However, in real systems checking these variables can be left "free" to verify all possible system states, or they can be connected to the environment emulation module.

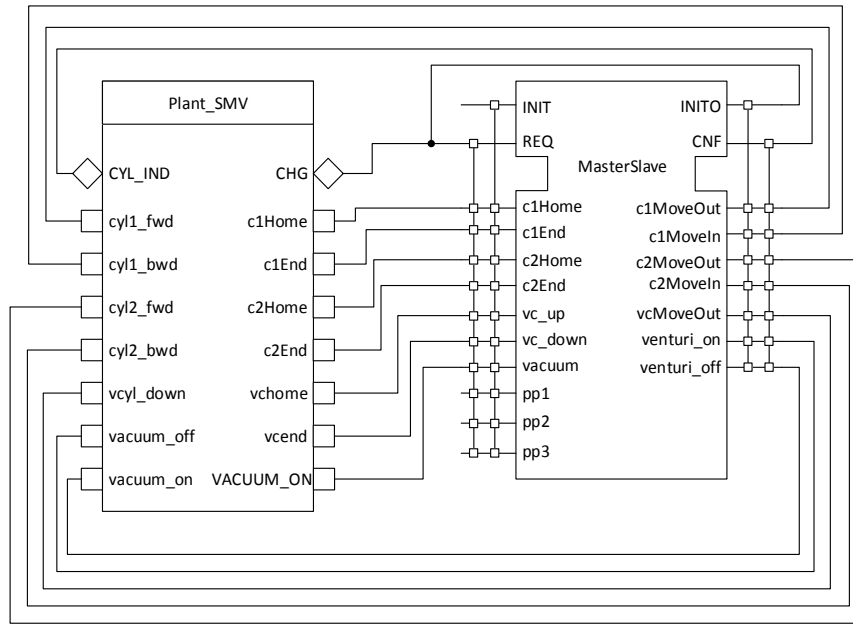


Fig. 3. Closed-loop model schema

The simplest abstract model of cylinder consists of only two states: ‘home’ and ‘end’ and two transition between these states. Transition emit event ‘CHG’ when it fires. Vacuum model contains only two states as well.

Fig. 4 (a) shows the simple NCES models for vacuum and cylinder modules. *Vacuum* module has two inputs ‘vacuum_on’ and ‘vacuum_off’ and two outputs: ‘vacState’ and event ‘CHG’. Two places ‘On’ and ‘Off’ represent simple vacuum states and transitions between this states emit event ‘CHG’ when fire. *Cylinder* model has two inputs and two states as well; transitions with inhibitor arcs provide operation regard to rules:

$$ToHome.enabled = fwd \& \overline{bwd};$$

$$ToEnd.enabled = \overline{fwd} \& bwd;$$

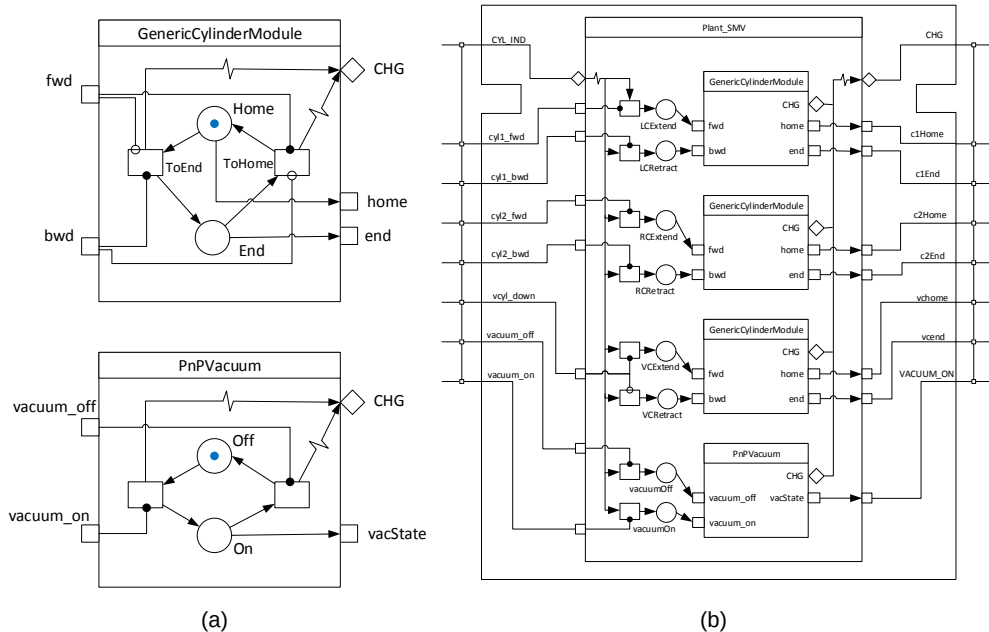


Fig. 4. Vacuum and cylinder NCES-models (a), plant model with FB wrapper (b)

Fig. 4 (b) shows plant model inside FB wrapper interface. Inputs of NCES cylinders and vacuum are connected to first-line places and outputs are merged with wrapper's outputs (in this case, event signal buffers reset is managed by wrapper and "outer" SMV module, regard to FB operational semantics). Table 1 shows verification time and memory usage for verification of full and reduced pick-and-place manipulator model.

Table 1

Model checking in Cadence SMV

Model	Time	RAM
Full PnP model	–	> 2GB*
Reduced PnP model	294 s	1082 MB

* On Win32 Cadence SMV running under Windows 7 x64, reaching of 2GB memory limit cause program failure, on Windows x32 versions, verification process stuck for hours without any results.

4. Model verification and improvements

For verification, two blocks of simple *CTL* formulas was used. The first block contains properties for the plant transitions. Some related *CTL* formulas are presented below:

```

AG ((MODEL.Cylinder_LCExtend&MODEL.Cylinder_LCAtHome)
->AFMODEL.Cylinder_LCAtEnd)
AG ((MODEL.Cylinder_LCRetract&MODEL.Cylinder_LCAtEnd)
->AFMODEL.Cylinder_LCAtHome)
AG ((MODEL.Cylinder_LCRetract&MODEL.Cylinder_LCAtEnd)
->AFMODEL.Cylinder_LC_CHG)
AG ((MODEL.Cylinder_LCExtend&MODEL.Cylinder_LCAtHome)
->AFMODEL.Cylinder_LC_CHG)
AG ((MODEL.Cylinder_RCExtend&MODEL.Cylinder_RCAtHome)
->AFMODEL.Cylinder_RCAtEnd)

```

The second block checks reachability of *WPMaster* function block *EC*-states. Some related *CTL* formulas are as follows:

```

EFCONTROL.Master.Q_smv=START
EFCONTROL.Master.Q_smv=S_INIT
EFCONTROL.Master.Q_smv=Pick1
EFCONTROL.Master.Q_smv=VCDown
EFCONTROL.Master.Q_smv=VCUp

```

In the above *CTL* formulas, **AG**, **AF**, and **EF** are temporal connectives [1].

Checking of properties on this example with the reduced plant model and counter-example analysis has shown several problems in the checked model.

Dispatchers in composite functional blocks operate with the following SMV rules:

```

next( alp ):=case
& omega & !ExistsInputEvent: TRUE;
TRUE : alp;
esac;

next( ):=case

```

```

    & omega : FALSE;
    TRUE :      ;
esac;

```

where ai is a “start execution” signal from dispatcher to i -th component FB instance, t is a signal from the instance to dispatcher which reports the end of i -th component instance execution.

If any input event ($ExistsInputEvent = TRUE$) exists, then signal i will be cleared in the next step, but signal $alpi$ would not be set, in this case signal i will be “missed” and execution of composite functional block will stop.

The problem can be solved with changing the $alpi$ setting rule by removing “!ExistsInputEvent”

```

next(  $alpi$  ):=case
    & omega & !ExistsInputEvent: TRUE;
    TRUE :  $alp$ ;
esac;

```

But, if condition $!ExistsInputEvent$ was removed from the $alpi$ setting rules, there is no guarantee that all input events would be processed, and input events that occur during the execution would be processed only in the next execution cycle of upper-level composite FB (one shall note that this problem would not occur in sequential execution semantics). As an alternative, dispatcher can give control to the upper level (by setting $beta$ signal) if and only if there is no “active” input events (events, that can be processed), otherwise execution of this function block runs in cycle until all input events would be processed.

In the manipulator model, *MasterSlave* composite FB is a top-level block (plant model is asynchronous) and its $beta$ and $alpha$ signals are connected regard to second option.

The second problem is related to event occurrence in cyclic execution models. In the existing model, made by converter [11, 12], *DeIEI3* [4] signal reset discipline is used (only processing input event signal buffer clears and other buffers save). For the function block *WPMaster* (

Fig. 5, the basic priority of input events is $INIT > REQ > LC\_CONF > RC\_CONF > VC\_CONF$. And in case of simultaneously occurrence of events REQ , LC_CONF and RC_CONF , the execution sticks as the higher priority signal REQ blocks LC_CONF and RC_CONF signals and it will never continue because there is no enabled transition from state *Pick3* to process (and clear) event REQ . (“simultaneously” for given FB in case of cyclic execution means these events occur after the end of previous execution of this basic FB and before the next execution (after signal $master\_beta$ was set by instance $master:WPMaster$ and before the signal $master\_alpha$ was set by scheduler)).

In this case, using of *DeIEI1* signal reset discipline (process only one event signal and clear another) with lowering the priority of REQ event below LC_CONF and RC_CONF priorities, can partially solve the problem.

Regarding the cyclic execution model, components of block *MasterSlave* executing in order: *Vaccum - FB5 - LC - RC - VC - Master - SpringCyl*.

In this case, with the simultaneously (during one execution cycle of composite FB *MasterSlave*) occurrence of events from the plant, component FB instances *LC* and *RC*

(type: *SlaveCylinder*) will consequentially output events *LC_CONF* and *RC_CONF* before execution of *Master:WPmaster*. In this case, only one of these events will be processed and another will be cleared.

Another solution is to use *DeIEI3* signal reset discipline, but select the highest priority event only among “active” event signals (which allow the transition from the current *EC*-state)

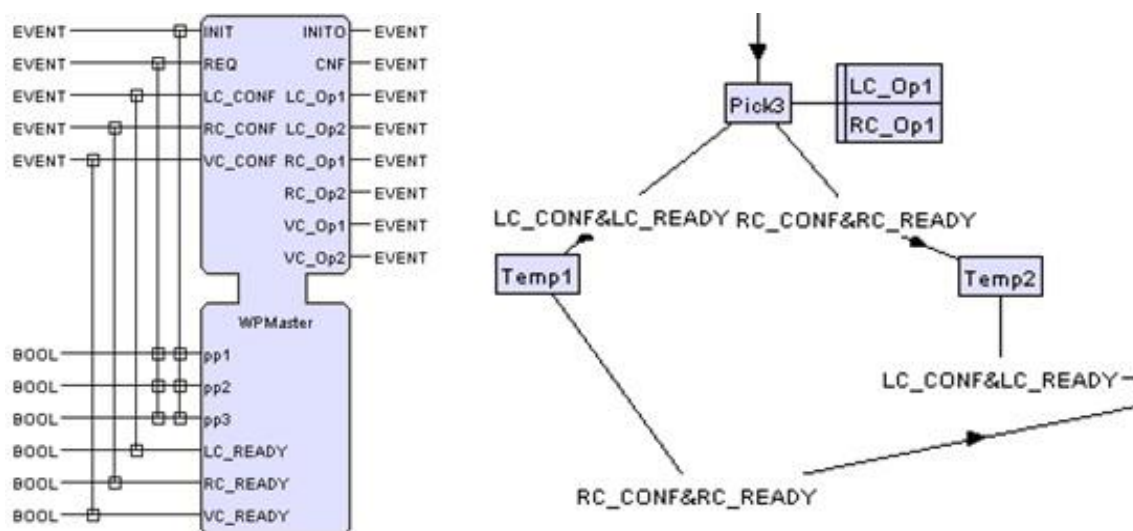


Fig. 5. Block WPMaster and fragment of its state machine

For the *PnP* model, *DeIEI1* reset discipline was used together with modified instance execution order *Vaccum – FB5 – LC – Master – RC – VC – SpringCyl*, so the events from *LC* and *RC* blocks come in different execution cycles.

5. Conclusion

We have presented a useful approach to model closed-loop automation systems with IEC 61499 function block control applications. The plant reduction proposed in this paper can be helpful when a full model, based on FB operational semantics, is complicated and its verification takes more resources, than we can provide. The future work can be directed to provide FB models with execution time and prove possible performance gain of *speculative* computation [13] usage in FB algorithms.

Acknowledgments. This work was partially supported by the Federal Targeted Programme "Research and development in priority areas of elaboration of STC of Russia for 2014-2020" (agreement of 19.06.2014 № 14.574.21.0045) and Luleå University of Technology grant 381119.

References

1. Clarke, E. M. Model checking / E. M. Clarke, O. Grumberg, D. A. Peled. – Cambridge : The MIT Press, 1999.
2. Дубинин, В. Н. Операционная семантика синхронных функциональных блоков IEC 61499 на основе машин абстрактных состояний. Ч. 1: Модель диспетчеров / В. Н. Дубинин, В. В. Вяткин // XXI век: итоги прошлого и проблемы настоящего Плюс. – 2012. – Вып. 4. – С. 233–240.

3. Дубинин, В. Н. Операционная семантика синхронных функциональных блоков ИЕС 61499 на основе машин абстрактных состояний. Ч. 2: Модели блоков и представление в SMV / В. Н. Дубинин, В. В. Вяткин // Современные информационные технологии : тр. Междунар. науч.-техн. конф. – Пенза, 2011. – Вып. 14 – С. 94–100.
4. Дубинин, В. Н. Модели функциональных блоков ИЕС 61499, их проверка и трансформации в проектировании распределенных систем управления : моногр. / В. Н. Дубинин, В. В. Вяткин ; под ред. проф. Н. П. Вашкевича. – Пенза : Изд-во ПГУ, 2012. – 350 с.
5. Rausch, M. Net condition/event systems with multiple condition outputs, in Emerging Technologies and Factory Automation (ETFA) / M. Rausch, H.-M. Hanisch. – Paris, 1995.
6. Vyatkin, V. Object-oriented modular place/transition formalism for systematic modeling and validation of industrial automation systems / V. Vyatkin, H.-M. Hanisch, T. Pfeiffer // Industrial Informatics. – 2003.
7. Patil, S. Closed-loop formal verification framework with non-determinism, configurable by meta-modelling / S. Patil, S. Bhadra and V. Vyatkin // IECON 2011 : 37th Annual Conference on IEEE Industrial Electronics Society. – 2011.
8. Patil, S. Formal verification of Intelligent Mechatronic Systems with decentralized control logic / S. Patil, V. Vyatkin, M. Sorouri // Emerging Technologies & Factory Automation (ETFA), 2012.
9. Wimmel, G. A BDD-based Model Checker for the PEP Tool / G. Wimmel // Major individual project. – University of Newcastle, 1997.
10. Patil, S. Neutralizing Semantic Ambiguities of Function Block Architecture by Modeling with ASM / S. Patil, V. Dubinin, C. Pang, and V. Vyatkin // 9th International Andrei Ershov Memorial Conference, PSI 2014. – Peterhof, St. Petersburg, Russia, 2014.
11. Дубинин, В. Н. Автоматическая трансляция систем функциональных блоков ИЕС 61499 в код SMV / В. Н. Дубинин, Д. Н. Дроздов, Н. П. Вашкевич // Проблемы техники и технологий телекоммуникаций : сб. тр. XIV Междунар. науч.-техн. конф. (ПТиТТ-2013). – Самара, 2013. – С. 259–261.
12. Свидетельство об официальной регистрации программы для ЭВМ № 2013661891. Транслятор XML-описания функциональных блоков стандарта ИЕС 61499 в код SMV / В. Н. Дубинин, Д. Н. Дроздов. – Заявка № 2013660001 ; Зарег. в реестре программ для ЭВМ 18.12.2013.
13. Дубинин, В. Н. Опережающие вычисления в системах функциональных блоков ИЕС 61499 / В. Н. Дубинин, В. В. Вяткин, Д. Н. Дроздов // Современные информационные технологии : тр. Междунар. науч.-техн. конф. – Пенза, 2013. – Вып. 17. – С. 9–24.

Ю. Н. Косников

Россия, Пенза, Пензенский государственный университет

РЕКОНСТРУКЦИЯ ПРОСТРАНСТВЕННОГО РЕЛЬЕФА НА ОСНОВЕ ЦВЕТОВОГО КОДИРОВАНИЯ ИЗОПОВЕРХНОСТЕЙ

Предлагается технология восстановления трехмерного рельефа по данным раскраски географической карты. Цветам карты сопоставляются высоты опорных точек в пространстве отображаемой физической величины, сглаживание рельефа осуществляется методом интерполяции с учетом топологической тенденции рельефа.

Решение многих прикладных задач требует представлять на экране компьютерного монитора изображение, которое должно нести информацию о пространственном рельефе, но, в то же время, может не быть фотореалистичным. Например, визуальная оценка рельефа местности нужна для принятия решений о проектировании дорог, трубопроводов, мостов. В научной визуализации трехмерное представление различных параметров, например, плотности населения, структуры почв, уровне загрязненности воздуха, облегчает восприятие и анализ ситуации в пределах региона.

Трудоемкость создания модели реального рельефа вручную средствами программ-моделеров (3DSMAX, Blender, AutoCADMap 3D и т.п.) весьма высока. Существуют специализированные программы – генераторы ландшафтов, такие как Terragen, Genesis, Bryce, инструменты для рельефного представления данных есть в геоинформационных системах (ArcView, MapInfo и др.). И те, и другие строят рельеф по карте высот, которая создается в интерактивном режиме оператором или импортируется из программ-моделеров, т.е. высокая трудоемкость создания рельефа сохраняется. Зачастую не могут помочь и специализированные программы. Например, известный продукт корпорации Google – модель планеты Земля GoogleEarth – создан на основе спутниковых снимков и не позволяет детально рассмотреть рельеф выбранного участка с различных сторон или с небольшой высоты.

В то же время существует общедоступная информация о рельефе земной поверхности и распределении на ней различных физических величин. Это географические карты, широко представленные в сети Internet, и планы местности, выполненные силами топографов, экологов, землеустроителей. Предлагается технология геометрического моделирования пространственного рельефа на основе использования раскраски географических карт и планов. Решается задача реконструкции пространственного (3D) рельефа по его 2,5D-представлению, т.е. плоскому представлению, содержащему информацию о координате высоты в косвенной форме, а именно, – в форме цвета.

Отличительной особенностью цветового кодирования физических величин на картах является малая глубина цвета: палитра карты насчитывает от 6 до 18 цветов. В связи с этим одному цвету на карте соответствует некоторый диапазон значений представляемой величины, например, высот земного рельефа. Детальная информация о рельефе при изготовлении карт оказывается утерянной, и раскраска карты, по сути, представляет собой набор изоповерхностей, расположенных в пространстве отображаемой величины с некоторым шагом. Другими словами, рельеф представляется набором дискретных уровней – своеобразных уступов, или террас. При такой исходной информации точное восстановление рельефа становится не-

возможным, и можно говорить лишь о реконструкции с некоторой погрешностью. Для этого предлагается использовать интерполяцию, за узлы которой приняты опорные точки, расставленные на карте регулярно или по границам цветовых зон. Программная реализация второго варианта сложна алгоритмически, а ручная расстановка опорных точек требует больших трудозатрат.

Регулярная расстановка опорных точек сводится к вычислению их координат на плоскости карты с заданным шагом и считыванию значения цвета (кода R-G-B), принадлежащего каждой полученной точке. Каждому цвету ставится в соответствие среднее значение диапазона высот, соотнесенного на карте с этим цветом. В отличие от опорных точек карты (на плоскости) точки, получившие третью координату, назовем опорными точками рельефа. На конечном этапе геометрического моделирования опорные точки рельефа становятся вершинами полигональной модели рельефа, передаваемой для отображения в графический процессор видеокарты компьютера. В связи с этим шаг расстановки опорных точек выбирается из условий эргономичного отображения и зависит от масштаба карты и предполагаемого расстояния наблюдения рельефа.

Опорные точки располагаются с одинаковой плотностью на всех участках карты: и на тех, где высота рельефа меняется с большой скоростью, и на тех, где она не меняется вообще. На участках второго вида полигональная интерполяция по опорным точкам рельефа даст уже упоминавшиеся террасы, ухудшающие визуальное восприятие рельефа. Такие террасы дадут и иные виды интерполяции в силу компланарности опорных точек рельефа, соответствующих одноцветным зонам карты. Для устранения террас предлагается ввести влияние на расположение вершин будущей полигональной модели рельефа более дальней окрестности, т.е. учесть топологическую тенденцию моделируемого участка поверхности.

На образ карты накладывается квадратное окно, включающее девять зон. Центральная зона подлежит полигонизации, окружающие восемь зон задают влияние на нее рельефа окрестности. Размер зоны определяется характером рельефа: для пересеченного рельефа он может быть меньше, для равнинного – больше. Внутри каждой зоны попадает некоторое количество опорных точек карты, к каждой из которых «привязан» свой цвет.

Внутри центральной зоны организуется перебор опорных точек и подсчитывается количество перемен цвета. Если оно больше заданного числа (зависит от характера рельефа), то без изменений опорных точек рельефа осуществляется переход к следующему положению окна. Оно перемещается на расстояние, равное стороне зоны. Тем самым опорные точки рельефа для предыдущего положения центральной зоны оказываются подготовленными к полигонизации, они сохраняют свои прежние высоты. Если число перемен цвета ниже заданного порога, размеры всех зон окна увеличиваются, например, в два раза по каждой координате, а число опорных точек, попавших в внутрь зон, редуцируется (тоже в два раза по каждой координате). Опорные точки редуцированного набора назовем новыми опорными точками. Для прореженных (новых) опорных точек центральной зоны снова рассчитывается и сравнивается с порогом число перемен цвета. Если это число по-прежнему мало, производится дальнейшее увеличение размеров всех зон окна и редукция числа попавших в них опорных точек. Если же число перемен цвета превышает пороговое значение, производится коррекция исходных (имевшихся до редуцирования) опорных точек рельефа, попавших в центральную зону. Новые опорные точки, благодаря особенностям процедуры интерполяции, сохраняют свои прежние высоты, а исходные опорные точки, не совпадающие с новыми, считаются текущими, и для них находятся скорректированные высоты.

Составляется интерполяционное выражение, например, на основе радиальных базисных функций (РБФ). Для рельефа, привязанного к земной поверхности, это однозначная функция геодезических координат, а для ограниченных участков рельефа – функция декартовых координат. Благодаря однозначности, интерполянт записывается в явной форме, и степень влияния РБФ каждой i -й новой опорной точки на текущую точку определяется декартовым расстоянием r_i между исходной и новыми опорными точками в плоскости карты [1]:

$$z(x, y) = \sum_{i=1}^N \lambda_i \varphi(r_i) \quad (1)$$

где x, y, z – координаты текущей точки рельефа; φ – РБФ, аргументом которой является расстояние r_i ; λ_i – весовой коэффициент i -й опорной точки, который характеризует степень влияния этой точки на текущую точку; N – число новых опорных точек окна.

Коэффициенты λ_i находятся из условия прохождения рельефа через новые опорные точки рельефа окна. После нахождения λ_i по выражению (1) находятся скорректированные высоты в исходных опорных точках рельефа центральной зоны. По окончании сканирования карты окном на скорректированных опорных точках рельефа строится полигональная сетка.

Следует отметить, что для крайних участков карты процесс коррекции высот опорных точек может протекать аналогично описанному. Различие лишь в том, что центральная зона окна будет вплотную примыкать к краям карты и количество окружающих зон, опорные точки которых будут влиять на текущие точки центральной зоны, будет уменьшено.

Предлагаемая технология характеризуется алгоритмической простотой, т.к. на каждом этапе коррекции опорных точек оперирует одним и тем же их количеством. Гладкое сопряжение участков рельефа, формируемых при каждом отдельном положении окна, достигается учетом влияния на рельеф в центральной зоне окна высот опорных точек, расположенных в остальных зонах. Изобразительные возможности описанной технологии иллюстрирует рис. 1, на котором показан фрагмент физической географической карты с удаленной координатной сеткой (рис.1,а) и восстановленный по этой карте рельеф (рис. 1,б).

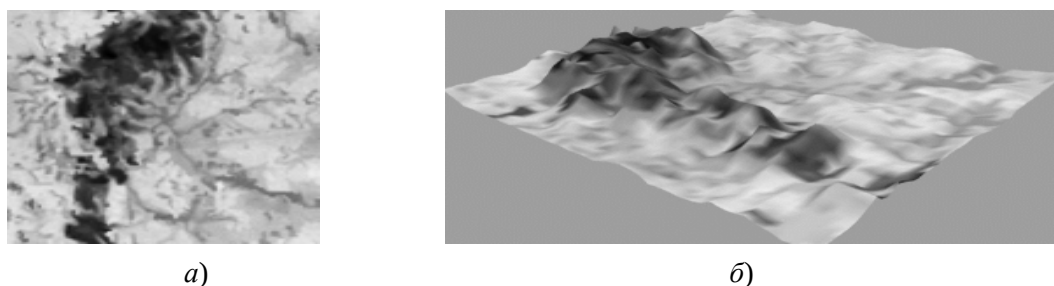


Рис. 1. Фрагмент географической карты (а) и реконструкция рельефа местности по этому фрагменту (б)

Анализ показывает возможность практического применения полученной трехмерной модели для выполнения задач, не требующих фотореалистического представления рельефа. Улучшение восприятия изображения достигается моделированием освещения рельефа и наложением реалистичной текстуры средствами компьютерной графики.

Список литературы

1. Косников, Ю. Применение радиальных базисных функций в научной визуализации / Ю. Косников // Научная визуализация. – 2013. – Т. 5, № 1. – С. 38–47. – URL: <http://sv-journal.com/index.php?lang=ru>

М. А. Тараник, Г. Д. Копаница

**Россия, Томск, Национальный исследовательский
Томский политехнический университет**

НЕОБХОДИМОСТЬ ФОРМИРОВАНИЯ НОВЫХ ИНТЕЛЛЕКТУАЛЬНЫХ ПОДХОДОВ ДЛЯ РЕШЕНИЯ УПРАВЛЕНЧЕСКИХ ЗАДАЧ В ЗДРАВООХРАНЕНИИ

Рассматриваются общие вопросы информатизации здравоохранения, описываются используемые технологии, а также их прикладные задачи. Среди технологий особое внимание уделяется интеллектуальному подходу. Описаны преимущества и недостатки его технологий. Обоснована необходимость разработки новых подходов в данном направлении.

Здравоохранение является одной из ключевых отраслей государственной деятельности, основной задачей которой является сохранение и повышение уровня здоровья населения. Совокупность объектов здравоохранения характеризуется множеством медицинских организаций, эффективная автоматизированная работа которых во многом обеспечивается за счет применения информационных технологий, позволяя решать необходимый круг клинических задач.

Среди таких подходов широкое распространение получили медицинские информационные системы (МИС). На данный момент МИС эффективно решают задачи, связанные с автоматизацией деятельности медицинских организаций: обеспечение сбора, хранения, поиска и анализа медицинских данных, организация единого информационного пространства за счет движений информационных потоков как внутри одного, так и между различными медицинскими организациями, формирование необходимой медицинской отчетности. В результате использование таких систем снижается число врачебных ошибок, уменьшается риск потери информации, оптимизируются расходы на лечение, сокращается время документооборота. Однако данный автоматизированный подход, ограниченный сбором, хранением и передачей информации, имеет низкоэффективную интеллектуальную, а также управляющую составляющую, что затрудняет эффективно решать задачи, требующие принятия решений, в частности, связанные с оказанием эффективной информационной поддержки медицинских специалистов при диагностировании заболеваний и назначении эффективного плана лечения. Более того, это не позволяет производить прогнозирование развития медицинской организации, а также принимать необходимые управленческие решения.

Следующим поколением МИС, решающих задачи интеллектуальной обработки медицинских данных, стали информационные системы, имеющие компо-

нент искусственного интеллекта. Обоснование использования интеллектуальных составляющих для МИС является целый ряд факторов. Среди данных факторов основным является сложность формализации клинической предметной области, в связи с неточностью, нечеткостью, а также неполнотой медицинских знаний. Более того, в здравоохранении важную роль играет ранее накопленный опыт экспертов, активное использование которого вносит весомый вклад в эффективное решение актуальных клинических задач диагностирования, прогнозирования, классификации, управления, обучения, оптимизации и т.д. Такие медицинские интеллектуальные информационные системы (МИИС) позволяют решать задачи более высокого уровня, обеспечивая эффективный процесс как принятия врачебных, так и управленческих решений в лечебно-профилактических учреждениях (ЛПУ).

Использование интеллектуальных систем в медицине также подкреплено солидным опытом, доказательством этому является большое количество опубликованных разработок за недавний период. Среди алгоритмов МИИС, обеспечивающих логический вывод можно выделить такие технологии как байесовские сети (BayesianNetworks), нечеткая логика (Fuzzylogic), метод опорных векторов (SupportVectorMachines), метод k-ближайших соседей (k-NearestNeighbor), метод Мамдани, технология DataMining(рисуюнок 1).



Рис. 1. Технологии логического вывода интеллектуальных систем

Однако среди представленных выше методов существуют значительные недостатки, так например, метод опорных вектор ограничен областью применения, так как позволяет решать задачи бинарной классификации, а алгоритм k-NNне детерминирует важность, вес и также мощность связи различных узлов.

В качестве способа разрешения недостатка алгоритма k-NNбыло предложено использование нечеткой логики, что позволило модифицировать классический метод с целью повышения достоверности результатов обработки при использовании слабоформализованных данных. Таким образом, синтез различных методов и технологий повышает эффективность обработки данных, а также достоверность результатов логического вывода МИИС. Однако, как показывают исследования, во многих работах отмечаются недостатки представленных методов логического вывода, но не присутствуют способы их разрешения при разработке медицинской ИИС, в особенности, когда система должна обеспечивать необходимый логический вывод для решения задач связанных с управлением, а также принятием решений.

Таким образом, несмотря на наличие моделей и средств, необходимость формирования новых подходов и технологий для организации интеллектуальных медицинских информационных систем, обеспечивающих эффективное решение управленческих задач представляет, как научный, так и практический интерес.

Е. А. Николаева, Г. И. Советкина, А. В. Дубравин

Россия, Пенза, Пензенский государственный университет

РАЗРАБОТКА ПРИЛОЖЕНИЯ ИМПОРТА ДАННЫХ

Представлен способ организации импорта данных из табличного процессора MS Excel в xml файл средствами языка C# и платформы .NetFramework

В связи с возрастающей сложностью решаемых задач и совершенствованием исполнительного механизма устройств увеличивается использование датчиков со сложной обработкой сигналов и возможностями настройки и регулирования параметров и стандартным интерфейсом системы управления.

В процессе работы в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития разработки научно-технологического комплекса России на 2014–2020» (соглашение № 14.574.21.0045 от 06.19.2014), была поставлена задача создания уникального универсального каталога цифровых и аналоговых датчиков.

Для решения данной задачи была создана база знаний, содержащая описание устройств и их параметров. Для первоначального накопления информации о приборах и устройствах был использован табличный процессор MS Excel, поскольку представление информации в табличном виде облегчает ввод и визуальный анализ информации.

Одной из сложных проблем всегда была и остается проблема обмена данными между различными информационными системами. Для решения этой проблемы часто применяют язык разметки XML.

Структура XML-файла была разработана исходя из табличного представления информации об устройствах в MS Excel. Схематично структура представлена ниже на рис. 1, 2.

XML-файл содержит корневой элемент Devices, включающий информацию о всех устройствах каталога. Каждый элемент Device, содержащий информацию о единственном устройстве, является потомком элемента Devices и включает в себя дочерние элементы:

- Model – модель устройства;
- Mod – модификация устройства;
- Name – наименование устройства;
- Manufacturer – производитель устройства;
- Url – ссылка на источник информации;
- Interfaces – интерфейсы устройства;
- Group – группа;
- Subgroup – подгруппа;
- Subgroup2 – субподгруппа;
- Params – параметры устройства;
- ConnectionsWith – устройства сопряжения;
- Comment – комментарий относительно работы устройства в целом.

При этом элементы Interfaces, Connections With имеют собственные дочерние элементы. Элемент Params является родителем для элементов Parameter, каждый из которых имеет следующую структуру (см. рис. 2):

- Name – наименование параметра;
- Value – значение параметра;
- Units – единица измерения параметра;
- Max – максимально допустимое значение параметра;
- Min – минимально допустимое значение параметра;
- Type – тип параметра;
- InOutType – тип контакта;
- Comment – комментарий.

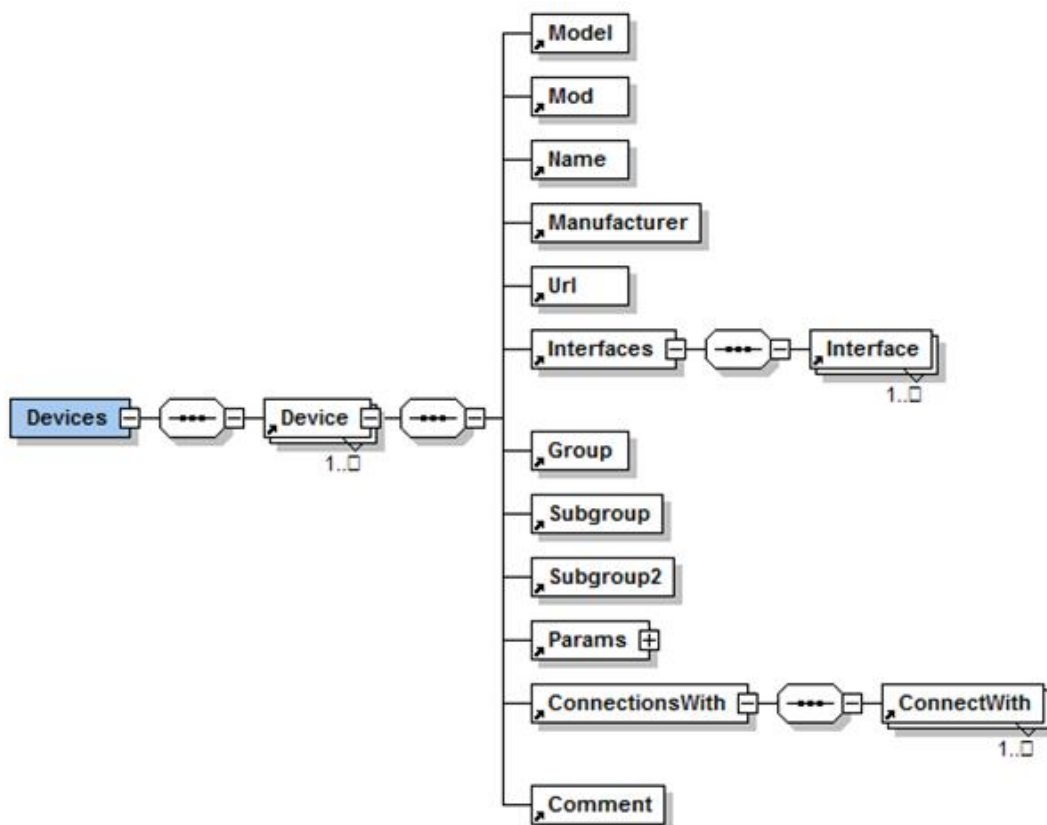


Рис. 1. Схематичное представление структуры XML-файла

Для того чтобы внести данные в XML-файл необходимо их извлечение из электронной таблицы MS Excel. Следовательно, стоит задача разработки программы-конвертера, которая бы выполняла преобразование данных из файла с расширением .xlsx в файл с расширением .xml. В качестве среды разработки такого приложения был выбран программный продукт Microsoft Visual Studio 2012, а в качестве языка программирования объектно-ориентированный язык C#.

Для хранения информации об отдельном устройстве был написан класс Plain Record, включающий в себя переменные с именами, аналогичными именам потомков элемента Device файла XML. Поля класса Interface, Connect With реализованы в виде списков, а поле Params – в виде списка классов типа Parameter, каждый из которых аналогичен структуре элемента Parameter файла XML.

Для корректного формирования списка устройств был создан класс trp, представляющий точную копию вышеописанного класса Plain Record, за исключением полей, реализованных списком. Потребность в данной организации данных возникла в связи со множественностью некоторых полей табличных данных и их последующей реализации в виде списков.

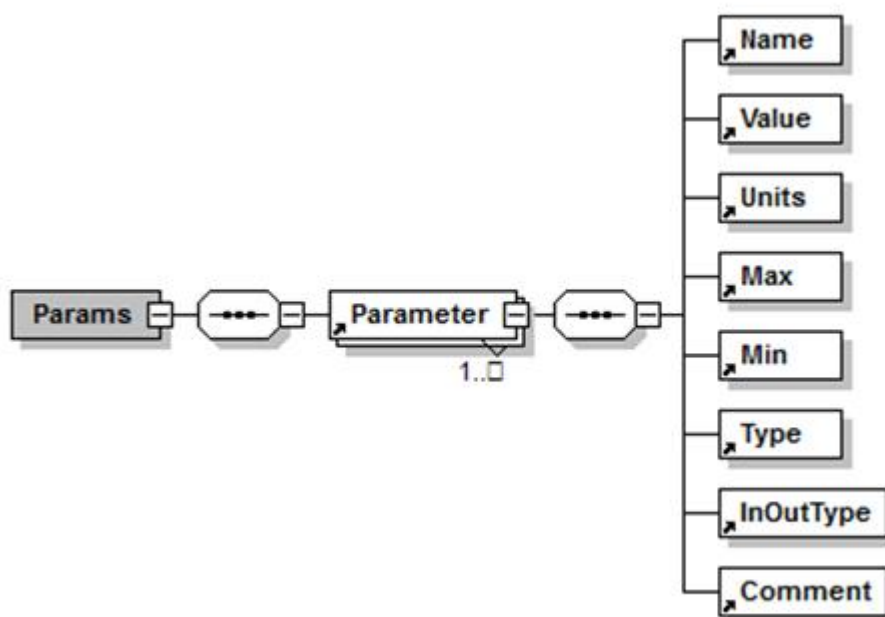


Рис. 2. Схематичное представление структуры элемента Params

Непосредственный алгоритм чтения данных из файла .xlsx реализован с помощью классов Xls Scanner и Data Parser.

С помощью методов класса XlsScanner выполняется открытие книги MSExcel, занесение используемых ячеек таблицы Excel в массив data, закрытие книги. Из массива data значение каждой ячейки присваивается переменной строкового типа.

Методы класса Data Parser используют функции вышеописанного класса для создания элемента типа Plain Record и последующего добавления его в список устройств. Собственно в классе Program реализуются главный цикл программы, где с помощью написанных методов осуществляется чтение данных из файла Excel с записью их в переменные классов.

С целью построения XML-дерева в программе использован набор функций LINQ to XML. В класс Plain Record был добавлен метод, осуществляющий создание XML-элементов и занесение в них полей данного класса в соответствии с разработанной выше структурой XML-файла. Данный метод использует функцию класса Parameter, реализующую создание элемента Parameter и его потомков. В классе Program выполняется создание корневого элемента файла, а также запись в него созданных с помощью написанных функций XML-элементов.

Разработанное приложение позволяет многократно сократить затраты временных и человеческих ресурсов по сравнению с непосредственно ручной обработкой собранной информации и обменом данными между несовместимыми системами.

Список литературы

1. Рихтер, Д. CL Rvia C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Д. Рихтер. – СПб. : Питер, 2012. – 986 с.
2. C# для начинающих. – URL: <http://postgresql.leopard.in.ua> (дата обращения: 08.11.2014).

В. А. Силич, М. П. Силич, С. В. Аксенов

Россия, Томск, Национальный исследовательский
Томский политехнический университет,
Томский университет систем управления и радиоэлектроники

КОСВЕННАЯ ОЦЕНКА ФАКТОРОВ ЭНЕРГЕТИЧЕСКОЙ ЭФФЕКТИВНОСТИ ТЕРРИТОРИАЛЬНЫХ ОБРАЗОВАНИЙ

Предлагается метод вычисления оценок факторов, определяющих энергетическую эффективность территориальных образований. Взаимовлияние факторов представляется в виде когнитивной карты. Метод предполагает представление оценок в виде нечетких чисел.

Целью анализа энергетической эффективности территориальных образований (ТО), в качестве которых могут выступать субъекты РФ или муниципальные образования, как правило, является определение уровня каждого ТО относительно других (в виде некоторой интегральной оценки или рейтинга) и выявление причин, обусловивших достижение этого уровня. Для того чтобы проанализировать опосредованное влияние множества факторов, целесообразно использовать когнитивные карты. С их помощью исследуют, как изменение одних факторов влияет на изменение других (импульсное моделирование). Разрабатываемый авторами подход [1] существенно расширяет возможности когнитивного анализа, так как позволяет наряду с оценкой влияния различных факторов на энергоэффективность получать нечеткие оценки текущего состояния факторов и прогнозные оценки. Вопрос несопоставимости условий в различных территориальных образованиях решается путем нечеткой кластеризации ТО в группы с сопоставимыми условиями. Оценки факторов определяются как на основе непосредственной интерпретации значений индикаторов по функциям принадлежности, так и косвенным образом на основе оценок других факторов. Для косвенной оценки факторов разработан метод построения системы нечеткого логического вывода типа Мамдани [2], автоматически генерирующий правила вывода на основе обучающей выборки. Однако процедура формирования обучающей выборки довольно трудоемка, а качество выборки сильно зависит от квалификации эксперта. Альтернативным способом косвенной оценки факторов является использование операций с нечеткими числами. В данной работе предлагается метод косвенной оценки факторов энергоэффективности, предполагающий представление оценок входных факторов в виде нечетких чисел (L - R)-типа, выполнение вычислений и интерпретацию вычисленных значений.

Разработанный авторами метод построения иерархических гибридных когнитивных карт предполагает выделение двух типов концептов – факторов и индикаторов.

Под фактором понимается некоторое свойство системы, оцениваемое на качественном уровне, например: «Уровень потребления тепловой энергии в жилищном комплексе»; «Степень оснащенности приборами учета», «Суровость климатических условий». Индикаторы представляют собой измеримые показатели, в той или иной степени характеризующие факторы. Индикатор сопоставляется фактору с помощью отношения ассоциации, которому приписывается коэффициент соответствия (число от 0 до 1). Чем выше значение коэффициента, тем точнее индикатор характеризует фактор. Между факторами устанавливаются отношения влияния. Сила влияния оценивается числом от -1 до 1 , причем эта величина может

быть переменной, принимающей разные значения для разных территориальных образований. К любому из факторов может быть «прикреплена» дочерняя когнитивная карта. На рис. 1 приведен пример когнитивной карты.



Рис. 1. Когнитивная карта

Рассмотрим ситуацию, когда несколько входных факторов $f_1, f_2, \dots, f_n$, связанных с индикаторами, влияют на выходной фактор f_0 , не имеющий индикаторов. Например, на карте, представленной на рис. 1, на фактор «Мотивация к экономии воды», не связанный с индикаторами, влияют три фактора («Учет воды», «Существенность расходов на воду», «Получение субсидий»), которым сопоставлены индикаторы. Силу влияния фактора f_i на f_0 обозначим через w_{i0} , степень соответствия индикатора d_j фактору f_i – через v_{ij} . Исходными данными являются значения индикаторов, соответствующих входным факторам, для оцениваемого территориального образования.

Сначала необходимо преобразовать векторы значений индикаторов, соответствующих входным факторам, в скалярную числовую оценку. Значения каждого индикатора нормируются. В случае, когда чем больше значение индикатора, тем выше должна быть оценка, используется формула: $d_j^* = \frac{d_j - d_{\min}}{d_{\max} - d_{\min}}$, где $d_{\min}$, $d_{\max}$ – соответственно минимальное и максимальное значение индикатора по всем сравниваемым территориальным образованиям. В случае, когда чем меньше значение индикатора, тем выше должна быть оценка, используется формула:

$$a_i = \frac{d_{\max} - d_i}{d_{\max} - d_{\min}}.$$

Интегральную оценку можно получить на основе свертки нормированных значений индикаторов. В качестве весовых коэффициентов логично использовать степени соответствия индикаторов фактору, нормированные по формуле:

$v_{ij}^* = v_{ij} / \sum_{j=1}^m v_{ij}$. При использовании аддитивной свертки числовое значение оцен-

ки фактора определяется следующим образом: $a_i = \sum_{j=1}^m v_{ij}^* d_j^*$. Степень соответствия

вычисленной оценки фактору определяется по формуле: $v_i = \sum_{j=1}^m v_{ij}^2 / \sum_{j=1}^m v_{ij}$.

Очевидно, что если степень соответствия не равна единице, то полученная числовая оценка является приблизительной. Чем ниже значения v_i , тем более «размытой» является оценка. Можно преобразовать ее в нечеткое число, коэффициенты нечеткости которого тем выше, чем ниже степень соответствия v_i . Используем функцию принадлежности (L - R)-типа, в частности треугольную, которая задается модальным значением и двумя коэффициентами нечеткости (левым и правым). В качестве модального значения логично использовать оценку a_i . Левый и правый коэффициенты нечеткости определяются по формуле: $\alpha_i = \beta_i = (1 - v_i) / 2$. Если степень соответствия $v_i = 1$, то $\alpha_i = \beta_i = 0$, т.е. мы имеем четкую оценку.

После того, как определены оценки всех входных факторов в виде нечеткого треугольного числа $F_{i\Delta} = \langle a_i, \alpha_i, \beta_i \rangle_{\Delta}$, $i = \overline{1, n}$, необходимо вычислить оценку выходного фактора. Будем считать, что каждый из входных факторов вносит свой «вклад» в оценку выходного пропорционально силе его влияния w_{i0} . В случае если какой-либо фактор оказывает отрицательное влияние, его оценка меняется на противоположную, т.е. модальное значение определяется по формуле: $1 - a_i$.

Для вычисления оценки выходного фактора в виде нечеткого числа можно использовать аддитивную свертку нечетких чисел, сопоставленных входным факторам. Весовые коэффициенты свертки определяются путем нормирования величин силы влияния: $w_{i0}^* = w_{i0} / \sum_{i=1}^n w_{i0}$, $i = \overline{1, n}$. Оценка выходного фактора: $F_{0\Delta} = \sum_{i=1}^n w_{i0}^* F_{i\Delta}$.

При этом вычисление модального значения a_0 и коэффициентов нечеткости α_0, β_0 выполняется по формулам:

$$a_0 = \sum_{i=1}^n w_{i0}^* a_i, \quad \alpha_0 = \sum_{i=1}^n \alpha_i, \quad \beta_0 = \sum_{i=1}^n \beta_i.$$

Схематично процедура получения косвенной оценки изображена на рис. 2.

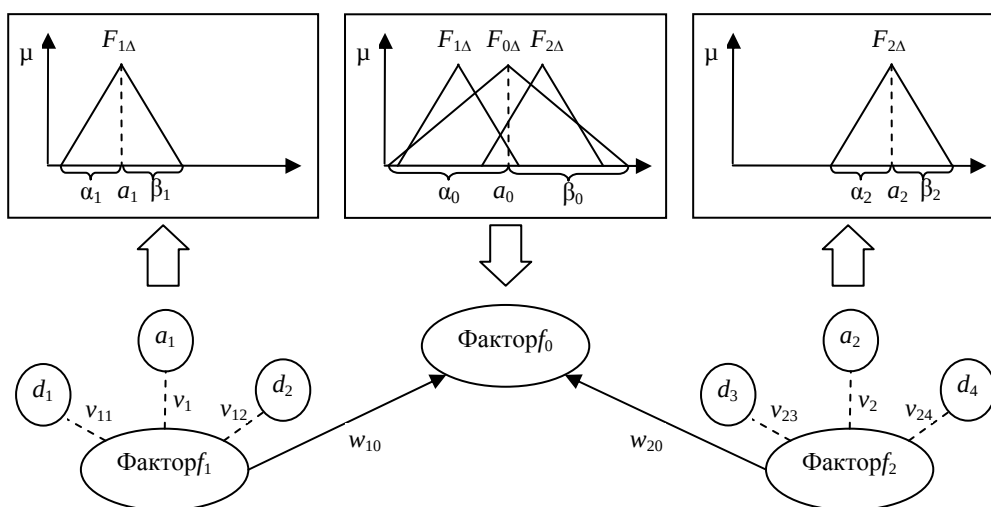


Рис. 2. Схема получения косвенной оценки

Полученную оценку выходного фактора необходимо интерпретировать – преобразовать в лингвистическое значение. Для подбора наименования терма, соответствующего нечеткому числу, можно использовать следующую схему, которая сопоставляет модальному значению a_0 различные термы: $0 < a_0 \leq 0.1$ – «очень низкое», $0.1 < a_0 \leq 0.25$ – «низкое», $0.25 < a_0 \leq 0.4$ – «ниже среднего», $0.4 < a_0 \leq 0.6$ – «среднее», $0.6 < a_0 \leq 0.75$ – «выше среднего», $0.75 < a_0 \leq 0.9$ – «высокое», $0.9 < a_0 \leq 1$ – «очень высокое». Значения коэффициентов нечеткости используются для подбора модификатора, например: $\alpha_0(\beta_0) = 0$ – «точно», $0 < \alpha_0(\beta_0) \leq 0.25$ – «определенно», $0.25 < \alpha_0(\beta_0) \leq 0.5$ – «скорее всего», $\alpha_0(\beta_0) > 0.5$ – «возможно».

Предложенный метод позволяет определять нечеткие оценки факторов на основе оценок других факторов в условиях неопределенности без формирования обучающих выборок.

Работа выполнена при финансовой поддержке гранта Российского фонда фундаментальных исследований № 13-07-00397А.

Список литературы

1. Силич, М. П. Анализ энергетической эффективности территорий на основе иерархии гибридных когнитивных карт / М. П. Силич, В. А. Силич, С. В. Аксенов // Известия Томского политехнического университета. – 2013. – Т. 323, № 5.
2. Силич, В. А. Алгоритм построения нечеткой системы логического вывода Мамдани, основанный на анализе плотности обучающих примеров / В. А. Силич, М. П. Силич, С. В. Аксенов // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2013. – № 3 (29). – С. 76–82.

А. В. Сасева

Россия, Смоленск, Смоленский филиал Московского энергетического института

ИМИТАЦИОННОЕ РЕШЕНИЕ ЗАДАЧИ ИЗГОТОВЛЕНИЯ РЕКЛАМНЫХ СТЕНДОВ

Представлены результаты оптимизации процесса изготовления рекламных стендов на основе имитационного моделирования.

Применение информационных технологий позволяет не только автоматизировать бизнес-процессы на предприятии, но и оптимизировать их структуру. В последнем случае часто применяется метод имитационного моделирования, который позволяет «проигрывать» процесс не только при различных влияющих факторов, но и при различной структуре построения исследуемого процесса. Данный метод был применен для оптимизации производственного процесса изготовления рекламных стендов на ООО «Marketingdrive», занимающийся изготовлением рекламных конструкций, в частности стендов.

Процесс изготовления рекламных стендов является многоэтапным, содержащим как последовательные участки, так и распараллеленные, при чем их комбинация может изменяться. Рассмотрение различных вариантов на основном предприятии практически невозможно, так как это привело бы к частичной оста-

новке производства, а непредсказуемость результата, даже к появлению убытка. В тоже время, ответы на интересующие вопросы по организации работ, параметров процессов может дать имитационная модель.

На первом этапе исследования была проведена временная хронометрия операций на производственном участке, проведена оценка интенсивности заказов, сбоев появления браков, характеристики очередей на наиболее нагруженных участках процесса.

Для отображения в имитационной модели обозначенных аспектов необходимо, чтобы среда разработки модели содержала соответствующие инструментальные средства. Такие средства имеются в известном программном продукте GPSSWorld – популярной среде разработки имитационных моделей [1]. Студенческая версия этой программы распространяется свободно и позволяет строить достаточно сложные модели, содержащие до ста пятидесяти блоков, а применение методов косвенной адресации позволяет (за счет усложнения алгоритма модели) решать даже те задачи, которые при обычном подходе требовали бы больше количество блоков.

В основу концепции разрабатываемой модели положен подход, применяемый в теории систем массового обслуживания, при котором исследуемый процесс рассматривается как совокупность источников заявок, устройств их обслуживания, очередей [2].

После построения модели был проведен полный факторный эксперимент, в котором критерием оптимизации было время прохождения всего цикла создания рекламного стенда. Параметрами оптимизации (переменные факторного пространства) являлись: количество рабочих занятых, параллельно на операции фотокопирования; длительность контроля каждой операции; продолжительность и количество смен. Некоторые операции не подлежали изменениям длительности, так как были строго регламентированы рамками технологического процесса.

В результате многократных прогонов модели с различными комбинациями параметров оптимизации, вызванных в соответствии с планом эксперимента было получено, что оптимальной конфигурацией производственного процесса является такая, при которой на операции фотокопирования задействовано два рабочих; длительность контроля операции составляет три с половиной минуты; режим работы - две смены по восемь часов. При такой комбинации параметрам оптимальное модельное время прохождения заказа было минимально и составило два часа семнадцать минут. Данные рекомендации были приняты менеджером предприятия для возможного дальнейшего применения в рабочем процессе.

Предложенный подход может быть применен в дальнейшем не только для основного бизнес-процесса, но для оптимизации вспомогательных процессов.

Список литературы

1. Советов, Б. Я. Моделирование систем / Б. Я. Советов. – М. : Юрайт, 2013.
2. Боев, В. Д. Моделирование систем инструментальными средствами GPSSWorld / В. Д. Боев. – М. : ВНУ, 2011.

Е. В. Петрунина

Россия, Пенза, Пензенский государственный университет

ГЕТЕРОГЕННАЯ ИНФОРМАЦИОННО-МОДЕЛИРУЮЩАЯ СИСТЕМА СЛОЖНЫХ ЭРГАТИЧЕСКИХ КОМПЛЕКСОВ

Представлены результаты, полученные в ходе исследований, проведенных в области системного анализа сложных техногенных эргатических систем, которые характеризуются как большим количеством внутренних элементов, так и сложностью взаимосвязей между ними. Рассматривается методика применения кластерных технологий при реализации параллельных аппроксимационных алгоритмов и программ в проведении научных исследований и учебных занятий.

Введение

Внешняя и внутренняя среда, в которой функционируют современные сложные техногенные эргатические комплексы (СТЭК), существенно изменилась за последние десятилетия: изменились средства производства и информационные технологии как база других важных технологий (вычислительных, коммуникационных и пр.) [1]. Поэтому происходит пересмотр способов системного анализа и используется принципиально иной подход, наиболее важным направлением которого является реинжиниринг, и который позволит в полной мере реализовать преимущества новых технологий и человеческих ресурсов, а также предотвратить кризисные явления на предприятии.

Большое количество работ за последние десятилетия посвящено изложению и обобщению результатов практического создания завершенной моделирующей среды, ориентированной на использование распределенных и параллельных вычислительных средств как новой парадигмы в области одного из наиболее приоритетных направлений исследований и разработок в системном анализе и компьютерном моделировании с использованием функциональных возможностей наиболее распространенных систем компьютерного моделирования типа Maple, MathCAD, Matlab, Mathematica и т.п.

На современном этапе речь идет уже не просто о тех или иных отдельных компьютерных моделях, а о различных вариантах интегрированного использования различных моделирующих средств решения широкого спектра задач системного анализа.

В качестве перспективного варианта реализации моделирующей среды выбрана разработка сосредоточенных и распределенных развивающихся кластерных систем и их приложений. Накопленный фонд моделей используется и для развития международного научного сотрудничества в области космических технологий [2].

Гетерогенная информационно-моделирующая система (ГИМС) сложных эргатических комплексов рассматривается как взаимосвязанная совокупность концепций, методов, технологий, технических и программных средств, используемых для имитационного моделирования, обработки, хранения и выдачи информации потребителю о свойствах исследуемого объекта. Современное понимание ГИМС предполагает использование вычислительной техники и электронных средства коммуникаций в качестве основного технического средства переработки информации. Глобальная сеть Internet сделала реальностью открытость и доступность в реальном масштабе времени информационных ресурсов самого разного объема и

содержания, а использование сервиса облачных хранилищ данных (GoogleDrive и Яндекс Диск) обеспечило совместную работу территориально распределенных пользователей.

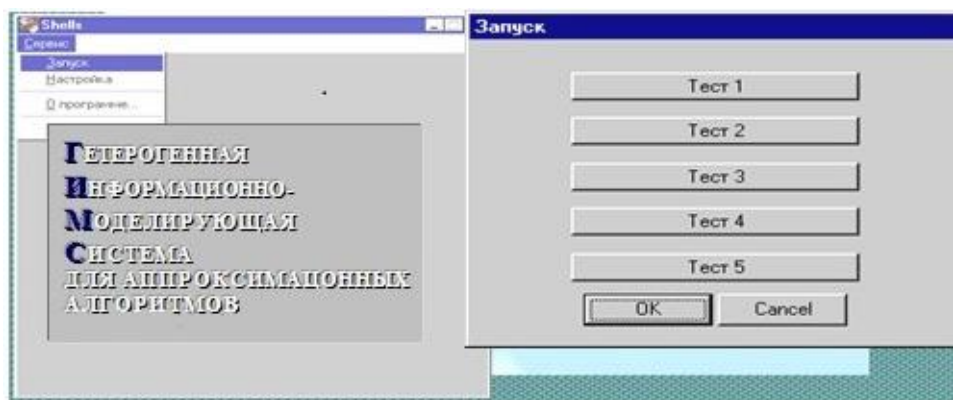
Цель работы

Основная цель работы состоит в развитии технологии гетерогенных информационно-моделирующих систем и создание аппаратно-программных средств применительно к региональным условиям исследования.

Теоретические основы аппроксимационных методов (метод Колмогорова для аппроксимации многоэкстремальных функций многих переменных [3] или метод ветвей и границ для решения задач теории графов [4] и т.д.) позволяют естественным образом реализовать параллельные вычисления как на уровне библиотек алгоритмов, так и на уровне структур отдельных алгоритмов [5].

Состав и структура системы. Кластер ГИМС – гетерогенная структура, в которую входят компьютеры, построенные на аппаратных платформах IntelPentium различных версий с установленными системами Linux и Windows.

Безопасность при обработке данных является одним из основных вопросов при работе с облачными и кластерными средствами особенно в отношении конфиденциальных, персональных данных. Для обеспечения защищенного доступа создана инструментальная оболочка диспетчера задач, работающая в режиме запуска программных средств и в режиме настройки кластера. Рабочие окна оболочки представлены на рис. 1.



а) рабочий режим



б) режим настройки

Рис. 1. Рабочее окно инструментальной оболочки

Основным элементом системы является адаптивная библиотека алгоритмов и программ. Большая актуальность применения методов аппроксимации многоэкстремальных функций при изучении сложных объектов и определенная трудность для широкого пользователя в выборе подходящего алгоритма при решении конкретной задачи, делают необходимым проведение группировки и классификации используемых подходов. Классификация обеспечивает строго постоянное размещение и ускоряет поиск алгоритмов в адаптивной библиотеке алгоритмов и программ.

При искусственной систематизации выбраны нижеследующие признаки, удобные для создания библиотеки.

1. Цели применения аппроксимации:

- исследование и прогнозирование характеристик физико-химических процессов в сложных технических системах;
- исследование и прогнозирование социально-экономических процессов в эргатических системах;
- исследование и обработка информационных данных в сложных системах.

Этот класс может быть разделен на подклассы алгоритмов интерполяции и экстраполяции.

2. Режимы аппроксимации (статический и динамический режим реального времени).

3. Используемые критерии и ограничения (точность решения, время поиска решения, число основных функций и их параметров, аддитивные или мультипликативные комбинации исходных критериев). Все практически применяемые алгоритмы в качестве основного критерия (ограничения) используют априорно заданную точность.

4. Применяемые классы и комплекты основных функций.

5. Методы поиска решений:

- детерминированные (аппроксимация со смещением величину тренда, моды, среднего или минимального значения, аппроксимация на постоянной и переменной сетке, алгоритмы направленного перебора, «жадные» алгоритмы и т.п.);
- статистические (метод Монте Карло для одного параметра, метод Монте Карло для группы параметров);
- эвристические (методы прогноза и коррекции, методы гипотез и ограничений);
- метаэвристические (различные модификации методов решения задачи о ранце или транспортной задачи, например, метода “модельной” закалки или метода отжига, метода поиска с запретами, метода муравьиной колонии, генетических и эволюционных алгоритмов [6]).

Сравнении алгоритмов происходит опытным путем и использует различные вычислительные ресурсы и типы решаемых задач, профессиональный уровень программирования алгоритма влияет на время поиска решения [7]. Тестирование алгоритмов на тестовых задачах показало, что качество решений полученных с использованием метаэвристических алгоритмов превосходит качество решений полученных с помощью классических конструктивных методов и методов локального поиска. Метаэвристические методы являются дальнейшим развитием эвристических методов, и в связи с полученными результатами являются перспективным направлением в решении задач экспресс-аппроксимации в условиях меняющихся целей и критериев. Метаэвристические методы более затратные по

времени и более сложные в применении, однако именно они являются наиболее эффективными в решении практических задач.

Реализация

Разработанная многофункциональная информационно-моделирующая система была испытана в условиях мониторинга гетерогенных ракетно-космических комплексов. Практическая значимость результатов работы определяется тем, что применение ГИМС повышает эффективность проведения экспериментов путем сокращения объемов измерений при достижении поставленной цели и обеспечивает решение задачи оптимизации режима внешнего влияния комплексов.

Эффективность разработанной системы, ее алгоритмического и программного обеспечения подтверждена результатами обработки многочисленных данных различных исследований, включая:

- совместные физические эксперименты, проведенные филиалом «Восход» МАИ и институтом ионосферы АН КазССР;
- лабораторные и полевые эксперименты, проведенные в рамках проекта МНТЦ № К-298 по установлению границ и изучению внутренней структуры зон влияния деятельности Байконурского космодрома и Сарышаганского полигона.

Для учебных целей создаются виртуальные кластеры, ориентированные на решение тематических задач. С целью ускорения выполнения программ используются кластеры размером не больше пяти узлов. В том случае когда нет необходимости устанавливать стационарный кластер (когда компьютерный класс после проведения занятий не используется в качестве вычислительного кластера) развивается виртуальный кластер с версией дистрибутива Pelican HPC GNU Linux [8]. Целенаправленная реализация при этом принципов параллельного моделирования существенно повышает эффективность проведения исследовательских работ и учебных занятий.

Список литературы

1. Динамика гетерогенных структур : в 3 т. / под ред. В. В. Смогунова. – Пенза : Изд-во Пенз. гос. ун-та, 2001. – Т 1. Эволюция ракетно-космических гетерогенных структур. – 311 с.
2. Петрунина, Е. В. Метод экспресс-обработки экспедиционных данных о загрязнении почв / Е. В. Петрунина, Б. П. Селенко // Measuring and computing devices in technological processes : International scientific-technical magazine, Khmel'nitsky. – 2002. – № 2. – С. 122–128.
3. Колмогоров, А. Н. О представлении непрерывных функций многих переменных в виде суперпозиции непрерывных функций одного переменного / А. Н. Колмогоров // Доклады АН СССР. – 1957. – Т. 114, № 5. – С. 953–956.
4. Ананий, В. Левитин. Алгоритмы: введение в разработку и анализ / В. Ананий. – М. : Вильямс, 2006.
5. Дорошенко, А. Е. Математические модели и методы организации высокопроизводительных параллельных вычислений / А. Е. Дорошенко. – Киев : Наукова думка, 2000.
6. The VRP with Time Windows. Chap. 7 / J.-F. Cordeau, Guy Desaulniers, Jacques Gesrosiers, Marius M. Solomon, Francois Soumis // SIAM, Monographs on Discrete Mathematics and Applications / Paolo Toth and Daniel Vigo (eds). – 2001.
7. Gendreau, M. Metaheuristic Approaches for the Vehicle Routing Problem with Time Windows / M. Gendreau, O. Bräysy // A Survey. MIC2003: The Fifth Metaheuristics International Conference, Kyoto, Japan, August 25–28, 2003. – Kyoto, 2003.
8. Сбитнев, Ю. Linux. Кластер. Практическое руководство по параллельным вычислениям / Ю. Сбитнев. – URL: <http://cluster.linux-ekb.info/>

М. С. Зыков, И. В. Огнев, А. Г. Горьков

Россия, Москва, Московский энергетический институт

СПОСОБ ФОРМИРОВАНИЯ БАЗЫ ДАННЫХ ДЛЯ ФРАГМЕНТАРНОГО МЕТОДА СЖАТИЯ ИНФОРМАЦИИ

Проведено исследование производительности структур данных для дальнейшей адаптивной модификации алгоритма построения деревьев секущих с учетом энтропийных характеристик входного потока с целью построения при помощи данного алгоритма баз данных для энтропийного кодирования и передачи информации.

В данной работе исследуется производительность ряда структура данных применительно к их использованию в качестве структур данных, взятых за основу для формирования больших и сверхбольших баз данных. Подобные базы данных используются при энтропийном кодировании для сжатия и передачи информации [1]. Базы данных, о которых идет речь в данной работе, отличаются от классических представлений о базах данных. Некоторая система (устройство наблюдения, спутник или нечто другое) генерирует данные, представленные бинарными строками фиксированной длины, которые идут на вход принимающего устройства. Приходящий поток данных обрабатывается, и конечным результатом является список уникальных строк с указанием частоты их появлений.

Таким образом, под термином «база данных» в данной работе понимается набор уникальных элементов, строк, с указанием частоты их появления.

Формирование базы данных (проверка наличия строки, совпадающей с пришедшей на вход системы, увеличение счетчика появлений и т.д.) должно происходить в реальном времени. Итоговая база данных в процессе работы по возможности должна храниться в ОЗУ. В том случае, если объема ОЗУ не хватает для эффективной работы системы, база данных сохраняется на внешний носитель.

Приходящий поток данных обладает определенными энтропийными характеристиками, которые должны быть учтены при разработке алгоритма формирования баз данных, а именно: определенные элементы могут встречаться значительно чаще, чем другие, вплоть до того, что весь входной поток может состоять более чем на 50–60 % из одного повторяющегося элемента. Соответственно, алгоритм должен обеспечивать более быстрый доступ к наиболее часто встречающимся на входе системы значениям.

Существует алгоритм получения баз данных, подобных рассматриваемым в данной работе, реализующий построение деревьев секущих, представленный в работе А. И. Огнева, и учитывающий энтропийные характеристики, однако его существенный недостаток состоит в том, что деревья секущих могут быть построены только для базы данных с заранее известными энтропийными характеристиками [2, 3, 4]. Анализ, проведенный в данной работе, был проделан с целью разработки адаптивной модификации данного алгоритма, включающей в себя возможность перестройки дерева секущих с учетом особенностей входного потока данных.

Далее рассмотрен ряд структур данных, которые возможно использовать в качестве основы для модификации алгоритма построения деревьев секущих:

1. Двоичное дерево поиска;
2. Красно-черное дерево;
3. Сплей-деревья;
4. AVL-деревья;

Двоичное дерево поиска – это двоичное дерево, для которого и правое, и левое поддеревья являются двоичными деревьями поиска, у всех узлов левого поддерева произвольного узла X значения ключей меньше, чем значения ключей самого узла X , а значения всех узлов правого поддерева того же узла X больше, чем значения ключа данных узла X . В среднем данная структура данных обеспечивает временную сложность $O(\log n)$ для индексации, поиска, вставки и удаления узлов.

Красно-черное дерево – это самобалансирующееся двоичное дерево поиска, гарантирующее логарифмический рост высоты дерева от числа узлов и быстрое выполнение операций поиска, добавления и удаления узла. Сбалансированность достигается за счет введения дополнительного атрибута узла – «цвета», который может принимать одно из двух значений – «черный» или «красный».

Сплей-деревья являются двоичными деревьями поиска, в которых поддерживается свойство сбалансированности. При каждом обращении к дереву происходит изменение его структуры с целью поддержания сбалансированности. Учетная стоимость в расчете на одну операцию с деревом составляет $O(\log n)$. Сплей-деревья обладают особенностью хранить узлы, к которым часто происходит обращение, вблизи вершины дерева, в то время как узлы к которым обращение происходит редко перемещаются ближе к листьям. Таким образом, время обращения к часто посещаемым узлам будет меньше, а время обращения к редко посещаемым узлам – больше среднего.

AVL-деревья – это сбалансированное по высоте двоичное дерево поиска, для каждой вершины которого высота двух его поддеревьев различается не более чем на 1.

Как было сказано, алгоритм должен обеспечивать быстрый доступ к наиболее часто встречающимся элементам, поэтому в ходе исследования был проведен ряд тестов для определения средней длины пути при поиске элемента в деревьях разных типов. Данный параметр является одним из определяющих в выборе подходящей для дальнейшей работы структуры данных.

В качестве тестовых данных были выбраны целые числа в диапазоне от 0 до 1000000000. Каждый тест проводился пять раз, а затем среди них бралось среднее арифметическое значение. На вход системы поступают последовательности элементов разной длины. В табл.1 представлены данные о средней длине траектории, полученные при входном потоке, обладающим нормальным распределением:

Таблица 1

	Двоичные деревья поиска	Красно-черные деревья	Сплей-деревья	AVL-деревья
100	18.850	17.960	18.510	17.910
1000	65.189	44.234	51.163	43.184
10000	147.196	81.697	102.480	80.414
100000	229.398	130.779	173.169	129.102
1000000	342.601	190.874	261.794	188.692
10000000	443.184	258.617	361.466	258.107
100000000	571.887	340.270	407.825	340.270

В табл. 2 представлены результаты, полученные при работе с входным потоком, обладающим нестандартными энтропийными характеристиками, а именно: определенные элементы в нем составляют более 50% входного потока.

Таблица 2

	Двоичные деревья поиска	Красно-черные деревья	Сплей-деревья	AVL-деревья
100	10.930	11.800	10.650	11.850
1000	33.017	33.566	24.891	33.053
10000	79.773	62.684	49.081	62.416
100000	104.499	105.581	83.278	113.278
1000000	180.092	154.902	126.534	166.090
10000000	248.547	206.514	177.335	211.496
100000000	291.472	247.053	216.788	236.159

Исходя из полученных данных, можно сделать вывод, что дальнейшие модификации алгоритма построения деревьев секущих должны в первую очередь базироваться на основе таких структур данных, как сплей-деревья. На входном потоке с нормальным распределением лучшие результаты показали красно-черные деревья и AVL-деревья, что можно объяснить алгоритмами перебалансировки деревьев. Двоичные деревья поиска дают выигрыш при малом количестве элементов во входном потоке, однако затем отсутствие встроенных механизмов перебалансировки не дает им показать лучшие результаты. Результаты сплей-деревьев посредственные, так как на нормальном распределении, при котором во входном потоке нет часто повторяющихся элементов, особенности сплей-деревьев не дают большой выгоды, а постоянные перестроения дерева лишь замедляют работу системы.

С другой стороны, на входном потоке с заданными энтропийными характеристиками, сплей-деревья демонстрируют заметный отрыв от остальных типов деревьев. Данные результаты можно объяснить свойством сплей-деревьев перемещать элементы, к которым идет наиболее частое обращение, ближе к корню, что позволяет сократить траекторию до них до минимальной.

Таким образом, исходя из полученных результатов, можно сделать вывод, что для дальнейшей модификации алгоритма построения деревьев секущих, целесообразно использовать особенности сплей-деревьев, совместив возможности сплей-деревьев по изменению своей структуры при каждом обращении и перемещению часто встречающихся элементов ближе к корню с алгоритмом построения деревьев секущих, для реализации адаптивной его модификации. Данные, полученные в работе Бена Плаффа «Анализ производительности двоичных деревьев поиска в системном программном обеспечении» также подтверждают полученные результаты и сделанные выводы [5]. Авторы резюмируют, что лучше всего на наборах данных с повторяющимися элементами ведут себя сплей-деревья.

Список литературы

1. Огнев, И. В. Алгоритм формирования базы данных для фрагментарного метода сжатия видеопотока без потерь / И. В. Огнев, А. И. Огнев, А. Г. Горьков // Информационные средства и технологии : тр. XX Междунар. науч.-техн. конф. – М. : Издат. дом МЭИ, 2012. – Т. 1. – С. 67–77.
2. Огнев, И. В. Аппаратная реализация дерева секущих на ПЛИС / И. В. Огнев, А. И. Огнев, А. Г. Горьков // Информационные средства и технологии : тр. XVIII Междунар. науч.-техн. конф. – М. : Издат. дом МЭИ, 2010. – Т. 1. – С. 36–43.
3. Огнев, И. В. Оптимизация конфигурации окна сканирования в фрагментарном методе сжатия видеопотока без потерь / И. В. Огнев, А. И. Огнев, А. Г. Горьков // Информационные средства и технологии : тр. XX Междунар. науч.-техн. конф. – М. : Издат. дом МЭИ, 2012. – Т. 1. – С. 78–85.

4. Огнев, И. В. Предварительная обработка кадров видеопотока для алгоритма фрагментарного сжатия видеопотока / И. В. Огнев, А. И. Огнев, А. Г. Горьков // Информационные средства и технологии : тр. XXI Междунар. науч.-техн. конф. – М. : Издат. дом МЭИ, 2013. – Т. 1. – С. 47–52.

5. Plaff, B. Performance analysis of BST in system software. / B. Plaff // ACM SIGMETRICS Performance Evaluation Review. – 2004. – № 32(1). – P. 410–411.

С. В. Дамбаева, Б. Д. Мердыгеев

Россия, Улан-Удэ, Восточно-Сибирский государственный университет технологий и управления

СТРУКТУРЫ ОНТОЛОГИИ

Предлагаются два направления дальнейшего совершенствования структуры онтологии предметной области: структура, ориентированная на четкую типизацию терминов, и структура, ориентированная на однотипность терминов.

Онтология в философском понимании является учением о бытии как таковом и представляет собой раздел философии, изучающий фундаментальные принципы бытия, его наиболее общие сущности и категории, структуру и закономерности окружающего мира в целом. В информатике термин «онтология» означает формальное представление не просто данных, а знаний. Она используется как форма представления знаний о реальном мире или его части. Основные сферы ее применения – моделирование бизнес-процессов, семантическая паутина, искусственный интеллект. Таким образом, в информатике онтология – это модель предметной области, использующая все доступные средства представления знаний, релевантные для данной области. На современном этапе развиваются, в основном, следующие типы онтологий: предметно-ориентированные (Domain-oriented), ориентированные на прикладную задачу (Task-oriented), общие онтологии (Top-level ontologies) [4].

Одной из основных задач в создании онтологий является задача определения структуры онтологии и представления ее в формальном виде. Существует множество видов представления знаний в форме онтологии. Большинство из них ориентировано на конкретные задачи и области знаний. Поэтому актуальной является задача определения такой структуры онтологии, которая могла бы объединить знания о различных предметных областях.

В данной статье предпринята попытка определения универсальной структуры онтологии. За основу дальнейших рассуждений о возможностях ее универсализации взята структура, описанная в [1], поскольку она не привязана к конкретной предметной области.

Представление онтологии. В идеале онтология есть способ представления знаний в компьютерной системе, который позволяет системе «понимать» смысл терминов онтологии, при этом воспринимать термин не только как отдельный объект, но и определять его во взаимодействующей с ним окружающей среде.

Приведем краткое формальное описание структуры онтологии [1].

Термин может относиться к одному из концептуальных объектов: понятие, действие, состояние, событие, свойство или величина. Конструкция знака концеп-

туальных объектов четко определяет структуры словарных статей тезауруса. Коротко рассмотрим эти структуры.

Словарная статья «Понятие» представляет собой восьмерку:

$$\mathbf{Concept} = \langle t, D, P, A, C, S, T, M \rangle, \quad (1)$$

где t – имя термина, тип концептуального объекта «Понятие» и вид сущности: материальный или нематериальный; D – множество субстанциальных дефиниций; P – множество пар (p_1, p_2) , где p_1 – имя свойств, p_2 – ссылка на словарную статью; A – множество пар (a_1, a_2) , где a_1 – действие, a_2 – ссылка на словарную статью; C – синонимы и корреляты (квантитативные отношения); S – множество состояний сущности; T – множество терминов, имеющих квалитативные отношения с термином t (родовитые отношения, отношения «часть-целое»); M – способы метаязыкового представления, который позволяют зафиксировать связь термина и его представления в метаязыке.

Словарная статья «Действие» представляет собой девятку:

$$\mathbf{Action} = \langle t, D, P, SO, C, I, T, E, M \rangle, \quad (2)$$

где SO – субъект или объект действия и ссылка на его описание; I – имя термина, идентифицирующего инструмент действия и ссылка на словарную статью этого термина; E – событие, при наступлении которого выполняется действие.

Словарная статья «Состояние» представляет собой шестерку:

$$\mathbf{State} = \langle t, D, P, E, C, M \rangle, \quad (3)$$

где E – событие, на основании которого объект переходит в данное состояние.

Словарная статья «Событие» представляет собой семерку:

$$\mathbf{Event} = \langle t, D, P, Y, C, S, M \rangle, \quad (4)$$

где Y – обозначение условия наступления события, его определение и тип определения условия; S – текущее состояние и новое состояние, в которое переводится объект при наступлении события.

Словарная статья «Свойство» представляет собой шестерку:

$$\mathbf{Property} = \langle t, D, K, Q, C, M \rangle, \quad (5)$$

где K – вид свойства (количественное или качественное) и вид свойства в другом отношении (показатель, параметр, фактор, признак и т.д.); Q – имя величины и ссылка на словарную статью величины.

Словарная статья «Величина» представляет собой шестерку:

$$\mathbf{Quantity} = \langle t, D, MS, L, U, M \rangle, \quad (6)$$

где MS – шкалы измерения, качественная и количественная, с входящими в них множествами значений; L – нижнее и верхнее нормальные значения; U – единица измерений.

Согласно приведенным структурам словарных статей внутрикатегориальные квалитативные отношения могут иметь только термины концептуального объекта «Понятие».

В онтологии все термины определяются через отношения с другими терминами. Все термины взаимосвязаны, что позволяет ассоциативно «определять значение» компьютерной системой.

Приведенная структура представляет собой категориальный аппарат, позволяющий выделить основные элементы онтологии в виде знаков концептуальных

объектов и их представление в виде словарных статей. Поэтому ее можно считать базовой (исходной).

Предлагаются два возможных направления развития данной структуры. Первое направление является путем прямого развития исходной структуры. Он ориентирован на рассмотрение терминов как предметов реального мира, из чего следует четкое разделение терминов на типы (концептуальные объекты). Второе направление основывается на представлении всех терминов в виде однотипных объектов, которые идентифицируются отношениями, связями и взаимодействием с другими терминами. Опираясь на принципы исходной структуры, это направление развивает другой путь представления онтологии.

Структура онтологии, ориентированная на четкую типизацию терминов. Структура онтологии [1] позволяет описать термин по любой из вышеуказанных словарных статей, т.е. в ней нет механизма определения типов терминов. Основой предлагаемой структуры является четкое разделение терминов онтологии по типам, как и в исходной. Каждый термин соответствует предмету реального мира, каковым может быть любая сущность, действие, событие и т.д., т.е. часть реального мира (предметной области), ассоциируемого с термином. Один и тот же предмет не может быть отнесен к нескольким типам одновременно. Однако может быть несколько терминов, выражающих его, а также один термин может быть соотнесен с разными предметами. Поэтому следует различать такие термины.

В исходной структуре онтологии термины предметной области делятся по шести типам концептуальных объектов. Для того чтобы ввести более точную типизацию терминов, дадим им следующие краткие определения:

- Понятие – это сущность реального мира, выраженная предметом термина, который может являться субъектом или объектом действий или процессов в реальном мире.
- Действие – это изменение или процесс, приводящий к смене состояния предмета термина. Действие является продолжительным процессом в отличие от события.
- Состояние – это определенная конфигурация предмета термина. Состояние меняется, когда производятся некоторые действия над предметом или самим предметом, который описывается данным состоянием.
- Событие – это то, что имеет место, происходит, наступает в произвольной точке пространства и времени (явление, происшествие факт и т.д.). Событие стоит между разными состояниями системы (предмета) и в отличие от действия, которое определяет сам процесс перехода из одного состояния в другой, событие описывает сам факт перехода.
- Свойство – это характеристика предмета термина.
- Величина – это мера, используемая для измерения свойств предмета термина.

Отличием от исходной структуры является то, что концептуальные объекты представляются не как отдельные категории, а как упорядоченная иерархическая система уровней предметов реального мира. Поэтому термины онтологии распределяются по уровням, соответствующим каждому типу концептуальных объектов. В табл. 1 представлены уровни иерархии и связанные с ними другие уровни.

Отношения между терминами устанавливаются на основе отношений между знаками концептуальных объектов исходной структуры. В отличие от исходной структуры здесь допускается возможность наличия внутрикатегориальных качественных отношений на каждом уровне терминов, кроме величины. Каждый уро-

вень показывает отдельную описательную часть реального мира. Эти части не пересекаются. Термины, имеющие одинаковые имена, но относящиеся к разным частям, воспринимаются отдельно друг от друга. Они не являются равнозначными и строго распределяются по иерархии уровней.

Таблица 1

Уровни структуры онтологии

Уровень термина	Связанные уровни терминов
Понятие	Действие, состояние, свойство
Действие	Состояние (как условие), понятие (как объект), свойство
Состояние	Событие (как источник), свойство
Событие	Состояние (текущее и новое), свойство
Свойство	Величина
Величина	–

Следует учитывать, что существуют качественные и количественные отношения между терминами одного уровня. Таким образом, основными параметрами таких терминов могут быть идентификатор (номер), имя (все его формы), тип, описание (дефиниции), качественные отношения, количественные отношения и отношения с терминами других типов. Такие параметры определены для термина любого типа.

Для анализа онтологий данной структуры может быть использован метод FCA, описанный в источнике [5]. Дополнительную информацию по данной структуре и отношениям между терминами можно получить в источниках [1] и [3].

Основным достоинством данной структуры является то, что она позволяет составить большую многоаспектную онтологию, описывающую окружающую действительность. Одновременно с этим, недостатком структуры является сложность автоматического формирования онтологии, которая заключается в сложности определения типов терминов и видов отношений между ними. Для упрощенного формирования можно использовать базовые термины или базовую онтологию, представляющую некоторое ядро терминов, на основе которого происходит дальнейшее расширение онтологии. Также к недостатку можно отнести сложность представления абстракций (формул, схем и т.д.) в онтологии, сохраняя их семантические свойства.

На основе данной структуры возможна реализация структуры онтологии с активной семантикой. Это сильно усложнит саму структуру, однако это позволит более полно описывать реальный мир.

Структура онтологии, ориентированная на однотипность терминов. В данном случае структура онтологии основывается на однотипности терминов, что значит, термины не имеют разграничений по типам предметов реального мира. Однако типы концептуальных объектов используются для нечеткой типизации терминов через роли терминов в отношениях. В данном случае ролью термина является его представление в конкретной ситуации, определяемой отношением.

Термины в основном определяются отношениями с другими терминами. Один и тот же термин может исполнять различные роли в зависимости от контекста. Для каждого термина отношение определяет собственную роль в зависимости от типа термина (рисунок 1). Таким образом, термины могут иметь различные роли в зависимости от отношения. Отношение может связывать больше чем два термина (к примеру, отношение типа субъект-действие-объект). Роли также могут

быть составные, т.е. в отношении термину присваивается не одна, а несколько элементарных ролей. Таким образом, главной отличительной чертой этого направления является то, что термин в первую очередь ассоциируется не с предметом реального мира, а со словом или словосочетанием, которое является именем термина.

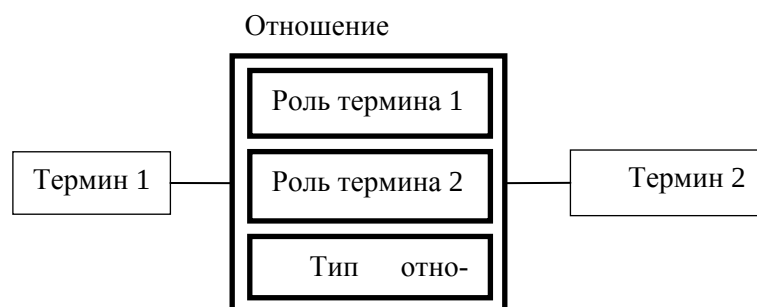


Рис. 1. Схема отношения между двумя терминами

Роли терминов могут включать типизацию по знакам концептуальных объектов исходной структуры. При этом в одном отношении термин может быть определен ролью не более чем одного знака.

Также следует учитывать особенности самого имени, выражающего термин. В данном случае необходимо четко определить все возможные формы имени термина, такие как падежи у существительного. Также эти формы не являются равнозначными. Определение значений формы имени может являться частью структуры онтологии. Формы имени могут влиять на типизацию отношений между терминами. Это увеличит семантическую ценность имени термина.

В онтологии могут храниться знания, извлеченные из разных источников. Из-за этого некоторые термины могут пониматься по-разному в зависимости от контекста или источника. Поэтому необходимо определить способ разграничения отношений терминов в зависимости от контекстов. Для этого онтология может состоять из двух списков. Первый список — это список терминов, второй — отношений. То есть отношения определяются отдельно от терминов. Однако один и тот же термин можно отнести к контекстам с разными значениями и ролями. Для выделения контекстов из онтологии можно составлять маски на отношения.

Одним из возможных развитий данного направления может быть структура, определяющая «себя через себя». То есть типы отношений и роли могут определяться через другие термины онтологии. В такой онтологии для поддержания семантики необходимо определить ядро базовых терминов и отношений, на основе которых формируются более сложные отношения и структуры терминов.

Онтология, основанная на данной структуре, может быть достаточно многообразной и хранить в себе разностороннюю информацию. Она подходит для описания предметов, как реального мира, так и его абстракций. Это упрощает описание семантической части структуры онтологии. При автоматическом расширении данной онтологии следует определить базовые отношения. Построение и расширение онтологии такой структуры сильно зависит от определенного ядра онтологии.

К недостаткам данной структуры можно отнести сложный анализ онтологии, так как термины не являются типизированными, и все роли терминов определены в отношениях.

В статье описаны структура онтологии из монографии [1] и эта структура была определена как исходная для двух направлений структур. Первое направление — структура, ориентированная на четкую типизацию терминов в соответствии

с типами предметов реального мира. Второе направление – структура, ориентированная на однотипность терминов и определение их через роли отношений. Каждое из направлений имеет потенциал к развитию и практическому применению.

Над онтологией могут проводиться следующие действия: создание начальной онтологии по источнику знаний, расширение онтологии за счет других источников знаний, извлечение знаний из онтологии и оценка онтологии. Последние два действия относятся к анализу онтологии. Формирование (построение) онтологии, основанной на первой структуре, проще, чем онтологии, основанной на второй структуре, так как первая структура имеет определенность в делении терминов по типам на основе частей реального мира. Во второй структуре сложнее определить роли терминов в каждом отношении. Расширение и анализ онтологии в обоих случаях является трудной задачей. В анализе онтологий возможно использование масок отношений (если отношения определены отдельно от терминов). Однако в первой структуре отделение отношений от терминов усложняет саму структуру онтологии, что увеличит объем анализируемой информации и соответственно время анализа.

Список литературы

1. Найханова, Л. В. Технология создания методов автоматического построения онтологий с применением генетического и автоматного программирования : моногр. / Л. В. Найханова – Улан-Удэ : Изд-во БНЦ СО РАН, 2008. – 244 с.
2. Найханова, Л. В. Анализ структуры автоматически генерируемой онтологии и постановка задачи ее преобразования в решетки формальных контекстов / Л. В. Найханова, С. В. Дамбаева // Математика, ее приложения и математическое образование (МПМО'11) : материалы IV Междунар. конф. – Улан-Удэ : Изд-во ВСГТУ, 2011. – Ч. 2. – С. 58–62.
3. Дамбаева, С. В. Способы определения структуры формального контекста на решетках понятий / С. В. Дамбаева, Б. Д. Мердыгеев / Математика, ее приложения и математическое образование (МПМО'14) : материалы V Междунар. конф. – Улан-Удэ : Изд-во ВСГТУ, 2014. – С. 99–105.
4. Место онтологий в единой интегрированной системе РАН // А. Н. Бездушный, Э. А. Гаврилова, В. А. Серебряков, А. В. Шкотин. – URL: http://www.benran.ru/Magazin/cgi-bin/Sb_03/pr03.exe?!15.
5. Ganter, B. Applied Lattice Theory: Formal Concept Analysis / B. Ganter, R. Wille. – URL: <http://www.math.tu-resden.de/~ganter/psfiles/concept.ps>

Э. В. Цукарев

Россия, Пенза, Пензенский государственный университет

МОДЕРНИЗАЦИЯ ИНИЦИАТИВНОГО ОПРОСА НА СЕРВЕРЕ КОНСОЛИДАЦИИ ТЕХНОЛОГИЧЕСКИХ ДАННЫХ

Рассматриваются изменения, внесенные в инициативный опрос технологических точек Сервера консолидации технологических данных, и связанные с ними изменения в управление проектом передачи данных.

Эксплуатация программно-технического комплекса «ЭнергоКруг» с остро поставила вопрос модернизации инициативного опроса [1], реализованного в ЭнергоКруг 11. Действующая реализация не учитывает множество параметров необходимых для получения данных по протоколу OPC DA/HDA.

Например, в параметрах инициативного запроса на чтение оперативных данных, приводятся только имена тегов и имя OPC – сервера. Библиотеки чтения OPC DA тегов содержат множество других параметров запроса, в частности:

- синхронный/асинхронный режим чтения;
- временная задержка перед инициированием повторного чтения DA данных;
- признак загрузки атрибутов на старте запроса;
- источник данных оперативных точек – кэш или устройство;
- максимальное отклонение точек от истинных значений;
- признак обновления атрибутов;
- интервал проверки состояния OPC DA сервера и другие.

Все вышеприведенные параметры будут устанавливаться в значение по умолчанию при реализации инициативного опроса, приведенного в статье [1]. Более того, обработать полученные технологические точки по какому-либо выбранному математическому алгоритму, не представляется возможным, так как компонент математической обработки не предусмотрен в инициативном опросе, приведенном в документе [1]. Таким образом, схема инициативного запроса точек, является слабо конфигурируемой и плохо масштабируемой.

Для устранения выше перечисленных недостатков инициативного опроса, необходимо маршрутизировать запрос, полученный в Сервере консолидации, от Сервера шлюза, в соответствующий проект передачи данных. Далее в проекте передачи данных, запрашивать точки у OPC – сервера, заданного в проекте, использовать при запросе параметры, заданные в компоненте получения данных. Следующим этапом полученные точки будут передаваться по всей цепочке компонентов проекта. Таким образом, технологические данные для примера, приведенного на рисунке, будут автоматически обработаны компонентом Вери-фикации, компонентом математической обработки и записаны в одну или две разные БД в зависимости от того какая строка подключения задана в компонентах записи в БД.

Для реализации функционала, описанного в последнем абзаце, необходимо модернизировать управление проектом передачи данных. Требуется ввести команду получения словаря тегов заданного проекта. Процесс проекта будет передавать процессу Сервера консолидации список зарегистрированных тегов. Таким образом на Сервере консолидации будет храниться консолидированный словарь всех технологических тегов с привязкой к проектам передачи данных. Следовательно, необходимо добавить следующие команды для Проекта передачи данных:

- получить карту тегов;
- выполнить запрос;
- остановить получение карты тегов;
- остановить запрос.

При выполнении любой команды, либо при невозможности выполнить ее в данный момент, в протокол событий будет выдано сообщение. Команды проекта не связанные с управлением остаются без изменений. Все возможные переходы при измененном управлении проектом передачи данных приведены на рисунке.

Активируется требуемый проект, с запуском процедур обработки точек в соответствии с деревом проекта. Данные обрабатываются с учетом всей конфигурации дерева проекта, записываются (если требуется в БД) и возвращаются инициатору внешнего запроса.

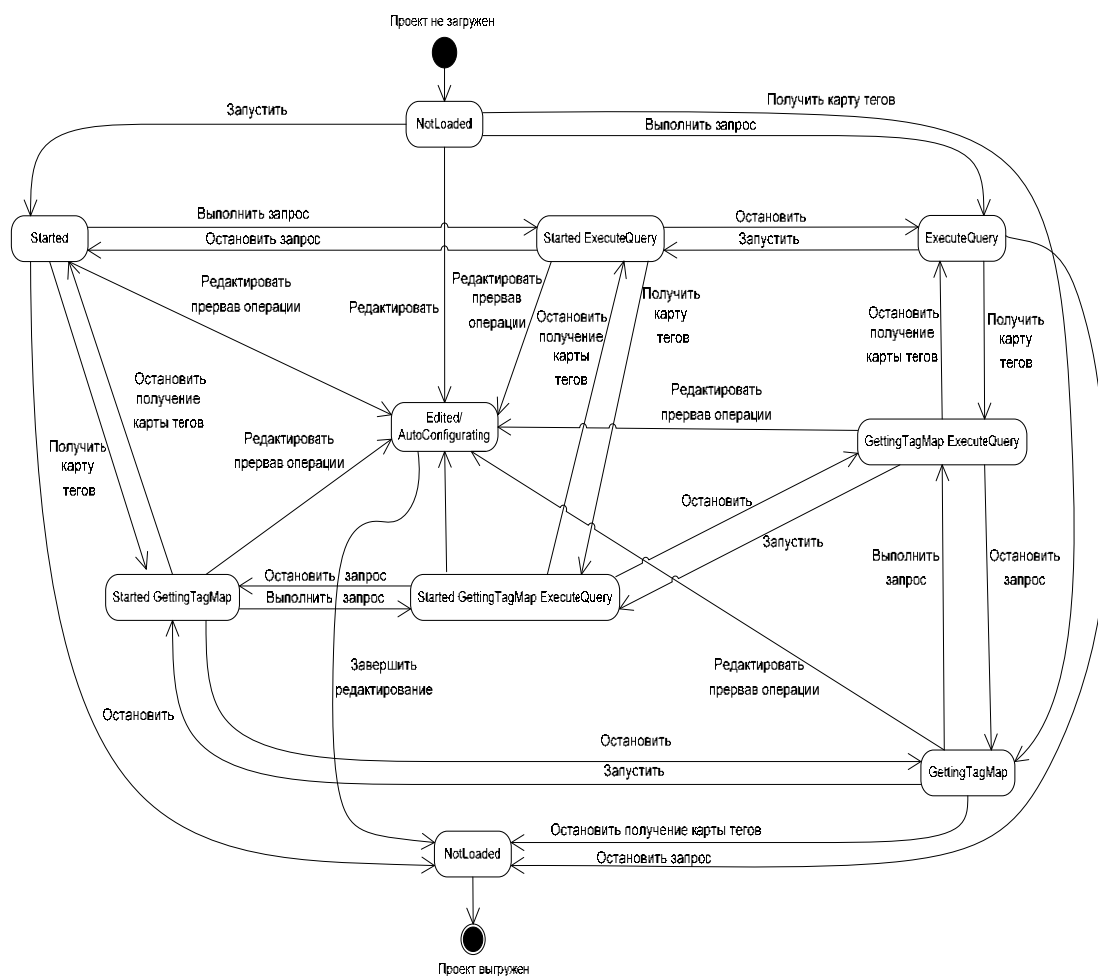


Рис. 1. Переход по командам при модернизации управления проектом

Введены следующие состояния проектов (см. таблицу):

Таблица 1

Возможные состояния проектов

Код состояния проекта	Название состояния проекта
NotLoaded	Проект не загружен
Edited	Проект редактируется
Started	Проект запущен
ExecuteQuery	Проект выполняет запрос
GetTagMap	Проект получает карту тегов
Started_ExecuteQuery	Проект запущен и выполняет запрос
Started_GetTagMap	Проект запущен и получает карту тегов
ExecuteQuery_GetTagMap	Проект выполняет запрос и получает карту тегов
Started_ExecuteQuery_GetTagMap	Проект запущен, выполняет запрос и получает карту тегов

Список литературы

1. Цукарев, Э. В. Реализация инициативного опроса на сервере консолидации технологических данных / Э. В. Цукарев // Открытые инновации – вклад молодежи в развитие региона. – Пенза : Изд-во ПГУ, 2013. – Т. 1. – С. 207–208.
2. Макарычев, П. П. Алгебраические модели процесса мониторинга распределенного программного проекта / П. П. Макарычев, В. В. Швецов // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2009. – № 3. – С. 76–82.

П. К. Чечель, А. А. Краснов, А. В. Дубравин

Россия, Пенза, Пензенский государственный университет

РАЗРАБОТКА УНИВЕРСАЛЬНОГО КАТАЛОГА ЦИФРОВЫХ И АНАЛОГОВЫХ ДАТЧИКОВ, ПРОИЗВОДИМЫХ НА ТЕРРИТОРИИ РФ И СНГ: СТРУКТУРА И СРЕДСТВА РЕАЛИЗАЦИИ

Рассмотрены проблемы разработки уникального универсального каталога цифровых и аналоговых датчиков в виде реляционной базы данных

В процессе работы в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития разработки научно-технологического комплекса России на 2014–2020» (соглашение № 14.574.21.0045 от 06.19.2014), задачей первого этапа явилось создание уникального универсального каталога цифровых и аналоговых датчиков в виде базы данных.

На первом шаге работы были определены подходы к решению поставленной задачи – разработка структуры описания датчиков, унификация структуры. Были созданы структуры данных, содержащие следующие поля и параметры, применимые ко всем типам датчиков при описании характеристик:

1. Марка устройства – название модели датчика, присвоенное ему на предприятии изготовителя.
2. Модификация – содержит сведения о конкретной модификации датчика, либо условное обозначение модификации.
3. Название – название датчика по функциональному назначению.
4. Производитель – содержит название производителя какого-либо датчика.
5. URL или имя файла – хранит путь к файлу-первоисточнику, из которого была взята информация о датчике, либо хранит URL-адрес, перейдя по которому, можно найти первоисточник сведений о датчике (как правило – это сайт предприятия изготовителя).
6. Интерфейс – способ подключения к датчику, другими словами это информация в виде название порта или разъема для подключения к датчику.
7. Группа – это группа, к которой относится датчик по функциональному назначению.
8. Подгруппа – иногда разбиения на группы недостаточно - для таких случаев выделен параметр «Подгруппа», хранящий сведения о дополнительной группировке.
9. Субподгруппа – дополнительный критерий группировки более высокого уровня.
10. Название параметра – полное наименование какого-либо параметра (характеристики) датчика.
11. Значение – содержит определенное значение параметра, описанного в предыдущем пункте. Данные хранящиеся в этом параметре могут быть как числовые, так и символьные.
12. Единицы измерения – единицы измерения для параметров «Значение», «Верхняя граница» и «Нижняя граница».
13. Верхняя граница – максимально допустимое значение какой-либо характеристики датчика.
14. Нижняя граница – минимально допустимое значение какой-либо характеристики датчика.

15. Тип (цифровой/аналоговый) – может содержать 3 значения: цифровой, аналоговый, аналого-цифровой.

16. Тип (вход/выход) – может содержать 3 значения: вход, выход, вход-выход.

17. Устройства сопряжения – содержит ссылки на другие устройства, которые можно соединить с датчиком. Параметр «Устройства сопряжения» как правило содержит марку устройства, для которого возможно сопряжение с описываемым датчиком.

18. Исполнение – описание исполнения датчика (или отдельных его частей, таких как корпус, измеряющий элемент и т.п.), содержит конструкционные или принципиальные особенности реализации датчика.

19. Примечание – текстовое поле, содержащее дополнительные сведения о датчике, не подходящие по формату, по содержанию или по смыслу для других полей параметров.

Следующий шаг работы над проектом заключался в выборе системы управления базами данных (далее СУБД). В результате рассмотрения нескольких вариантов СУБД (MSSQL, MySQL, Oracle, SQLite и других) выбор был сделан в пользу PostgreSQL.

Данная СУБД обладает рядом преимуществ:

- множество реализаций, модификаций и сборок;
- реализация на BSD-системах;
- имеет дистрибутивы для Linux, Mac OS X, а также для различных ОС семейства MicrosoftWindows;
- размер базы данных, ограничен только аппаратными ресурсами;
- наличие мощных и надежных механизмов транзакций и репликации;
- наличие обширной и свободно расширяемой системы встроенных языков программирования (в стандартном дистрибутиве поддерживаются PL/pgSQL, PL/Perl, PL/Python и PL/Tcl);
- наследование [1].

Также плюсом PostgreSQL является широкая поддержка индексов и удобная работа с ними, что позволяет применять их для более эффективной работы базы данных.

Поддерживаются индексы следующих типов: B-дерево, хэш, R-дерево, GiST и GIN. Важными особенностями являются поддержка частичных индексов и возможность просмотра индексов в обратном порядке.

Важными аспектами, повлиявшими на выбор СУБД, явились:

- возможность планировщика запросов использовать несколько индексов одновременно для выполнения сложных запросов;
- поддержка одновременной модификации базы данных несколькими пользователями с помощью специального механизма MultiversionConcurrencyControl (MVCC) [2];

MVCC – это многоверсионный контроль конкурентных транзакций, который управляет конкурентным доступом к данным на многоверсионной основе. Иными словами это означает, что при запросе к базе данных каждая транзакция видит «снимок данных» (версию) на момент этого «снимка», а не текущее состояние данных в базе. Таким образом, транзакции защищаются от просмотра нецелостных

данных, которые могут еще только формироваться другими конкурентными транзакциями в тех же самых строках таблицы.

Подобным образом достигается изоляция транзакций для каждой сессии применимой к базе данных.

ММВС позволяет избегать методов явной блокировки, которые применяются в некоторых альтернативных СУБД и таким образом, минимизирует блокировки данных, тем самым позволяя увеличить производительность в многопользовательской работе.

Неоспоримым преимуществом в ММВС является то, что чтение данных никогда не блокирует запись, а запись, в свою очередь, никогда не блокирует чтение.

Благодаря этому становится возможным соблюдение всех четырех требований ACID [3]:

- Atomicity – атомарность: это требование гарантирует, что никакая транзакция не будет зафиксирована в системе частично. И либо выполнены все ее подоперации до конца, либо не выполнено ни одной.
- Consistency – согласованность: транзакция достигая своего нормального завершения и, фиксирующая свои результаты, сохраняет согласованность базы данных. Проще говоря, каждая успешная транзакция фиксирует только допустимые (правильные) результаты.
- Isolation – изолированность: во время выполнения транзакции параллельные транзакции не должны оказывать влияние на ее результат.
- Durability – надежность: независимо от проблем на аппаратных уровнях (обесточивание системы или сбой) изменения, сделанные успешно завершенной транзакцией, должны сохраниться после возвращения системы в нормальный рабочий режим.

В процессе разработки была спроектирована и реализована база данных, отвечающая поставленным требованиям (создать универсальный каталог датчиков). Формально, в графическом виде, можно представить созданную базу данных в виде диаграммы из таблиц (сущностей базы данных) и связей между ними. Диаграмма базы данных представлен на рис. 1.

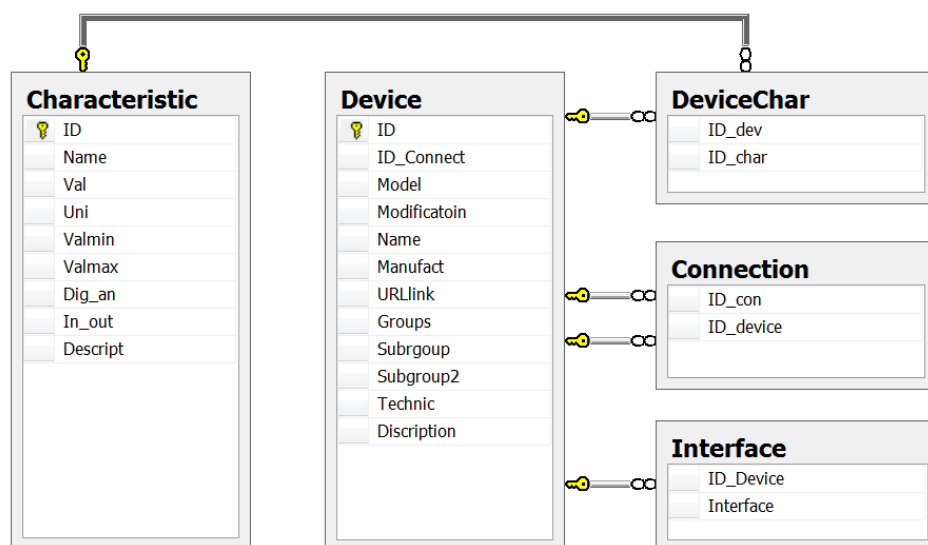


Рис. 1. Диаграмма базы данных

Предложенная структура базы данных обеспечивает минимальную избыточность данных, при этом сохраняется удобство выполнения различных запросов к разработанной базе данных, в частности поиск информации по различным параметрам.

Для решения задачи была выбрана удобная в использовании, безопасная и легко модифицируемая система управления базами данных PostgreSQL, а спроектированная модель базы данных позволяет хранить и обрабатывать информацию о различных датчиках в унифицированной форме.

Список литературы

1. Краткий обзор возможностей PostgreSQL: PostgreSQL.Ru.Net. – URL: <http://postgresql.ru.net/docs/overview.html> (дата обращения: 12.10.2014).
2. Васильев, А. Ю. Работа с PostgreSQL настройка и масштабирование / А. Ю. Васильев. – 4-е изд. – URL: <http://postgresql.leopard.in.ua> (дата обращения: 17.10.2014).
3. Бегг, К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / К. Бегг, Т. Коннолли. – М. : Вильямс, 2003. – 1436 с.

П. П. Макарычев

Россия, Пенза, Пензенский государственный университет

ПРОЦЕДУРА ПОИСКА ОПТИМАЛЬНОГО РЕШЕНИЯ МЕТОДОМ ПОСЛЕДОВАТЕЛЬНЫХ УСТУПОК

Рассматривается процедура многокритериальной оптимизации, реализуемая на основе преобразования целевых функций в ограничения типа неравенств с учетом заданных допустимых уступок.

Процедура решения многокритериальной задачи методом последовательных уступок используется при проведении системного анализа и создании систем поддержки принятия решений [1]. Процедура выполняется в несколько этапов.

На *первом этапе* все целевые функции располагают и нумеруют в порядке их относительной важности. На *втором этапе* определяют максимум (минимум) первой целевой функции. Назначают величину допустимого снижения значения первой целевой функции (уступки). На *третьем этапе* определяют максимум (минимум) второй по важности целевой функции при условии, что значение первой функции не должно отличаться от максимального значения более чем на величину уступки. Назначают величину уступки по второй функции. Далее подобным же образом поочередно используются все остальные целевые функции (критерии). Таким образом, при использовании метода последовательных уступок многокритериальная задача сводится к поочередной максимизации (минимизации) частных функций цели и выбору величин уступок [2].

Предположим, что задача многокритериальной оптимизации сформулирована в следующем виде:

- задана система ограничений

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,m} \\ a_{2,1} & a_{2,2} & \dots & a_{2,m} \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,m} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} \text{ или } \mathbf{M} \times \mathbf{x} \leq \mathbf{b}, \quad (1)$$

где $\mathbf{M}$ – матрица параметров ограничений; $\mathbf{x}$ – вектор переменных; $\mathbf{b}$ – вектор вещественных чисел, $x_i \geq 0, i = 1, 2, \dots, m$;

– задан набор функций цели в виде:

$$F_k(\mathbf{x}) = \sum_{i=1}^m c_{ki} x_i, F_k(\mathbf{x}) \rightarrow \max, k = 1, 2, \dots, K;$$

– установлена степень важности функций цели $\beta_k, k = 1, 2, 3 \dots K$ и величина уступки по каждой функции цели $\Delta_k, k = 1, 2, \dots, K - 1$.

Решаем первую оптимизационную задачу в следующей постановке:

$$\mathbf{M} \times \mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0, \text{ Maximize } (F_1(\mathbf{x}), \mathbf{x}).$$

Получаем решение $\mathbf{x}^* = (x_1^*, x_2^*, \dots, x_m^*)$. Находим значения всех функций цели $F_k(\mathbf{x}^*)$ и значение интегрального критерия:

$$\sum_{k=1}^K \beta_k \cdot F_k(\mathbf{x}^*).$$

Определяем допустимое отклонение первой функции цели $F_1(\mathbf{x}^*) - \Delta_1$.

Преобразуем первую функцию цели в новое ограничение. Поскольку при решении первой оптимизационной задачи искали максимум функции цели, то ограничение задаем в виде:

$$\sum_{i=1}^m c_{1i} x_i \leq F_1(\mathbf{x}^*) - \Delta_1. \quad (2)$$

С использованием введенного ограничения (2) модифицируем заданную систему ограничений (1). В результате имеем:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,m} \\ a_{2,1} & a_{2,2} & \dots & a_{2,m} \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,m} \\ c_{1,1} & c_{1,2} & \dots & c_{1,m} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \\ F_1(\mathbf{x}^*) - \Delta_1 \end{bmatrix} \text{ или } \mathbf{W} \times \mathbf{x} \leq \mathbf{b}^*, \mathbf{x} \geq 0, \quad (3)$$

Решаем вторую оптимизационную задачу

$$\mathbf{W} \times \mathbf{x} \leq \mathbf{b}^*, \mathbf{x} \geq 0, \text{ Maximize } (F_2(\mathbf{x}), \mathbf{x}).$$

Получаем новое решение $\mathbf{y}^* = (x_1^{**}, x_2^{**}, \dots, x_m^{**})$. Находим значения первых двух функций цели $F_k(\mathbf{y}^*), k = 1, 2$ и определяем $\sum_{k=1}^2 \beta_k \cdot F_k(\mathbf{y}^*)$. Если

$$\sum_{k=1}^2 \beta_k \cdot F_k(\mathbf{y}^*) \geq \sum_{k=1}^2 \beta_k \cdot F_k(\mathbf{x}^*) \text{ и } F_1(\mathbf{y}^*) \geq F_1(\mathbf{x}^*) - \Delta_1,$$

то найденное решение соответствует выбранной стратегии.

На следующем этапе с использованием функции цели $F_2(x)$ и уступки Δ_2 определяем второе дополнительное ограничение,

$$\sum_{i=1}^m c_{2i} x_i \leq \Delta_2,$$

и модифицируем систему ограничений (3) и т.д.

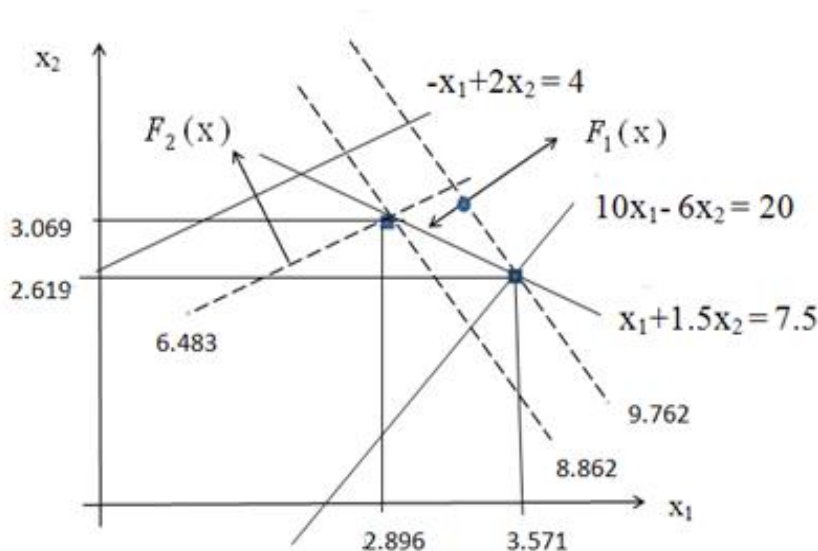
Рассмотрим пример реализации процедуры средствами математического пакета Mathcad. Предположим, что

$$F_1(x) := \sum_{k=1}^2 c_{1,k} x_k, \quad F_2(x) := \sum_{k=1}^2 c_{2,k} x_k, \quad \beta_1 = 0.6, \quad \beta_2 = 0.4, \quad \Delta_1 = 0.9.$$

Система ограничений имеет вид:

$$\mathbf{M} = \begin{bmatrix} 1 & 1.5 \\ 10 & -6 \\ -1 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 7.5 \\ 20 \\ 4 \end{bmatrix}, \quad x_1 = 0, \quad x_2 = 0.$$

Геометрическая интерпретация постановки и решения задачи методом последовательных уступок приведена на рисунке. Область допустимых решений задачи ограничена координатными осями и прямыми линиями: $-x_1 + 2x_2 = 4$, $10x_1 - 6x_2 = 20$, $x_1 + 1.5x_2 = 7.5$



Геометрическая интерпретация задачи

Решаем первую оптимизационную задачу

$$\mathbf{M} \times \mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq 0, \quad \text{Maximize}(F_1, \mathbf{x}) = \begin{pmatrix} 3.571 \\ 2.619 \end{pmatrix}, \quad \mathbf{y} := \begin{pmatrix} 3.571 \\ 2.619 \end{pmatrix}. \quad (4)$$

На основе решения (4) определяем значения всех функций цели:

$$F_1(\mathbf{y}) = 9.762, \quad F_2(\mathbf{y}) = 3.333 \quad \text{и} \quad F(\mathbf{y}) = \beta_1 \cdot F_1(\mathbf{y}) + \beta_2 F_2(\mathbf{y}) = 7.190.$$

Положение прямой, соответствующей максимуму целевой функции $F_1(x)$, отмечено штриховой линией, проходящей через точку с координатами: $x_1 = 3.571$, $x_2 = 2.619$.

Выполняем преобразование целевой функции в ограничение типа неравенства с учетом, что максимальная величина уступки $\Delta_1 = 0.9$ и значение целевой

функции может быть уменьшено до величины 8.862. В результате имеем:

$$\mathbf{W} := \begin{bmatrix} 1 & 1.5 \\ 10 & -6 \\ -1 & 2 \\ -2 & -1 \end{bmatrix}, \quad \mathbf{d} := \begin{bmatrix} 7.5 \\ 20 \\ 4 \\ -8.762 \end{bmatrix}.$$

Решаем вторую оптимизационную задачу $\mathbf{W} \times \mathbf{x} \leq \mathbf{d}, \mathbf{x} \geq 0$:

$$\text{Maximize}(F_2, \mathbf{x}) = \begin{pmatrix} 2.896 \\ 3.069 \end{pmatrix}, \quad \mathbf{y}^* := \begin{pmatrix} 2.896 \\ 3.069 \end{pmatrix}. \quad (5)$$

Используя решение (5), определяем значения функций цели:

$$F_1(\mathbf{y}^*) = 8.862, \quad F_2(\mathbf{y}^*) = 6.483 \quad \text{и} \quad F(\mathbf{y}^*) = \beta_1 \cdot F_1(\mathbf{y}^*) + \beta_2 F_2(\mathbf{y}^*) = 7.910 \quad (6)$$

Положение прямой, соответствующей максимуму целевой функции $F_2(\mathbf{x})$, отмечено штриховой линией, проходящей через точку с координатами: $x_1 = 2.896, x_2 = 3.069$. Найденные значения x_1, x_2 соответствуют максимуму интегральной оценки решения многокритериальной задачи $F(\mathbf{x}) = F(\mathbf{y}) = 7.910$. Из выражений (6) также следует, что все условия многокритериальной задачи при заданных параметрах, ограничениях выполнены и найдено оптимальное решение.

В рассмотренной выше процедуре на каждом этапе предполагается использование одной и той же формы постановки задачи без уточнения начальных значений переменных $\mathbf{x} = (x_1, x_2, \dots, x_m)$ в соответствии с величиной уступки. Это становится возможным за счет преобразования функций цели в ограничения типа неравенств. В результате последовательности преобразований многокритериальная задача сводится к однокритериальной задаче. Однако, несмотря на кажущуюся простоту предложенной процедуры, компьютерная реализация этой процедуры потребует проведения дополнительных исследований по обоснованию области применения.

Список литературы

1. Новосельцев, В. И. Теоретические основы системного анализа / В. И. Новосельцев. – М. : Майор, 2006. – 592 с.
2. Подиновский, В. В. Оптимизация по последовательно применяемым критериям / В. В. Подиновский, В. М. Гаврилов. – М. : Сов. радио, 1975. – 192 с.

Е. С. Дёмин⁽¹⁾, В. В. Вяткин⁽¹⁾, В. Н. Дубинин⁽²⁾

⁽¹⁾ Швеция, Лулео, Технический университет Лулео

⁽²⁾ Россия, Пенза, Пензенский государственный университет

ИСПОЛЬЗОВАНИЕ СЕРВИСНО-ОРИЕНТИРОВАННЫХ АРХИТЕКТУР В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ НА ОСНОВЕ СТАНДАРТА ИЕС 61499

Представлен подход к встраиванию сервисно-ориентированных решений в распределенные системы управления на основе функциональных блоков стандарта ИЕС 61499. Пример использования *Web*-сервисов проиллюстрирован на примере *Pick-and-Place* манипулятора.

Введение

Большинство производителей промышленного оборудования и системные интеграторы используют модульную машинную архитектуру для увеличения гибкости и сокращения издержек проектирования. В то же время, программное обеспечение в сфере автоматизации в значительной степени остается монолитным и использует централизованное управление [1]. Повышение эффективности разработки и степени повторного использования существующих программных компонентов стало важной областью исследований в области промышленной автоматизации. На основе сервисно-ориентированной архитектуры (SOA) можно строить децентрализованные модульные системы [6,7]. Подобные системы обладают многими преимуществами, среди которых возможность быстрого реконфигурирования, высокая степень повторного использования компонентов, масштабируемость и мобильность.

Структурная модель *Pick-and-Place* манипулятора

В качестве примера для использования сервисно-ориентированной архитектуры была выбрана модель *Pick-and-Place* (PnP) манипулятора. Это имитационная модель производственной установки, включающей в себя три горизонтальных цилиндра, три вертикальных цилиндра и одну вакуумную присоску [2]. Манипулятор выполняет функцию доставки детали из одной позиции в другую. Модель PnP-манипулятора представлена на рис. 1.

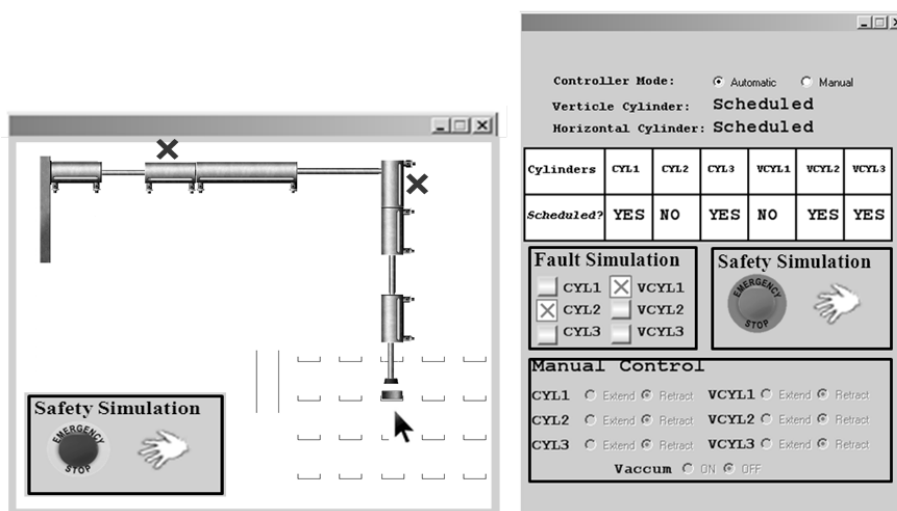


Рис. 1. Модель PnP манипулятора на основе ФБ ИЕС 61499

Управление *PnP*-манипулятором реализовано на функциональных блоках (ФБ) международного стандарта IEC 61499 [5,8] в программной среде *nxtOne* 2.0 [3] с использованием децентрализованной модели управления. К преимуществам использования данной архитектуры децентрализованного управления можно отнести быстроту развертывания и простоту обслуживания в условиях реальной работы [9]. Более подробную информацию о работе и структуре *PnP*-манипулятора можно найти в [1, 2].

Интеграция Web-сервисов с функциональными блоками стандарта IEC 61499

Для интеграции программного обеспечения сторонних производителей, необходимо наличие программного интерфейса для двухсторонней передачи данных. Программная среда *nxtOne* не имеет подобного интерфейса для Web-сервисов, однако с помощью функционального блока специального типа, называемого Composite Automation Type (CAT), можно реализовать поддержку передачи и приема данных во внешнюю программу с использованием объектно-ориентированного языка (*C/C++*, *C#*, *Java* и т.д.). Реализация подобного интерфейса необходима для получения и отправки информации из Web-сервиса.

Ограниченная поддержка языка программирования *C#* в функциональных CAT-блоках накладывает некоторые ограничения на подключение Web-сервисов. Функционал языка *C#* в CAT-блоке не позволяет выполнить вызов Web-сервиса напрямую. Для решения этой проблемы была разработана специальная динамически подключаемая библиотека (*DLL*) для организации обмена данными между Web-сервисом и CAT-блоком.

На рис.2 наглядно проиллюстрирована структура взаимодействия всех основных компонентов разработанной системы. Разработанная *DLL* библиотека использует низкоуровневую обработку *HTTP*-запросов, т.к. данный программный модуль не имеет возможности вызова функций Web-сервисов.

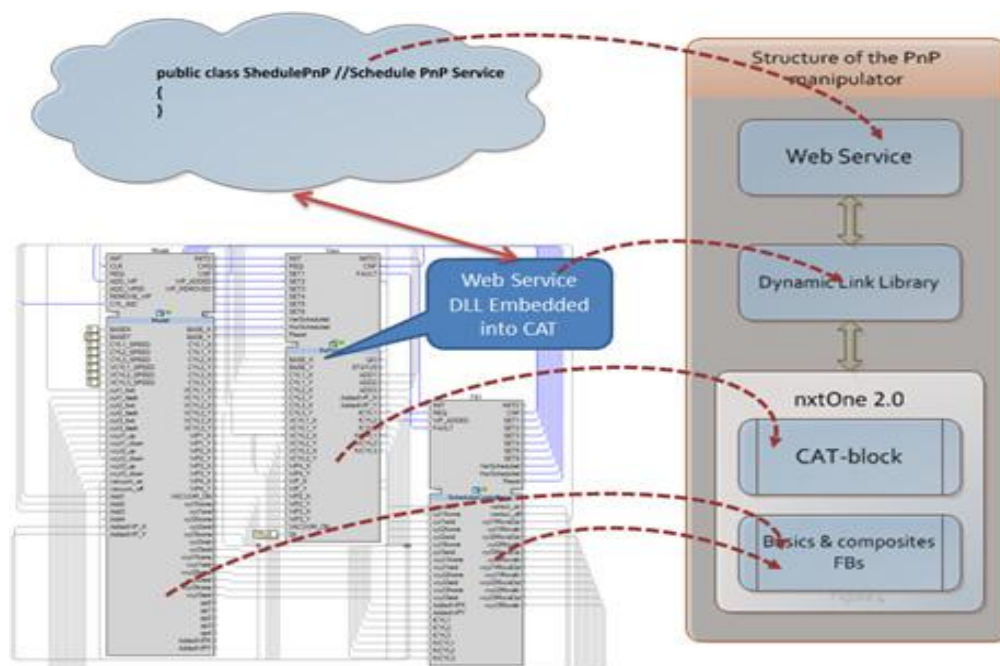


Рис. 2. Структура взаимодействия основных компонентов проекта

Ниже приведен фрагмент программы, иллюстрирующий обращение к *Web*-сервису из *CAT*-блока. Планирование работы для вертикальных и горизонтальных цилиндров выполняется отдельно. В качестве параметров передается путь к *Web*-сервису, имя *Web*-сервиса, имя функции планирования в *Web*-сервисе, координаты местоположения детали, значения длин горизонтальных или вертикальных цилиндров, значения порядковых номеров не работоспособных цилиндров.

```
namespace TestConsoleApplication {
    class Program {
        static void Main(string[] args) {
            Uri uri = new Uri("http://localhost:1577/Service1.asmx");
            WebServiceInvoker webService = new WebServiceInvoker(uri);
            string res = webService.LoadService(uri, "Service1", "ShedulePnP",
                "225:300.75.150.75");
            res += webService.LoadService(uri, "Service1", "ShedulePnP", "150:75.75.150.75");
            ...
        }
    }
    ...
}
```

Во фрагменте кода, приведенном ниже, представлена функция, обеспечивающая вызов функции *Web*-сервиса и передачу параметров в вызываемую функцию.

```
namespace DLL_Lib {
    public class WebServiceInvoker {
        ...
        public string LoadService(Uri uri, string service, string method, string args)
        {
            string result = "";
            try {
                invoker = new WebServiceInvoker(uri);
                result = invoker.InvokeMethod<string>(service, method, args);
            }
            catch (Exception ex) {
                return "Unable To Load Service: " + ex.Message
            }
            return result;
        }
        ...
    }
}
```

Следует заметить, что интерфейс *Web*-сервисов описывается при помощи специального языка описания *Web*-сервисов *WSDL* [10]. Современные средства разработки программного обеспечения позволяют генерировать данный код в соответствии со структурой разработанного *Web*-сервиса. Модуль *WSDL* представляет собой *XML*-документ стандарта 2.0 и содержит всю необходимую информацию о типах входных и выходных данных, именах функций и местоположения

функций для доступа по *URL*. Преимуществом использования языка *WSDL* является его универсальность, возможность изменения кода программистом, а так же его простота.

Ниже приведен код функции разработанной *DLL* библиотеки для импорта *WSDL*-кода. Функция импортирует *WSDL*-модуль в структуру под названием *ServiceDescriptionImporter*. Данную структуру используют функции низкого уровня вызова *Web*-сервиса, реализованные в данной библиотеке.

```
private Assembly BuildAssemblyFromWSDL(Uri webServiceUri) {  
    if (String.IsNullOrEmpty(webServiceUri.ToString()))  
        throw new Exception("Web Service Not Found");  
  
    XmlTextReader xr = new XmlTextReader(webServiceUri.ToString() + "?wsdl");  
    ServiceDescriptionImporter descriptionImporter = BuildServiceDescriptionImporter(xr);  
    return CompileAssembly(descriptionImporter);  
}
```

Реализация планирования транспортировки детали с использованием Web-сервисов

Разработанный *Web*-сервис выполняет функцию планирования движения цилиндров для доставки детали. Алгоритм планирования, основанный на предложенном в [1], является универсальным и масштабируемым. Его универсальность заключается в том, что он позволяет выполнять планирование для неограниченного количества цилиндров и координат деталей.

Для выполнения планирования *Web*-сервису необходимо передать информацию о длинах всех горизонтальных и вертикальных цилиндров в системе, координаты местоположения детали, а так же номера цилиндров, которые не могут участвовать в процессе доставки детали. Это необходимо для того, чтобы избежать при работе манипулятора некоторых нештатных ситуаций.

Подобная защита необходима в том случае, если произошла поломка и конкретный цилиндр нуждается в ремонте. Если после выполнения планирования, получается несколько вариантов доставки детали, то алгоритм выбирает наиболее оптимальный, с учетом номеров сломанных или требующих обслуживания цилиндров.

Описание функции планирования приведено ниже.

```
public class SchedulePnP //Schedule PnP Service  
{  
    ...  
    static List<bool> Schedule(int Coordinate, List<int> cylLength,  
        List<int> brokenCylinders) {  
        ...  
        //возвращает булевый массив с порядковыми номерами цилиндров, с помощью ко-  
        торых можно выполнить доставку детали  
    }  
}
```

Итеративность алгоритма придает ему гибкость и масштабируемость. К примеру, можно разделить процесс планирования на несколько параллельных задач, увеличив тем самым быстродействие работы всей системы.

Заключение

В данной работе представлен инженерный подход к интеграции технологии ФБ стандарта IEC 61499, реализованной в инструментальном средстве *nxtOne* фирмы *nxtControl*, с сервисно-ориентированными архитектурами. Подход проиллюстрирован на примере реализации функций планирования в имитационной модели *PnP*-манипулятора в виде *Web*-сервиса.

Направлением дальнейших исследований является интеграция на основе SOA технологий ФБ и баз знаний. В частности планируется разработка баз знаний систем промышленной автоматике для выполнения функций мониторинга, распознавания правильного поведения и раннего предупреждения об опасных ситуациях, диагностики и исправления неисправностей, поддержки интеллектуального интерфейса и принятия решений и т.д.

Список литературы

1. Sorouri, M. Software Composition and Distributed Operation Scheduling in Modular Automated Machines / M. Sorouri, S. Patil, V. Vyatkin // IEEE journal (under review).
2. Sorouri, M. Distributed Control Patterns for Intelligent Mechatronic Systems / M. Sorouri, S. Patil, V. Vyatkin // IEEE Conference on Industrial Informatics (INDIN), Beijing, July, 2012. – Beijing, 2012.
3. *nxtControl*. (2012). *nxtSTUDIO*. – URL: www.nxtcontrol.com
4. Vyatkin, V. IEC 61499 as Enabler of Distributed and Intelligent Automation: State of the Art Review / V. Vyatkin // IEEE Transactions on Industrial Informatics. – 2011. – No 7(4). – P. 768–781.
5. Vyatkin, V. IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design / V. Vyatkin // Instrumentation Society of America, USA, October, 2011. – P. 297.
6. Erl, T. Service-oriented architecture: concepts, technology and design / T. Erl // Prentice Hall Professional Technical Reference. – 2005. – 760 p.
7. Ruby, S. Restful Web Services / S. Ruby, D. H. Hansson // O`Reilly Media, 2007. – P. 454.
8. Lewis, R. W. Modelling control systems using IEC 61499: Applying function blocks to distributed systems: IET, 2001 / R. W. Lewis.
9. Dai, W. Function Block Implementation of Service-Oriented Architecture: Case Study / W. Dai, V. Vyatkin, J. Christensen, V. Dubinin // IEEE Conference on Industrial Informatics INDIN'2014, Porto-Allegre, Brazil, July 27-30, 2014. – Porto-Allegre, 2014.
10. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation 26 June 2007. – URL: <http://www.w3.org/TR/2007/REC-wsdl20-20070626/>.

А. Д. Адамова

Республика Казахстан, Астана, Евразийский национальный университет
им. Л. Н. Гумилева

РАЗРАБОТКА КОМПЛЕКСНОГО АЛГОРИТМА ПРИЕМА СИГНАЛОВ В ВСТРОЕННЫХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМАХ

Рассматриваются актуальные вопросы синтеза и исследования оптимальных алгоритмов приема сигналов, ориентированных на цифровую обработку сигналов. В качестве примера используется алгоритм расчета потерь при различении биполярных сигналов.

В настоящее время благодаря быстрому развитию встраиваемых информационно-управляющих систем и появлению быстродействующих и многофункциональных цифровых элементов стало возможным использование цифровой аппаратуры не только в крупных радиотехнических комплексах, но и сравнительно простой аппаратуре. При этом растет роль задач, связанная с синтезом и исследованием оптимальных алгоритмов приема сигналов, ориентированных на использование цифровой обработки.

Рассмотрим теоретический и экспериментальный анализ возможности аналого-цифрового преобразования с потерями за счет временной дискретизации наблюдения и квантования по уровням. Сигнал дискретизированный и по времени и по уровням, называется цифровым [1]. Пусть на вход временного дискретизатора в течении времени T воздействует случайный процесс

$$\xi(t) = s(\theta) + n(t), \quad 0 \leq t \leq T, \quad (1)$$

Где $s(\theta)$ – полезный постоянный сигнал, который в зависимости от информационного параметра θ может принимать на интервале $[0; T]$ положительную или отрицательную полярность: $s(\theta) = s$ при $\theta = 1$ и $s(\theta) = -s$ при $\theta = 0$ [2]. Значения случайного параметра $\theta = \{0, 1\}$ полагаются равновероятными. Шумовой процесс $n(t)$ предполагается гауссовским стационарным с нулевым математическим ожиданием и корреляционной функцией $R(\tau) = D \exp(-\alpha|\tau|)$, $D = \alpha N_0 / 4$, где N_0 – односторонняя спектральная плотность БГШ, формируемого процесса $n(t)$.

Во временном дискретизаторе периодически, в моменты времени $t_v = t_{v-1} + \Delta$, берутся выборки из реализации случайного процесса (1): $\xi_v = s(\theta) + n_v$. По принятой реализации $\xi_1^N = \{\xi_1, \dots, \xi_N\}$ требуется получить оценку параметра θ . Здесь $N = [T / \Delta]$ – число отсчетов на интервале наблюдения.

Алгоритм оценки по критерию минимума вероятности полной ошибки различения имеет вид

$$\theta = \max_B^{-1} \left\{ \ln p(\xi_1^N | \theta) \right\} \quad (2)$$

Здесь $p(\xi_1^N | \theta)$ – функция правдоподобия, которую удобно представить следующим образом:

$$p(\xi_1^N | \theta) = \frac{1}{(2\pi D)^{N/2} (1-\beta^2)^{\frac{N-1}{2}}} \exp \left\{ -\frac{(1-\beta^2)(\xi_1 - s(\theta))^2 + \sum_{v=2}^N [\xi_v - s(\theta) - \beta(\xi_{v-1} - s(\theta))]^2}{2D(1-\beta^2)} \right\}, \quad (3)$$

где $\beta = \exp(-\alpha\Delta)$ – коэффициент корреляции между соседними шумовыми отсчетами.

Подставив логарифм функции правдоподобия (3) в (2), упрощая, получим

$$\theta = \max_B^{-1} \left\{ (1-\beta^2) \xi_1 s(\theta) + \sum_{v=2}^N (\xi_v - \beta \xi_{v-1}) (1-\beta) s(\theta) \right\}$$

Отсюда получаем следующее правило различения:

$$l_N = (1+\beta) \xi_1 + \sum_{v=2}^N (\xi_v - \beta \xi_{v-1}) \begin{matrix} \theta=1 \\ > \\ < \\ \theta=0 \end{matrix} \quad (4)$$

Определенная статистика l_N в (4) имеет нормальное распределение и ее условные математические ожидания $M\{l_N | \theta\} = s(\theta)[N(1-\beta) + 2\beta]$ при различных гипотезах θ противоположны, а дисперсии $D\{l_N | \theta\} = D(1+\beta)[N(1-\beta) + 2\beta]$ равны между собой, то полная вероятность ошибки различения

$$\tilde{p}_e = 1 - \Phi(\sqrt{Q_d}), \quad (5)$$

где

$$Q_d = \frac{(M\{l_N | \theta\})^2}{D\{l_N | \theta\}} = \frac{s^2[N(1-\beta) + 2\beta]}{D(1+\beta)} \quad (6)$$

Параметр Q_d можно трактовать как отношение сигнал-шум в рассматриваемой задаче различения.

Для сравнения точности алгоритмов различения в непрерывном и дискретном времени осуществим в (6) переход к непрерывному времени, устремив $N \rightarrow \infty$ при постоянном времени наблюдения $T = N\Delta$:

$$Q_c = \lim_{N \rightarrow \infty} Q_d = \frac{s^2}{D} \lim_{N \rightarrow \infty} \frac{N[1 - \exp(-\alpha T / N)] + 2 \exp(-\alpha T / N)}{1 + \exp(-\alpha T / N)} = \frac{2s^2 T}{N_0} \left(\frac{2}{\alpha T} + 1 \right). \quad (7)$$

Ухудшение точности алгоритма различения в дискретном времени по отношению к аналоговому алгоритму можно характеризовать коэффициентом энергетических потерь:

$$\chi = \frac{Q_c}{Q_d} = \frac{(1 + 2/\alpha T)[1 + \exp(-\alpha T / N)]}{\left(\frac{4}{\alpha T}\right) \exp\left(-\frac{\alpha T}{N}\right) + (2/\alpha T N^{-1})[1 - \exp(-\alpha T / N)]}. \quad (8)$$

Графики зависимости потерь χ от числа отсчетов N , построенные для различных значений αT (αT – число интервалов корреляции шума на интервале наблюдения) приведены на рис. 1.

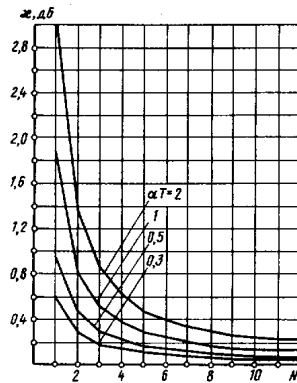


Рис. 1. Зависимость потерь от числа отсчетов при различных значениях αT

Рассчитаем дополнительные потери, возникающие за счет квантования отсчетов наблюдения ξ_v на два уровня (например, $q_v = 1$ при $\xi_v \geq 0$ и $q_v = 0$ при $\xi_v < 0$).

Оптимальный алгоритм различения в рассматриваемом случае состоит в вычислении условных вероятностей $p(q_1^N | \theta)$ наблюдаемой квантовой реализации $q_1^N = \{q_1, q_2, \dots, q_N\}$ для различных θ и сравнении их между собой

$$\theta = \max_B^{-1} p(q_1^N | \theta). \quad (9)$$

Если для многих реализаций каким-то образом вычислены условные вероятности $p(q_1^N | \theta)$, то полная вероятность ошибки цифрового алгоритма различения находится по формуле

$$p_e = \frac{1}{2} \sum_{q_1^N} \min_{\theta} p(q_1^N | \theta). \quad (10)$$

Здесь суммирование производится по всем возможным квантованным реализациям. Слагаемые суммы равны меньшей из двух вероятностей $p(q_1^N | 1)$ или $p(q_1^N | 0)$. Сравнив (10) с вероятностью ошибки (5) для непрерывнозначных отсчетов, можно оценить дополнительные потери при квантовании.

Функция правдоподобия может быть найдена интегрированием многомерной постоянной величины $p(\xi_1^N | \theta)$, описываемый выражением (3):

$$p(q_1^N | \theta) = \int_{\Omega_1} d\xi_1 \int_{\Omega_2} d\xi_2 \dots \int_{\Omega_N} d_1 p(\xi_1^N | \theta) d\xi_N. \quad (11)$$

Области интегрирования Ω_v , $v = \overline{1, N}$ связаны с квантованными значениями q_v следующим образом: $\Omega_v = (0, \infty)$ при $q_v = 1$ и $\Omega_v = (-\infty, 0)$ при $q_v = 0$.

Расчет функций правдоподобия (11) проводится путем математического моделирования на ЭВМ. Моделирование проводилось по одной реализации при отношении сигнал-шум в отсчете $\rho^2 = \frac{s^2}{D} = 0,1$ и различных значениях $\alpha\Delta$. На рисунке 2 приведены результаты моделирования – зависимость потерь η за счет квантования от $\alpha\Delta$, αT и N .

При больших $\alpha\Delta$ (независимые отсчеты) расчеты хорошо совпадают с теоретическими результатами моделирования, согласно которым потери при увеличении числа отсчетов N носят осциллирующий характер.

Характерно, что при уменьшении αT (увеличение корреляции отсчетов) осцилляции потерь сглаживаются. При этом сами потери уменьшаются. Следовательно, некоррелированность шума является самым неблагоприятным случаем с точки зрения погрешностей квантования и известные для этого случая результаты можно рассматривать как верхнюю границу потерь.

Общая концепция практической реализации предлагаемого решения для аналого-цифрового преобразования с потерями за счет временной дискретизации наблюдения и квантования по уровням представлена на структурной схеме (рис. 2).

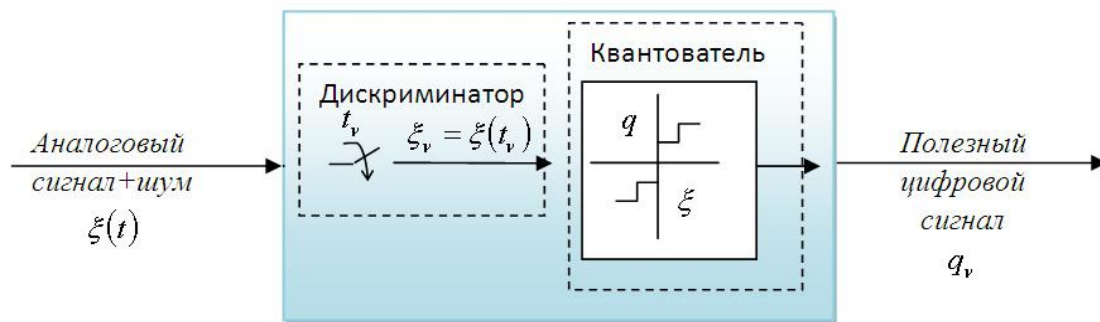


Рис. 2. Структурная схема базовой концепции устройства

Список литературы

1. Advanced digital signal processing and noise reduction / S. V. Vaseghi. – 3rd ed. – Wiley, 2006.
2. Шрюфер, Е. Обробка сигналів. Цифрова обробка дискретизованих сигналів / Е. Шрюфер. – Київ : Либідь, 1992. – 294 с.

N. S. Belinskaya

Russia, Tomsk, Tomsk Polytechnic University

**DETERMINATION OF THE REFLUX AMOUNT
IN THE STABILIZATION COLUMN OF HYDRODEWAXING
UNIT USING HYSYS MODELLING ENVIRONMENT**

In this work the model of the stabilization process of unstable hydrogenate (product of diesel fuels hydrodewaxing process) developed among HYSYS. The variation of reflux in stabilization column was calculated. Calculations allowed defining the optimal reflux flow into column in order to obtain stable product with low content of hydrogen sulphide.

The process of diesel fuels hydrodewaxing is used for production of high-quality diesel fuels with improved low-temperature properties, which allows their use in severe climatic conditions of the northern regions of the Russian Federation [1]. In addition to obtaining diesel fuels a gasoline fraction which has a low octane number is received as a desired product and can be directed for further processing in a reforming process to increase its octane number [2]. Currently, however, gasoline fraction can not be involved in the processing in the reforming because of high hydrogen sulphide content, since sulfur compounds deactivate the Pt catalyst of the process.

The reason for the high hydrogen sulphide content in the gasoline fraction is that all of the hydrogen sulphide is concentrated in the gasoline fraction on the stable hydrogenate rectification stage due to insufficient separation on the stage of stabilization of unstable hydrogenation (the product of the hydrodewaxing process). Unstable hydrogenate and reflux come in stabilization column, at the top of the column hydrocarbon gas with hydrogen sulfide go out, and at the bottom of the column stable hydrogenate goes out.

In this paper we developed a model of the process of stabilization of unstable hydrogenate among HYSYS modelling environment. The variation of reflux in stabilization column was calculated. The studies revealed that increasing the separation of hydrogen sulphide at the stage of stabilization is possible by increasing the flow of reflux in the stabilization column.

The initial data is the fractional composition of unstable hydrogenate and column products, the technological conditions of operation of the column, dimensions of the column.

The results showed that by increasing the flow of reflux, there is a change in the chemical composition of the product leaving the column due to the fact that the mass transfer is increased at the top tray of the column and feed gas components (hydrocarbons to C₄), repeatedly subjected to thermal stresses and pass into the gaseous fraction, thus increasing the upward vapor flow.

This increases the yield of hydrocarbon gas from the stabilization column (Fig. 1).

Increasing the rising vapor flow, not only reduces the amount of light hydrocarbons at the bottom of the column, but also reduces the amount of hydrogen sulphide in the stable hydrogenate (Fig. 2).

Thus, increasing the flow of irrigation stabilization column hydrogenate unstable to 60 m³/h will reduce hydrogen sulphide content in a stable hydrogenation to 0.00001%, thus obtaining gasoline, which can be further processed in a reforming pro-

cess to produce high octane gasoline.

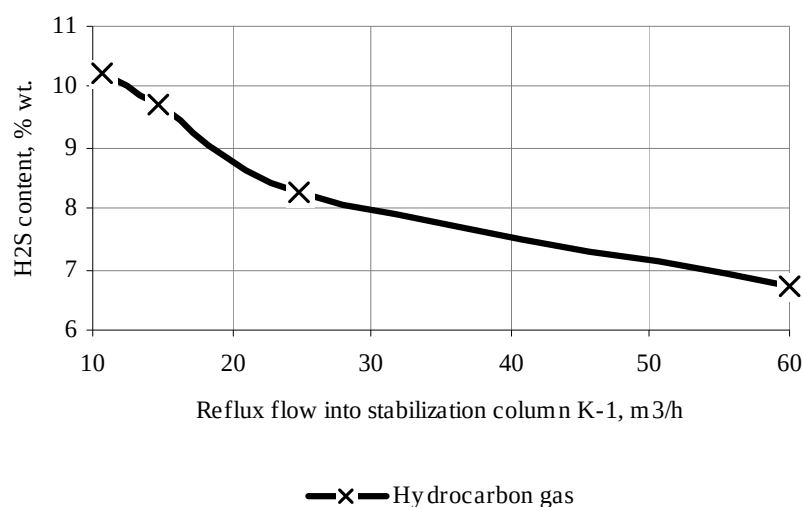


Fig. 1. The H₂S content in the hydrocarbon gas depending on the reflux flow

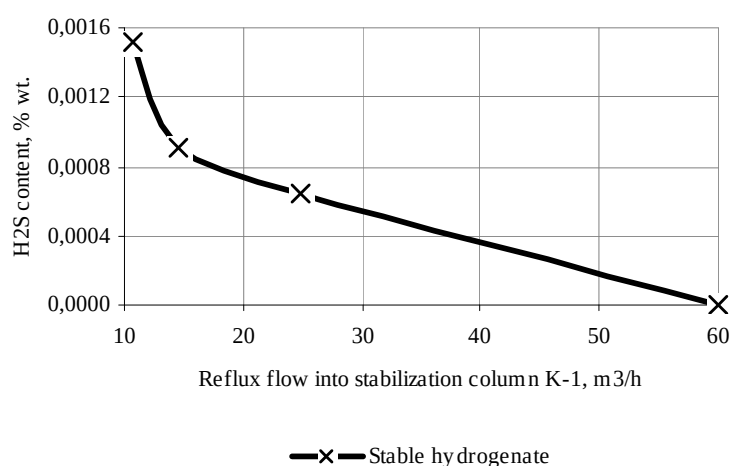


Fig. 2. The content of H₂S in the hydrogenation stable depending on reflux flow

References

1. Development of formalizing hydrocarbons conversion scheme and kinetic model of diesel fuel hydrodewaxing process / N. S. Belinskaya, G. Y. Silko, E. V. Frantsina, E. N. Ivashkina, E. D. Ivanchina // Bulletin of The Tomsk Polytechnic University. – 2014. – № 3 (322). – P. 129–133.
2. Optimization of hydrocarbon composition of raw material on reforming and hydrodewaxing units by mathematical modelling method / S. A. Faleev, N. S. Belinskaya, E. D. Ivanchina, E. N. Ivashkina, E. V. Frantsina, G. Y. Silko // Petroleum Processing and Petrochemistry. Science-technical achievements and best practice. – 2013. – № 10. – P. 14–18.

МОДЕЛЬНОЕ ПРОГНОЗИРУЮЩЕЕ УПРАВЛЕНИЕ

Предлагается метод модельного прогнозирующего управления, приводится его применение при построении модели разгона центрифуги.

Введение

Как было упомянуто в [1], все реальные проблемы подчинены различного типа ограничениям. Наиболее общими являются ограничения исполнительного механизма (такие как ограничения амплитуды и скорости нарастания). Кроме того, многие задачи имеют также ограничения на переменные состояния (например, максимальные давления, которые нельзя превышать, минимальные уровни резервуаров и т.п.). Во многих задачах проектирования эти ограничения могут игнорироваться, по крайней мере, на начальной стадии проектирования. Однако в других задачах эти ограничения – неотъемлемая часть формулировки общей проблемы, поскольку система работает вблизи ограничений. Действительно, во многих задачах управления производственным процессом оптимальная рабочая точка установившегося состояния часто находится на границе рабочей области. В подобных случаях желательно при проектировании включать ограничения с самого начала. В [1] описаны методы, основанные на стратегиях противонакопления. Однако в более сложных задачах, особенно имеющих входные ограничения и ограничения состояния, часто желательно иметь более формальный механизм обращения с ограничениями при проектировании систем управления. Будет описан здесь один такой механизм, основанный на модельном прогнозирующем управлении (МПУ). Это фактически было главным успехом на практике проектирования современных управляющих систем.

I. Этапы развития модельного прогнозирующего управления

Метод изначально возник в промышленности как реакция на потребность учета ограничений. Посвященная этому вопросу литература может быть разделена на четыре поколения следующим образом: **первое поколение** (1970-е гг.) – импульсные и линейные модели с реакцией на ступеньку, с квадратичной функцией стоимости и для специальной обработки ограничений; **второе поколение** (1980-е гг.) – линейные пространственные модели состояний, квадратичная функция стоимости, входные и выходные ограничения, выражаемые в виде линейных неравенств, и квадратичное программирование, применяемое для решения задач оптимизации управления с ограничениями; **третье поколение** (1990-е гг.) – различные уровни ограничений (мягкие, жесткие, упорядоченные), механизмы выявления неосуществимых решений; **четвертое поколение** (конец 1990-х гг. – 2014) – гарантированная устойчивость и робастные модификации.

II. Модельное прогнозирующее управление в MATLAB

Для того, чтобы применить метод МПУ к системе, закодированной в MATLAB, необходимо:

1. Представить систему дифференциальных уравнений исследуемой системы в матричной форме (state-spacemodel).
2. Собрать в MATLAB Simulink схему МПУ (пример на рис. 1).

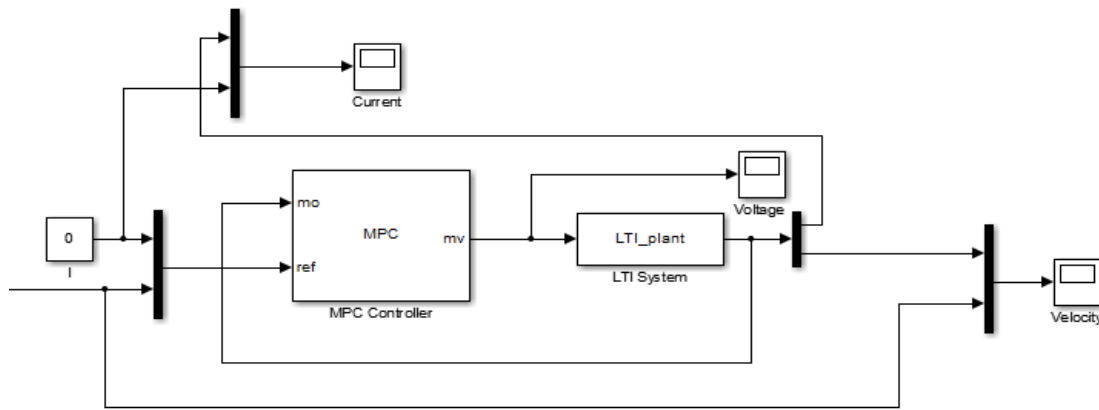


Рис. 1. Схема МПУ

На входы слева подаются желаемые характеристики.

3. С помощью набора инструментов МПУ (ModelPredictiveControlToolbox; см. рис. 2) спроектировать контроллер МПУ:

3.1. Импортировать из рабочей области MATLAB модель, созданную в п. 1. (см. рис. 3).

3.2. Ввести параметры сигналов.

3.3. Указать необходимые ограничения, параметры контроллера, вес сигналов и т.п.

III. Пример применения МПУ в среде MATLAB

Рассмотрим применение МПУ при проектировании привода центрифуги. В первую очередь составим модель, как сказано в п. 1 предыдущего раздела. Система дифференциальных уравнений выглядит следующим образом:

$$\begin{aligned} \frac{dI}{dt} &= \frac{u - IR - c\omega}{L} \\ \frac{d\omega}{dt} &= \frac{CI - M_C}{J} \end{aligned} \quad (1)$$

Запишем эту систему в матричной форме:

$$\begin{aligned} A_X &= \begin{bmatrix} \frac{-R}{L} & \frac{-C}{L} \\ \frac{C}{J} & 0 \end{bmatrix} \\ B_X &= \begin{bmatrix} \frac{1}{L} & 0 \end{bmatrix} \end{aligned} \quad (2)$$

Модель на языке MATLAB будет выглядеть следующим образом:

$L = 0.01;$

$R = 1;$

$C = 2;$

$M_C = 0;$

$J = 0.01;$

$A_x = [-R/L \ -C/L; C/J \ 0];$

$B_x = [1/L; 0];$

```

Cx1 = [1 0];
Cx2 = [0 1];
Dx1 = [0];
Dx2 = [0];
LTI_plant = ss(Ax,Bx,[Cx1;Cx2],[Dx1;Dx2]);

```

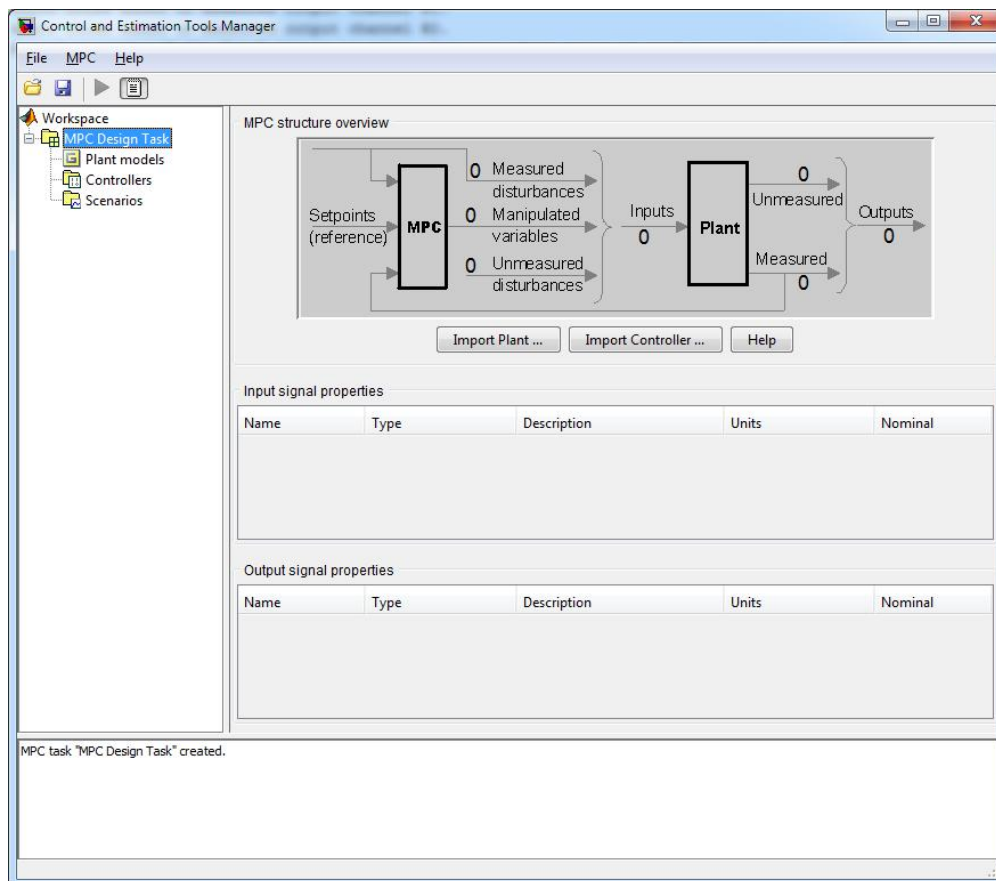


Рис. 2. Model Predictive Control Toolbox

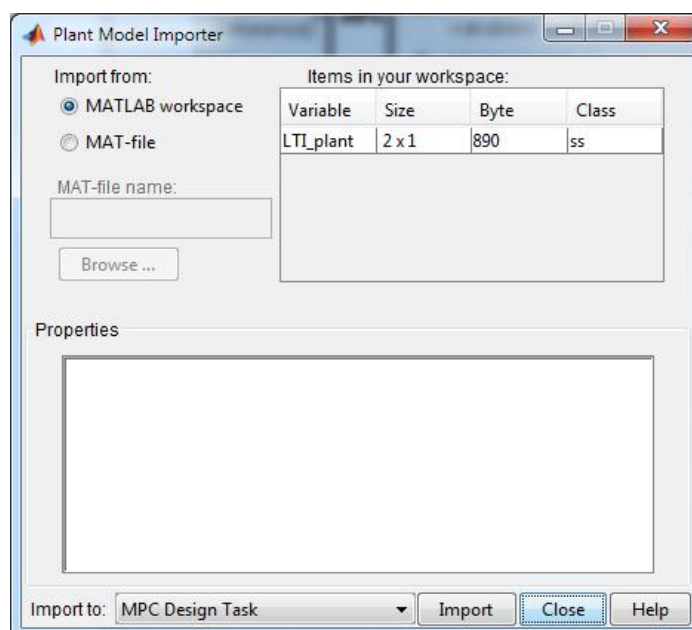


Рис. 3. Импортируем модель...

Схема в Simulink имеет вид, как показано на рис. 1. Далее настраиваем контроллер МПУ, как показано в п. 3. На рис. 4 приводятся графики, из которых следует, что метод МПУ имеет более высокую точность, чем «традиционный». На рис. 6 приводятся графики зависимости ошибки по времени.

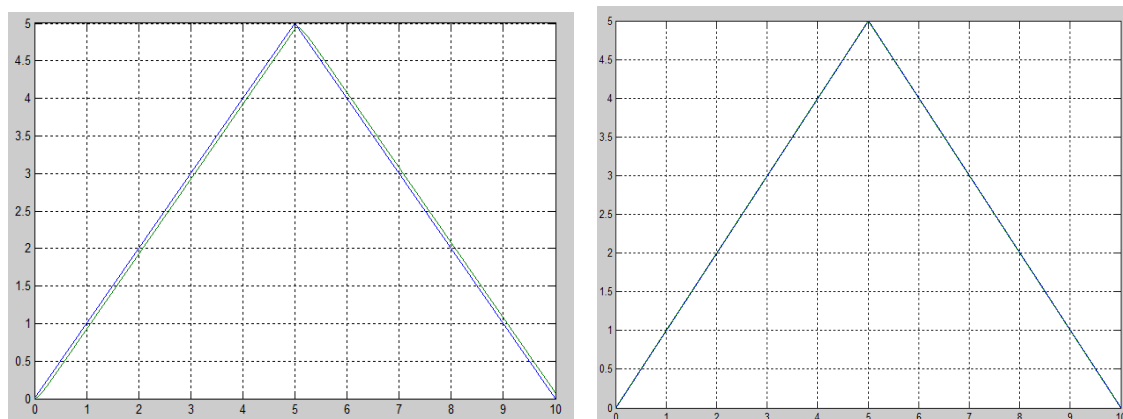


Рис. 4. Результаты «традиционного» метода (слева) и метода МПУ (справа)

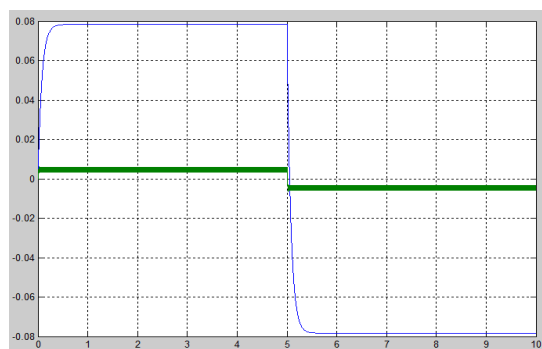


Рис. 5. Графики ошибки при использовании различных методов:
синий график – «традиционный», зеленый – МПУ

Как сказано выше, метод МПУ имеет более высокую точность, чем «традиционный». В дальнейшем планируется усложнить модель привода центрифуги путем добавления дополнительной нагрузки.

Список литературы

1. Гудвин, Г. К. Проектирование систем управления / Г. К. Гудвин, С. Ф. Гребе, М. Э. Сальгадо. – М. : БИНОМ, 2004.
2. Герман-Галкин, С. Г. Компьютерное моделирование полупроводниковых систем в MATLAB 6.0 / С. Г. Герман-Галкин. – СПб. : Корона Принт, 2001.
3. Александров А. Г. Оптимальные и адаптивные системы / А. Г. Александров. – М. : Высш. шк., 1989.
4. Ключев, В. И. Теория электропривода / В. И. Ключев. – М. : Энергоатомиздат, 2001.
5. Черных, И. В. Среда создания инженерных приложений Simulink / И. В. Черных. – М. : ДИАЛОГ-МИФИ, 2004.
6. Веремей, Е. И. Линейные системы с обратной связью / Е. И. Веремей. – СПб. ; М. ; Краснодар : Лань, 2013.
7. Ануфриев, И. Е. MATLAB 7. Наиболее полное руководство / И. Е. Ануфриев, А. Б. Смирнов, Е. Н. Смирнова. – СПб. : БХВ-Петербург, 2005.

Л. В. Гурьянов, И. В. Авдонин, Б. С. Башев

Россия, Пенза, Пензенский государственный университет

ИСПОЛЬЗОВАНИЕ CAN ШИНЫ ДЛЯ УПРАВЛЕНИЯ СОСТОЯНИЕМ АВТОМОБИЛЯ

Рассмотрены возможности CAN шины для комплексного управления и диагностики состояния автомобиля. Предложена топология системы управления автомобилем.

Бортовая электроника современного автомобиля в своем составе имеет большое количество исполнительных и управляющих устройств, для обмена информацией между которыми требуется надежная коммуникационная сеть.

Компанией BOSCH была предложена концепция сетевого интерфейса CAN (ControllerAreaNetwork).

CAN является синхронной шиной с типом доступа CollisionResolving (CR, разрешение коллизии), который в отличие от CollisionDetect (CD, обнаружение коллизии) сетей (Ethernet) детерминировано (приоритетно) обеспечивает доступ на передачу сообщения, что особо ценно для промышленных сетей управления (fieldbus). Передача ведется кадрами. Полезная информация в кадре состоит из идентификатора длиной 11 бит (стандартный формат) или 29 бит (расширенный формат, надмножество предыдущего) и поля данных длиной от 0 до 8 байт. Идентификатор говорит о содержимом пакета и служит для определения приоритета при попытке одновременной передачи несколькими сетевыми узлами [1].

CAN шина обеспечивает подключение любых устройств автомобиля, которые могут одновременно принимать и передавать цифровую информацию (дуплексная система). Собственно шины представляет собой витую пару. Данная реализация шины позволила снизить влияние внешних электромагнитных полей, возникающих при работе двигателя и других систем автомобиля. По такой шине обеспечивается достаточно высокая скорость передачи данных.

Обмен информацией идет в обоих направлениях, т.е. можно не только включить, например лампочку заднего хода, но и получить информацию светит ли она.

Возможности доступа CAN шины современного автомобиля

- электронный блок управления двигателем;
- электронный блок управления КПП;
- блок управления подушками безопасности;
- электронный блок управления ABS;
- блок управления электроусилителя руля;
- блок управления ТНВД;
- центральный монтажный блок;
- электронный замок зажигания;
- датчик угла поворота рулевого колеса;
- комбинация приборов;
- электронные блоки дверей;
- электронный блок контроля парковочной системы;
- блок управления стеклоочистителей;
- контроль давления в шинах.

CAN шина информационно-командной системы

- комбинация приборов;
- система звуковоспроизведения;
- информационная система;
- навигационная система.

Преимущества CAN шины

- возможность работы в режиме жесткого реального времени;
- простота реализации и минимальные затраты на использование;
- высокая устойчивость к помехам;
- арбитраж доступа к сети без потерь пропускной способности;
- надежный контроль ошибок передачи и приема;
- широкий диапазон скоростей работы;
- большое распространение технологии, наличие широкого ассортимента продуктов от различных поставщиков.

Недостатки CAN шины

- основным недостатком является не обязательное подключение всех бортовых устройств и датчиков. Количество доступных для управления устройств варьируется от производителя, марки автомобиля и его комплектации;
- небольшое количество данных, которое можно передать в одном пакете (до 8 байт);
- большой размер служебных данных в пакете (по отношению к полезным данным);
- отсутствие единого общепринятого стандарта на протокол высокого уровня, однако же, это и достоинство. Стандарт сети предоставляет широкие возможности для практически безошибочной передачи данных между узлами, оставляя разработчику возможность вложить в этот стандарт все, что туда сможет поместиться. В этом отношении CAN шина подобна простому электрическому проводу. Туда можно «затолкать» любой поток информации, который сможет выдержать пропускная способность шины. Известны примеры передачи звука и изображения по шине CAN (Россия). Известен случай создания системы аварийной связи вдоль автодороги длиной несколько десятков километров (Германия). (В первом случае нужна была большая скорость передачи и небольшая длина линии, во втором случае – наоборот). Изготовители, как правило, не афишируют, как именно они используют полезные байты в пакете.

Использование CAN для управления состоянием автомобиля

К главным преимуществам CAN шины необходимо отнести возможность проведения быстрой и достаточно простой диагностики всей системы автомобиля. Шина имеет единый разъем, который позволяет подключить различные устройства для диагностики.

В зависимости от доступности блоков управления автомобиля при помощи CAN шины можно обеспечить:

- контроль топливной системы автомобиля;
- дублирование информации приборной панели (обороты двигателя, скорость, пробег, температура масла и т.д.);
- диагностику ошибок двигателя и подсистем автомобиля;
- контроль состояния автомобиля (наличие ключа зажигания, состояние дверей, стеклоподъемников и т.д.).

Анализируя данные с CAN шины можно диагностировать не только неполадки внутренних систем, но и глобальные события, такие как, например, угон.

Современные системы автосигнализации производятся с интегрированным CAN модулем, что позволяет подключать их непосредственно к цифровой шине автомобиля CAN.

Система управления автомобиля с интегрированным CAN модулем получает в цифровом виде информацию о состоянии концевиков, замков, зажигания, ручника, тахометра и т.д. а также может управлять замками дверей, стеклоподъемниками, люком, штатной сигнализацией и некоторыми другими системами автомобиля.

Установка подобной системы с CAN модулем значительно сокращает вмешательство в штатную электропроводку (подключения производятся всего к 6-8 проводам, вместо 15-20 в варианте без использования CAN шины) и позволяет избежать проблем с гарантией на электрооборудование автомобиля.

На рис. 1 показана топология системы контроля состояния автомобиля подключенная через CAN шину.

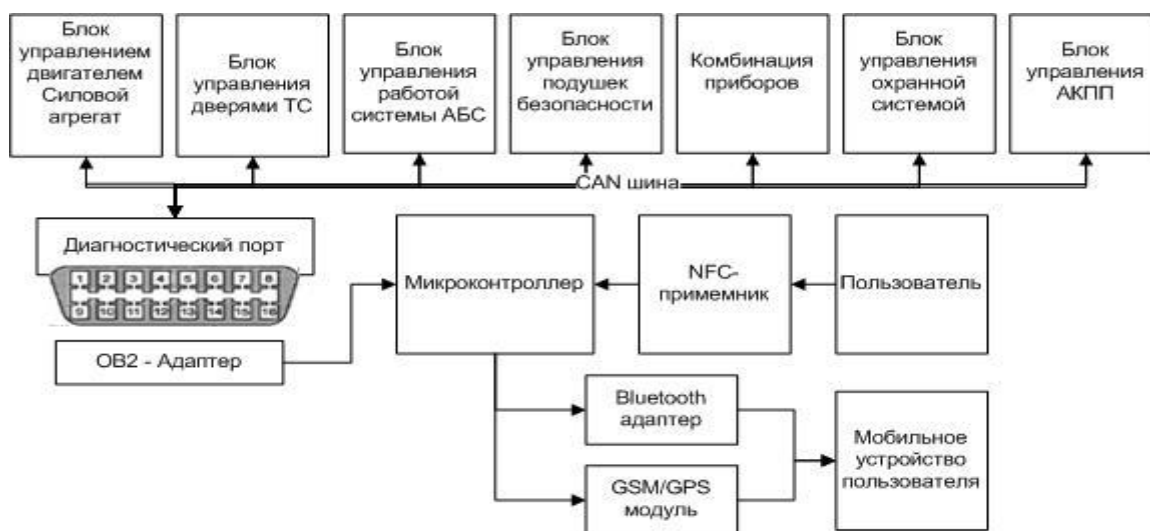


Рис. 1. Топология системы контроля состояния автомобиля

Встроенная в управляющий микроконтроллер интеллектуальная система обеспечивает контроль и диагностику состояния автомобиля, а так же реагирует на ряд критических событий. Так, например, при старте двигателя, без сигнала от управляющего устройства, двигатель будет глушиться. При использовании GSM модуля можно отслеживать местонахождение машины (например, в случае ее угона или эвакуации на штрафстоянку).

Таким образом, использование CAN шины позволяет обеспечить не только считывание данных со встроенных систем автомобиля, но так же прямое управление ими, позволяя создавать комплексные системы контроля состояния автомобиля.

Список литературы

1. Грибов, И. В. Сетевые технологии сап. Обзор инструментальных средств / И. В. Грибов // Электроника: наука, технология, бизнес. – 2011. – № 4.

D. Drozdov^(1, 2), V. Dubinin⁽¹⁾, S. Patil⁽²⁾, V. Vyatkin^(2, 3)

⁽¹⁾ **Russia, Penza, Penza State University**

⁽²⁾ **Sweden, Luleå, Luleå University of Technology**

⁽³⁾ **Finland, Helsinki, Aalto University**

REFACTORING SMV MODELS BY REPLACING ASYNCHRONOUS PROCESSES WITH SYNCHRONOUS ONES AND ITS APPLICATION TO PETRI NETS AND IEC 61499 FUNCTION BLOCKS DOMAINS

In some works, ways to model discrete transition systems in SMV with use of asynchronous processes were described. However, modern model-checking techniques, e.g. bounded model-checking, have more favorable complexity with synchronous models. In this paper, we propose an approach to represent asynchronous models as synchronous SMV ones (without use of processes) and demonstrate application of this approach to Petri-net model and IEC 61499 function block model.

1. Introduction

Model-checking technique for formal verification was introduced in early 1980s by Clarke and Emerson and since then it has come a long way of improvement and has proven itself effective on many real projects [1]. Model checker generates a state space with all different paths possible. Each path in the state space is equivalent to one traditional run or a single test case. Model checking enables the unsupervised automatic verification process of a system by generating this state space and identifies system failures via counterexamples (a scenario (state trace) that breaks the functional (such as safety) properties of the system). Properties to be verified are specified using many methods, including temporal logic, automata etc. For safety-critical applications, formal verification is one of the most efficient ways to prove system's correctness.

When model-checking was first introduced, it was suitable mostly for hardware verification with limited state space. Now, with growth of computers' power and state-of-art problem solving algorithms, there is a need for verification of these complex programs.

2. Asynchronous processes in SMV

Most of SMV language implementations (Cadence [2], NuSMV [3]) support asynchronous module instances, implemented with keyword '*process*'. Usage of asynchronous processes provides simple representation of concurrent tasks, but analysis of all possible execution orders together with increasing number of processes causes exponential state space growth (state space explosion [4]). This problem affects resource consumption, for example system RAM and increased verification time. Otherwise, synchronous models allow avoiding this problem and they are more suitable for bounded model checking (BMC) [5, 6] with modern SAT-solvers. Also, the latest bounded model checkers (such as *NuXMV* [7]) do not support asynchronous '*process*' instances in input language at all.

FB application SMV model, based on function blocks (FB) operational semantics, proposed in works [8, 9] uses FB instance mapping to asynchronous SMV processes. This approach provides intuitively clear FB types and instances mapping and allows avoiding the problem with access of multiple modules to one signal buffer, also it is the simplest way to model asynchronous execution discipline in SMV. Fig. shows access of

two modules to event signal buffer. Value *TRUE* of the buffer SMV variable means the event occurred and has not been read yet. So one module “puts” an event to the buffer (set *TRUE* value to the buffer SMV variable) and other module receives event (reads the variable) and resets SMV variable value to *FALSE*.

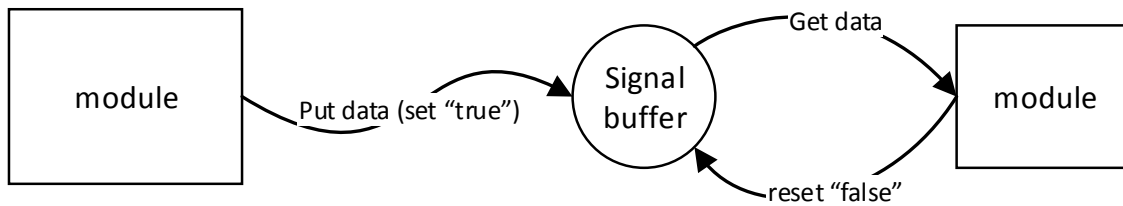


Fig. 1. Access to shared signal buffer

3. Synchronous modules instances

To build the model with synchronous module instances, we have to move all assignments for each variable to one single module. This can be achieved in two different ways. The first way to achieve this is to use additional handshake variables (Fig.). It means that all variable assignments are gathered in one of accessing modules and other modules informs this module, when they need to change the variable.

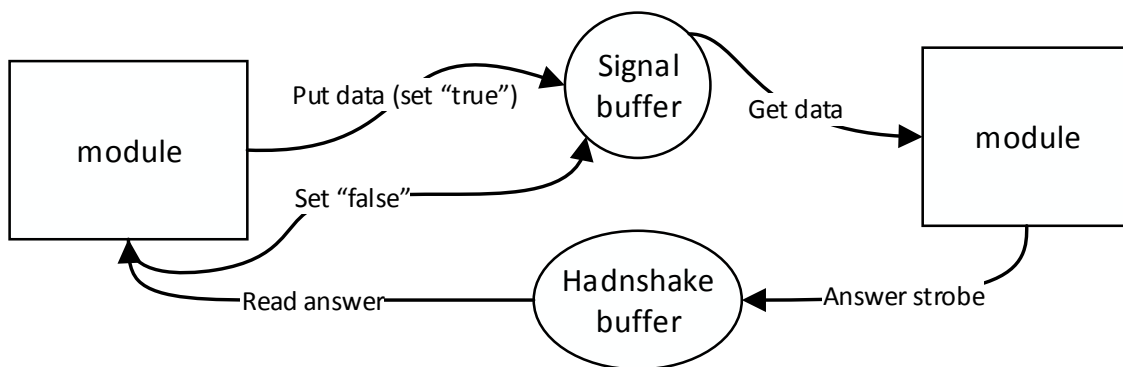


Fig. 2. Handshake buffer

This approach can be used for simple Boolean signals, which are set as ‘true’ by one module and as ‘false’ by another, but if more than two modules have an access to shared buffer or this buffer contains integer value, the interaction logic may become too complicated and cause model growth. Even with two accessing modules, this approach brings more complexity as two SMV variables are used to represent each shared buffer.

The second way is more useful (in common case) approach and is based on external buffer controller, which contains all assignments of the shared buffer. Fig. shows controller that uses module selector variable to select one of many assignment rules, collected from accessing modules. As an option, in flat models, each module may not contain its own selector, but buffer controller may use common selector variable with range $1..N$, where N is number of modules, accessing shared buffer. Selector variables may change regard to specific rules, or be a “free” variables.

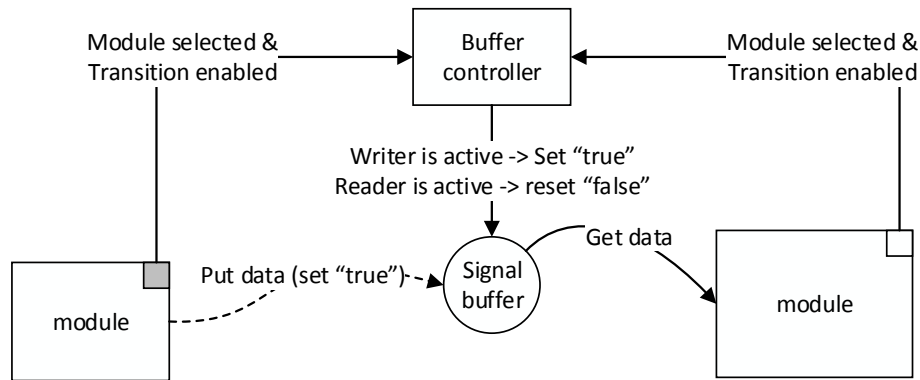


Fig. 3. External buffer controller

4. Simple Petri net example

The second approach was tested on a simple Petri-net model, which is shown in Fig..

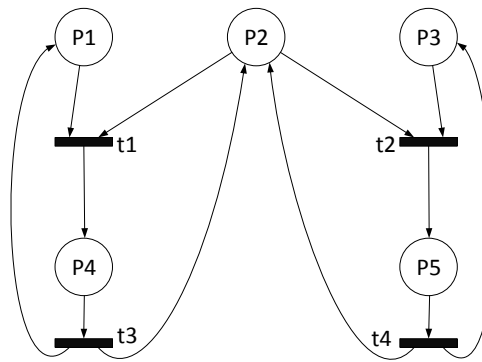


Fig. 4. Petri net example

The SMV model is based on Wimmel's model 1 ("transitions as processes") [10].

For each Petri-net place (represented by SMV variable), one controller was instantiated. Two SMV modules were created for unique "place templates" (one in- one out and two in-two out). In each module, free Boolean variable 'active' is declared. When this variable is TRUE, buffer controller selects assignment rule for active module.

The simplest 1in-1out module is shown in listing below (variable "active" is a free variable in transition module (analog of "running" variable in process)).

```
MODULE i1_o1_controller(module_i1, module_o1, p)
ASSIGN
next(p):=case
  module_i1.enabled & module_i1.active: TRUE;
  module_o1.enabled & module_o1.active: FALSE;
  TRUE: p;
esac;
```

This simple model is suitable for synchronous models like composite FB with dispatcher, but for pure asynchronous models, like Petri nets, it does not provide safety (for SMV-processes it is guaranteed that only one process is running at any moment).

In this case, SMV safety property (guarantee that only one asynchronous module instance is active at any modelling time moment) should be added in variable assignment rule.

```

MODULE i1_o1_controller(module_i1, module_o1, p, safe)
ASSIGN
next(p):=case
  module_i1.enabled & module_i1.active &safe: TRUE;
  module_o1.enabled & module_o1.active &safe: FALSE;
  TRUE: p;
esac;

```

For Petri nets SMV model, it is easier to declare SMV ‘unsafe’ property (what should never happen) and define safety as *safe*:=*(NOT unsafe)*. The ‘unsafe’ property for the given example is shown below.

```

unsafe:= (t1.active & (t2.active | t3.active | t4.active)) |
  (t2.active & (t1.active | t3.active | t4.active)) |
  (t3.active & (t2.active | t1.active | t4.active)) |
  (t4.active & (t2.active | t3.active | t1.active));

```

5. Synchronous function blocks modules in SMV

In function block SMV model, the execution order is controlled by dispatchers and no other execution selector is needed.

The problem of concurrent access to SMV variable occurs in event signal buffers, which are accessible from composite FB and it’s components. In this case, the schema, shown in Fig. may be modified and an external buffer controller may be placed in composite FB as it is shown in Fig..

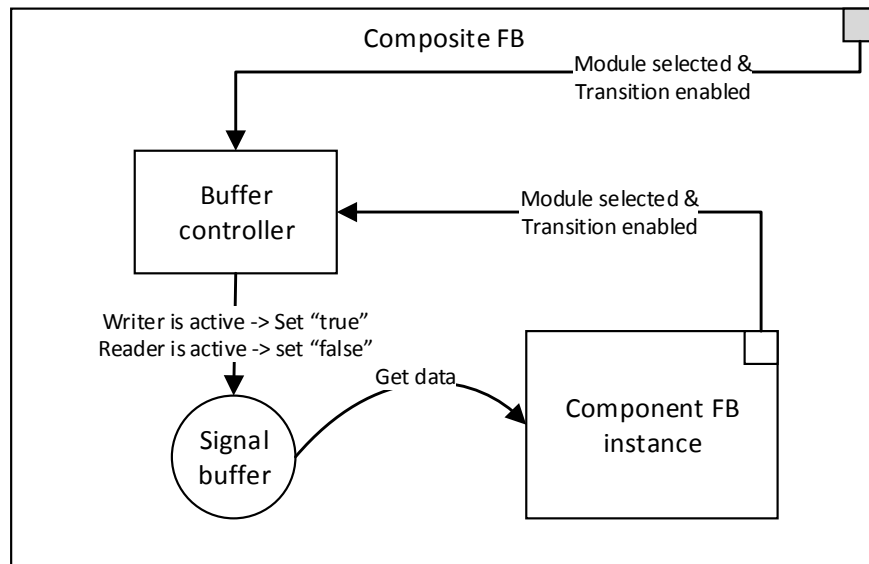


Fig. 5. Buffer controller in composite FB

Composite FB contains buffer SMV variables and it should contain buffer controllers. In each FB module, changes in event signal buffers (setting of output and resetting of input events) subjects to simple SMV ‘next’ expressions:

For component event inputs:

Component	Composite FB
<pre>next(event_signal):= case signal_reset_rule : FALSE; TRUE : event_signal; esac;</pre>	<pre>next(event_signal):= case signal_set_rule : TRUE; TRUE : event_signal; esac;</pre>

For component event outputs:

Component	Composite FB
<pre>next(event_signal):= case signal_set_rule: TRUE; TRUE : event_signal; esac;</pre>	<pre>-- unconditional reset next(event_signal) := FALSE;</pre>

The rules can be merged in composite FB and placed either in separated buffer controller modules, or in a composite FB module definition.

For component event inputs.

```
next(event_signal):= case
    signal_set_rule : TRUE;
    component.signal_reset_rule : FALSE;
    TRUE :event_signal;
esac;
```

For component event outputs:

```
next(event_signal):= case
    component.signal_set_rule : TRUE;
    module_is_active : FALSE; -- unconditional reset
    TRUE :event_signal;
esac;
```

Rule *ssignal_reset_rule* and *signal_set_rule* in component FBs are declared with DEFINE statements. *Alpha* and *beta* rules [8, 9] should be defined in components and their ‘next’ statements should be merged into composite FB dispatcher as it is shown in the listing below.

```
next(Middle_alpha):= case
    Home_beta&omega : TRUE;
    Middle.alpha_reset : FALSE;
    TRUE :Middle_alpha;
esac;
next(Middle_beta):= case
    Middle_beta&omega : FALSE;
    Middle.beta_set : TRUE;
    TRUE :Middle_beta;
esac;
```

Table 2 shows comparison of RAM usage and verification time for bounded model checking with single bounds for *PnP* model with asynchronous modules and synchronous ones.

Table 2

Bounded model checking with non-optimized and optimized model

Singleboundvalue	PnP model with async. instances		PnP model with sync. instances	
	Time	RAM (MB)	Time	RAM
20	42.93 s	2134 MB	20.37 s	1875 MB
50	> 12 hours	~7 GB	58.15 s	4721 MB
70	> 12 hours	~ 11 GB	73.18 s	5689 MB

6. Conclusion and future work

We have presented a useful approach to refactor asynchronous SMV models, e.g. Petri nets and IEC 61499 function block applications in SMV without using deprecated ‘process’ instances. This approach allows for verification of such applications using bounded model checking (BMC) with state-of-art SAT-solvers.

BMC has difference with “traditional” BDD model-checking and main of these differences is that the result of BMC is counterexample or its absence on set of paths with length k . In other words, negative result of BMC counterexample search means only that counterexample does not exist on k -paths [5]. In this case, for the approval of the specification truth, we have to determine the upper estimate for k - the maximum paths’ length in the model state space.

The future work may be oriented to optimize FB models in SMV to have predictable maximum paths’ length in model state space.

References

1. Clarke, E. M. Model checking / E. M. Clarke, O. Grumberg and D. A. Peled . – Cambridge: The MIT Press, 1999.
2. Cadence SMV Model Checker. – URL: <http://www.kenmcmil.com/smv.html>
3. Cimatti, A. NuSMV 2: An OpenSource Tool for Symbolic Model Checking / A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, et al. // Computer Aided Verification / ed. by E. Brinksma and K. Larsen. – Heidelberg : Springer Berlin, 2002. – Vol. 2404. – P. 241–268.
4. Clarke, E. Progress on the State Explosion Problem in Model Checking / E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith // Informatics / ed. by R. Wilhelm. – Heidelberg : Springer Berlin, 2001. – Vol. 2000. – P. 176–194.
5. Biere, A. Bounded Model Checking / A. Biere, E. Cimatti, M. Clarke and Y. Zhu // Advances in Computers. – 2003. – Vol. 58. – P. 117–148.
6. Clarke, E. Bounded Model Checking Using Satisfiability Solving / E. Clarke, A. Biere, R. Raimi and Y. Zhu // Formal Methods in System Design. – 2001. – Vol. 19, no. 1. – P. 7–34.
7. The nuXmv model checker. – URL: <https://nuxmv.fbk.eu>
8. Patil, S. Neutralizing Semantic Ambiguities of Function Block Architecture by Modeling with ASM / S. Patil, V. Dubinin, C. Pang, and V. Vyatkin // 9th International Andrei Ershov Memorial Conference, PSI 2014. – Peterhof, St. Petersburg, Russia, 2014.
9. Дубинин, В. Н. Модели функциональных блоков ИЕС 61499, их проверка и трансформации в проектировании распределенных систем управления : моногр. / В. Н. Дубинин, В. В. Вяткин ; под ред. проф. Н. П. Вашкевича. – Пенза : Изд-во ПГУ, 2012. – 350 с.
10. Wimmel, G. A BDD-based Model Checker for the PEP Tool / G. Wimmel // Major individual project. – University of Newcastle, 1997.

Л. А. Брякин, А. В. Кучин

Россия, Пенза, Пензенский государственный университет

ДАТЧИК ХОЛЛА НА ПОЛЕВОМ ТРАНЗИСТОРЕ

Предлагаются конструктивные изменения датчика Холла, позволяющие придать датчику новые свойства.

Датчики Холла находят широкое применение не только для измерения магнитных полей, но и для измерения механических перемещений и для целей управления в составе электродвигателей.

Известно [1, 2], что электродвижущая сила, возникающая благодаря эффекту Холла на боковых контактах датчика, пропорциональна величине магнитной индукции приложенного внешнего магнитного поля и скорости движения носителей заряда в направлении приложенного внешнего электрического поля.

Известно также [2,3], что в полевых транзисторах, выполненных на основе p – n -перехода или с использованием структуры «металл-диэлектрик-полупроводник», с повышением напряжения на стоке наблюдается отсечка тока стока, при которой вблизи стока образуется обедненная область. С увеличением напряжения на стоке ширина этой области растет, приближаясь к истоку. Особенностью этой области является то, что скорость носителей заряда в ней оказывается максимально возможной для выбранного полупроводника.

Предлагается контакты для съема напряжения Холла расположить в области отсечки полевого транзистора, что позволит гарантированно получать максимальную величину напряжения. А заданием напряжения на затворе относительно истока можно регулировать величину рабочего тока датчика. Дальнейшее увеличение напряжения Холла можно получить, если использовать предложенную на рис. 1 структуру [4].

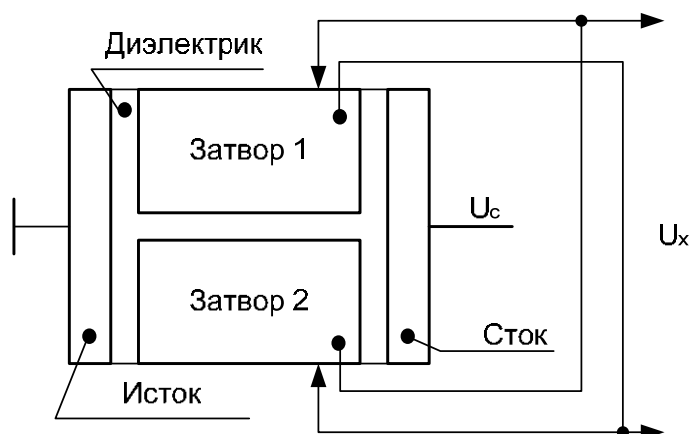


Рис. 1. Возможная топология датчика Холла

При отсутствии внешнего магнитного поля напряжение Холла $U_x = 0$, под действием напряжения стока U_c течет ток стока, затворы под одинаковым потенциалом. Внешнее магнитное поле вызывает смещение носителей заряда в канале в соответствующую сторону, появляется ненулевое напряжение на выходных контактах, ненулевая разность потенциалов на затворах приводит к дальнейшему смещению зарядов в том же направлении, что увеличивает выходное напряжение.

Список литературы

1. Китайгородский, А. И. Введение в физику / А. И. Китайгородский. – М. : Наука, Главная редакция физико-математической литературы, 1973. – 688 с.
2. Пасынков, В. В. Полупроводниковые приборы : учеб. для вузов / В. В. Пасынков, Л. К. Чиркин. – М. : Высш. шк., 1987. – 479 с.
3. Ричман, П. Физические основы полевых транзисторов с изолированным затвором / П. Ричман ; пер. с англ. под ред. Г. Г. Смолко. – М. : Сов. радио, 1971. – 144 с.
4. А.с. №1636816. Датчик для измерения магнитного поля / Л. А. Брякин. – Опубл. 23.03.91, БИ № 11.

А. В. Дудоров⁽¹⁾, В. Н. Дубинин⁽¹⁾, Е. С. Дёмин^(1,2)

⁽¹⁾Россия, Пенза, Пензенский государственный университет

⁽²⁾Швеция, Лулео, Технический университет Лулео

РАЗРАБОТКА ОНТОЛОГИИ ТЕХНОЛОГИЧЕСКИХ СХЕМ ДЛЯ СИСТЕМ АВТОМАТИЗАЦИИ

В работе представлены начальные шаги в разработке Web-онтологии технологических схем для систем автоматизации, построенных в соответствии с международным стандартом IEC 61424. Для онтологического представления технологических схем используются язык дескриптивной логики OWL DL и язык правил SWRL.

Введение

В настоящее время большое распространение получает онтологическое представление знаний [1]. В первую очередь это связано с развитием концепций семантического Web, «паутины вещей» (*Web of Things*), сервисно-ориентированных архитектур (SOA), а также мультиагентных систем. Под онтологией понимается формальное представление множества концептов внутри домена и отношений между этими концептами [1]. Наиболее известным онтологическим языком является язык OWL [2]. В качестве языка правил в семантическом Web используется язык SWRL [3], значительно расширяющий описательные возможности OWL-базируемых онтологий.

Известны следующие преимущества описаний на основе Web-онтологий: *формальность* – основа – дескриптивная логика (ДЛ) и логика хорновских дизъюнктов; *разрешимость* – процесс вывода (классификации) заканчивается за конечное время; *гибкость и расширяемость* – поддерживается концепция «открытого» мира, онтология строится по модульному принципу благодаря наличию инструкций *import*, классы и свойства являются независимыми друг от друга (в отличие от языка UML), используется декларативное представление знаний.

В мире разработано и ежегодно разрабатывается множество онтологий в самых различных предметных областях. Для примера можно привести онтологию функциональных блоков (ФБ) стандарта IEC 61499 [4,5] и онтологию языков ПЛК [6]. Web-онтологии размещаются в сети Интернет и становятся разделяемыми ресурсами. Для поиска онтологий в сети Интернет создано несколько поисковых машин, например, *Joomla* [7].

В проектировании управления технологическим процессом многие требования для системы управления условно специфицируются в технологической схеме трубопроводов и контрольно-измерительных приборов и автоматики (КИПиА) (*pipingsandinstrumentationdiagrams – P&IDs*)[8]. Для того, чтобы упростить обмен информацией между *P&ID* и средствами проектирования управления технологическим процессом, стандарт IEC 62424 [9] предписывает представление функциональных требований к системе управления технологическим процессом (называемых *PCE*-запросом) в виде диаграмм *P&ID* с использованием соответствующего языка обмена данными *CAEX* [10]. Таким образом, с использованием стандарта IEC 62424 возможно автоматизировать преобразование информационных моделей между *P&ID* и средствами автоматизации проектирования систем управления технологическими процессами. Язык *CAEX* основан на *XML* и определяется с использованием *XMLschema*. Недостатком всех *XML*-ориентированных языков является то, что они представляют только синтаксическое описание предметной области, но не семантические ограничения и аспекты выполнения. В данной работе предприняты начальные шаги по разработке онтологии *P&ID*-диаграмм, которая в определенной мере может использоваться как семантическое расширение языка *CAEX*.

Разработка онтологии

В качестве начального источника информации при создании онтологии *P&ID*-диаграмм использовалась палитра технологических схем широко известного средства *MicrosoftVisio*. В онтологическом описании инструменты палитры представлены классами, классы объединены в группы в соответствии с тем, как они объединены в палитре: *Vessels* – резервуары; *Valves* – вентили (клапаны); *Heat_Exchangers* – теплообменники; *Instruments* – датчики и контроллеры; *Pipelines* – трубопроводы; *Pumps* – насосы. Для удобства описания *Signals* – сигналы связи и *Fittings* – приборы выделены в отдельные классы. Иерархия классов, реализованная в системе *Protege 4.3* [11], показана на рис. 1.

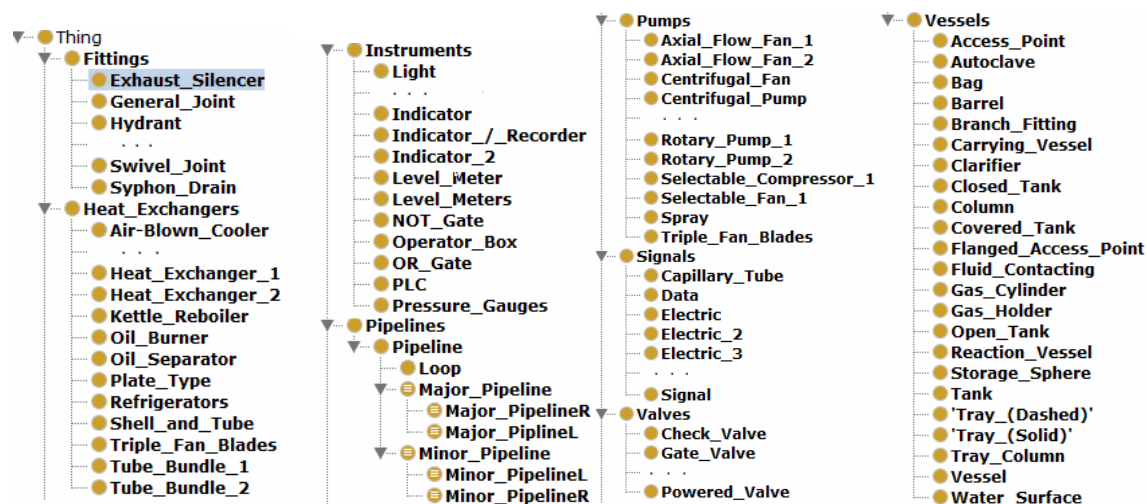


Рис. 1. Иерархия классов онтологии *P&ID*

Класс *Pipelines* включает в себя подклассы *Loop* – петля и *Pipeline* – трубопровод. *Pipeline* в свою очередь делится на подклассы *Major_Pipeline* и *Minor_Pipeline* – главный и второстепенный трубопроводы. Их можно представить в виде аксиом:

$$Major\_Pipeline \equiv Pipeline \sqcap (\exists has\_small\_size.0) \sqcap (\exists has\_big\_size.1),$$

$$Minor\_Pipeline \equiv Pipeline \sqcap (\exists has\_small\_size.1) \sqcap (\exists has\_big\_size.0).$$

По направлению потока эти классы включают в себя *Major_PipelineL*, *Major_PipelineR*, *Minor_PipelineR* и *Minor_PipelineL* соответственно.

Их описание также можно представить в виде аксиом:

$$Major\_PipelineR \equiv Major\_Pipeline \sqcap (\exists directed\_to\_the\_left.0) \sqcap (\exists directed\_to\_the\_right.1);$$

$$Major\_PipelineL \equiv Major\_Pipeline \sqcap (\exists directed\_to\_the\_left.1) \sqcap (\exists directed\_to\_the\_right.0),$$

$$Minor\_PipelineR \equiv Minor\_Pipeline \sqcap (\exists directed\_to\_the\_left.0) \sqcap (\exists directed\_to\_the\_right.1);$$

$$Minor\_PipelineL \equiv Minor\_Pipeline \sqcap (\exists directed\_to\_the\_left.1) \sqcap (\exists directed\_to\_the\_right.0).$$

Классы в онтологии могут иметь свойства по данным и объектные свойства. Объектные свойства предполагают связь объектов между собой, а свойства по данным описывают параметры объектов. Иерархия объектных свойств и свойств данных представлена на рис. 2.

Объектные свойства: *has_connection_with* – определяет соединение сигнала с источником и приемником, инверсное свойство *connected_with*; *coming_to* – определяет, откуда выходит трубопровод, инверсное свойство *has_output_pipeline*; *enters* – определяет, куда входит трубопровод, инверсное свойство *has_input_pipeline*; *has_input* – данное свойство определяет наличие входного сигнала у датчика, ПЛК или компьютера, инверсное свойство *has_transmitter*; *has_output* – определяет наличие выходного сигнала у датчика, ПЛК или компьютера, инверсное свойство *has_receiver*; *has_name* – имя объекта; *has_number_of_block* – номер блока; *has_number* – порядковый номер. Классы *Indicator*, *Indicator_2*, *PLC* и *Computer* имеют свойство *has_instrumental_type*. Данное свойство определяет предназначение этих инструментов. Классы *Indicator* и *Indicator_2* имеют свойство *remoteness* – удаленность. Класс *Pipeline* имеет свойства *has_small_size* и *has_big_size*, определяющие, какой трубопровод (главный или второстепенный) представлен, а также свойства *directed_to_left* и *directed_to_right*, определяющие направления движения потока.

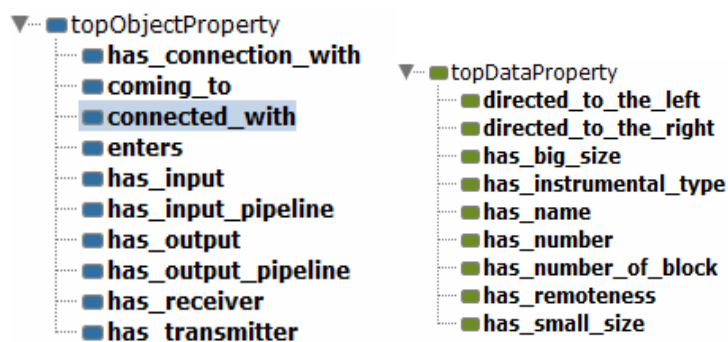


Рис. 2. Объектные свойства и свойства по данным

Система *Protégé* позволяет визуализировать онтологию в виде графа. На рис. 3 представлен фрагмент терминологической части (*T-Box*) онтологии технологических схем.

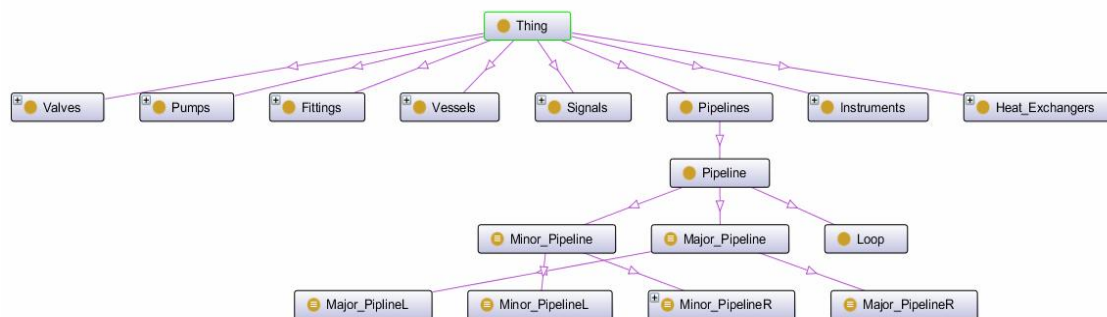


Рис. 3. Диаграмма классов онтологии.

С использованием *OWL* и *SWRL* возможно описание семантических ограничений, накладываемых на *P&ID*. Пример семантического ограничения: «трубопровод не должен образовывать кольцо». Данное семантическое ограничение может быть выражено на языке *SWRL*:

$$\begin{aligned} & Pipeline(?p) \wedge enters(?p, ?p) \rightarrow Loop(?p); \\ & Pipeline(?p) \wedge coming\_to(?p, ?p) \rightarrow Loop(?p). \end{aligned}$$

Если такой кольцевой трубопровод найдется, то он добавится в класс *Loop*.

Средство для семантического анализа онтологий *P&ID*-диаграмм

В работе [11] был предложен метод семантического анализа ФБ ИЕС 61499 с использованием *Web*-онтологий. Аналогично этому, могут быть разработаны методы и средства семантического анализа *P&ID*-диаграмм. На рисунке 4 приведена структура системы семантического анализа *P&ID*-диаграмм на основе *Web*-онтологий.

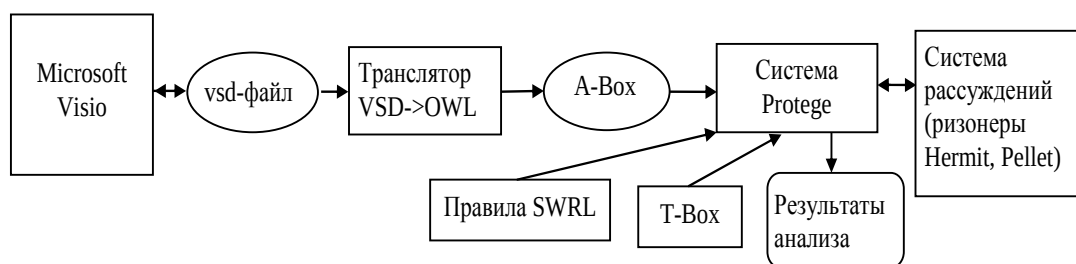


Рис. 4. Система семантического анализа *P&ID*-диаграмм

Заключение

В данной работе представлена начальная версия онтологии *P&ID*-диаграмм. На текущий момент онтология содержит 149 классов, 9 свойств по данным и 10 объектных свойств. В дальнейшем, по мере изучения новых источников информации, данная онтология будет корректироваться и расширяться, в том числе, за счет включения новых семантических ограничений. Кроме того, предполагается размещение онтологии в сети Интернет и разработка транслятора технологических схем, сформированных в редакторе *MSVisio*, в онтологическое представление (*A-Box*).

Представленная работа частично выполнена в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития разработки научно-технологического комплекса России на 2014–2020» (соглашение № 14.574.21.0045 от 19.06.2014).

Список литературы

1. Гаврилова, Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб. : Питер, 2000. – 384 с.
2. W3C Консорциум. Язык Web-онтологий OWL. – URL: <http://www.w3.org/2004/OWL/>.
3. SWRL: A Semantic Web Rule Language. W3CMemberSubmission, 2004. – URL: <http://www.w3.org/Submission/SWRL/>.
4. Дубинин, В. Н. Онтология функциональных блоков стандарта IEC 61499 / В. Н. Дубинин, В. В. Вяткин // Современные информационные технологии : тр. Междунар. науч.-техн. конф. – Пенза, 2010. – Вып. 12. – С. 113–126.
5. Дубинин, В. Н. Обнаружение циклов в системах функциональных блоков IEC 61499 с использованием онтологий / В. Н. Дубинин, В. В. Вяткин // Современные информационные технологии : тр. Междунар. науч.-техн. конф. – Пенза, 2011. – Вып.13. – С. 149–161.
6. Будаговский, Д. А. Онтология языков программирования стандарта IEC 61131-3 / Д. А. Будаговский, В. Н. Дубинин // Университетское образование (МКУО-2014) : сб. ст. XVIII Междунар. науч.-метод. конф. – Пенза, 2014. – С. 164–166.
7. SwoogleSemanticWebSearchEngine. – URL: <http://swoogle.umbc.edu/>.
8. Pang, C. IEC 61499 Based Model-Driven Process Control Engineering / C. Pang, V. Vyatkin, W. Dai // IEEE 19th International Conference on Emerging Technologies & Factory Automation. – Barcelona, Spain, 2014. – P. 1–8.
9. IEC Standard 62424. Representation of process control engineering – Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools, 2008.
10. CAEX. – URL: <http://en.wikipedia.org/wiki/CAEX>.
11. Protégé. – URL: <http://protege.stanford.edu>.
12. Дубинин, В. Н. Семантический анализ описаний систем управления промышленными процессами на основе стандарта IEC 61499 с использованием онтологий / В. Н. Дубинин, В. В. Вяткин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2010. – № 3 (15). – С. 3–15.

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Е. А. Войнова

Россия, Пенза, Пензенский государственный университет

ФОРМАЛИЗАЦИЯ ПРОЦЕССОВ УПРАВЛЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ЯЗЫКА СОБЫТИЙНЫХ НЕДЕТЕРМИНИРОВАННЫХ АВТОМАТОВ

Раскрываются некоторые особенности формализации процессов управления информационной безопасностью на основе использования языка событийных недетерминированных автоматов.

Современный уровень развития информационных технологий выдвигает на передний план новые требования к построению систем управления процессами информационной безопасности.

При проектировании современных информационных систем вопросы защиты информации должны рассматриваться в качестве ее неотъемлемой составляющей. Автоматизация организационных, технологических и производственных процессов привела к зависимости эффективности их функционирования от защищенности информационных и поддерживающих систем. Развитие современных информационных технологий приводит к все более возрастающему количеству уязвимостей элементов информационных систем, а также угроз безопасности, способных привести к нанесению злоумышленниками значительного ущерба организации.

В связи с этим, возрастает актуальность вопроса формализации процессов управления информационной безопасностью. Формализация процессов управления информационной безопасностью позволит оптимизировать процессы управления информационной безопасностью, исключить влияние различных негативных факторов, например, ошибок, вызванных человеческим фактором. Формальное описание процессов также положит основу для создания более совершенных методик и системы управления информационной безопасностью.

В настоящее время особенностью процессов управления информационной безопасностью является необходимость использования экспертных оценок, наличие большого количества неопределенностей.

При разработке и создании систем логического управления информационными процессами при реализации вычислительных задач или задач промышленной автоматизации и управления объектами формальным методом описания управляющих алгоритмов придается большое значение. Это связано с тем, что такие формальные методы должны обеспечивать комплексное решение задач спецификации, разработки, верификации, реализации и анализа сложных систем управления, в том числе систем управления рисками информационной безопасности. Математический аппарат формальных методов описания управляющих алгоритмов должен обладать при этом высокой эффективностью, связанной с повышением производительности и надежности систем управления.

Под эффективностью формального языка описания управляющих алгоритмов обычно подразумевается выполнение следующих требований:

- простота, наглядность, компактность и удобство использования;
- высокая степень выразительности при описании сложных вариантов алгоритмов;

- наличие методов оптимизационных равносильных преобразований, связанных с минимизацией, композицией, декомпозицией и верификацией управляющих алгоритмов на моделях;
- простой переход от формального аналитического представления к структурной реализации управляющего алгоритма в виде программы;
- математический аппарат должен быть ориентирован на описание параллельно-последовательных алгоритмов, допускающих одновременное выполнение нескольких событий, реализуемых в объектах управления.

Формальный метод описания управляющих алгоритмов на основе использования событийных недетерминированных автоматов удовлетворяет в достаточной степени этим требованиям.

Поэтому формализация процессов управления информационной безопасностью на основе использования языка событийных недетерминированных автоматов, нацеленная на повышение точности оценок, используемых в процессах управления информационной безопасностью, позволит автоматизировать процессы управления информационной безопасностью наиболее эффективно.

В качестве примера может быть рассмотрена формализация процесса управления проверкой нарушения свойств информации. Под информацией, свойства которой могут быть нарушены, подразумеваются информационные активы, ключевая информация и иная ценная информация. В качестве свойств информации применяются свойства целостности, конфиденциальности и доступности. Процесс управления проверкой нарушения свойств информации выглядит следующим образом. Первоначально, тестовое значение исследуемой информации, поступающее на вход рассматриваемого алгоритма, путем некоторых преобразований сравнивается с эталонным значением. Если в результате преобразований тестовое значение совпало с эталонным значением, то полученный результат сохраняется и можно сделать вывод, что проверяемые свойства исследуемой информации не нарушены. В противном случае, сохраняется результат проверки, свидетельствующий о нарушении исследуемых свойств информации.

Частные события, используемые в данном примере:

- $S_{\text{input}}(t)$ – сокращенное обозначение событий, свидетельствующих о появлении на входе алгоритма информации, свойства которой следует проверить;
- $S_{\text{test}}(t)$ – сокращенное обозначение событий, определяющих эталонные значения свойств информационной безопасности для информации, свойства которой следует проверить; $S_{\text{violation}}(t)$ – сокращенное обозначение событий, свидетельствующих о нарушении исследуемых свойств проверяемой информации;
- $S_{\text{success}}(t)$ – сокращенное обозначение событий, свидетельствующих об успешности проведенной проверки и том, что свойства исследуемой информации не нарушены; заключительное состояние;
- $S_{\text{save}}^{\text{ok}}(t)$ – сокращенное обозначение событий, обозначающих сохранение состояния, характеризующего успешное завершение проверки;
- $S_{\text{save}}^{\text{violation}}(t)$ – сокращенное обозначение событий, обозначающих сохранение состояния, характеризующего завершение проверки с обнаружением нарушения свойств исследуемой информации;
- $S_k(t)$ – окончание процесса.

Система канонических уравнений, описывающих эти события и соответствующий им граф недетерминированного автомата, будут иметь следующий вид:

$$S_{\hat{1}\emptyset\hat{e}\hat{a}\hat{e}\hat{a}}(t+1) = S_{\hat{y}\hat{o}\hat{a}\hat{e}\hat{1}\hat{i}}(t) \wedge [(S_{\hat{o}\hat{a}\hat{n}\hat{o}}(t) \wedge S_{\hat{y}\hat{o}\hat{a}\hat{e}\hat{1}\hat{i}}(t))] \vee S_{\hat{n}}^{\hat{i}}(t) \wedge S_{\hat{1}\emptyset\hat{e}\hat{a}\hat{e}\hat{a}}(t),$$

$$S_{\hat{o}\hat{n}\hat{i}\hat{a}\hat{o}}(t+1) = S_{\hat{y}\hat{o}\hat{a}\hat{e}\hat{1}\hat{i}}(t) \wedge [(S_{\hat{o}\hat{a}\hat{n}\hat{o}}(t) \wedge S_{\hat{y}\hat{o}\hat{a}\hat{e}\hat{1}\hat{i}}(t))] \vee S_{\hat{n}}^{\hat{o}}(t) \wedge S_{\hat{o}\hat{n}\hat{i}\hat{a}\hat{o}}(t).$$

Соответствующий граф приведен на рис. 1.

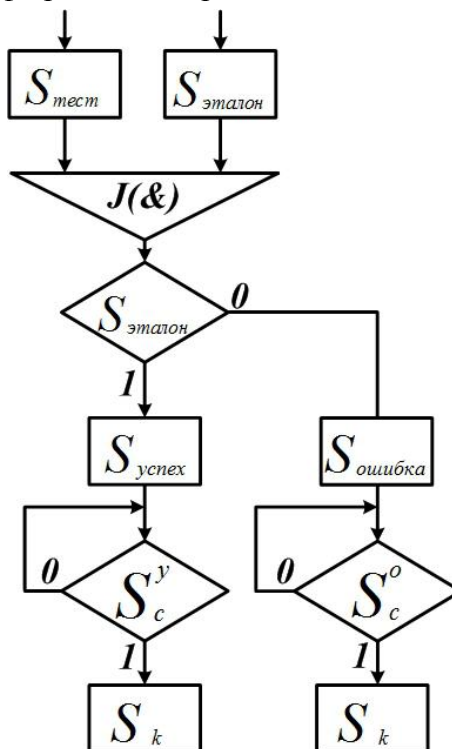


Рис. 1. Граф недетерминированного автомата управления процесса проверки нарушения свойств информации

В заключение следует отметить, что таким образом можно формализовать и более сложные алгоритмы для различных процессов управления информационной безопасностью.

Формализация процессов управления на основе языка событийных недетерминированных автоматов позволяет обеспечивать наглядность, удобство использования и возможность эффективной оптимизации и реализации систем, построенных на основе такой формализации, на программном и аппаратных уровнях. Аналитическая форма представления алгоритма управления процессами информационной безопасности в виде систем канонических уравнений позволит легко выполнить верификацию алгоритмов путем моделирования.

Список литературы

1. Вашкевич, Н. П. Достоинства формального языка, основанного на концепции недетерминизма, для функционального описания и преобразования алгоритмов логического управления процессами и ресурсами в параллельных системах обработки информации / Н. П. Вашкевич, Р. А. Бикташев // Телекоммуникации. – 2011. – № 1.
2. Вашкевич, Н. П. Выразительные возможности и эффективность формального языка, построенного на основе использования моделей недетерминированных автоматов / Н. П. Вашкевич // Известия вузов. Поволжский регион. Технические науки. – 2006. – № 6.

А. А. Гавришев, В. А. Бурмистров, И. В. Анзин

Россия, Ставрополь, Северо-Кавказский федеральный университет

К ВОПРОСУ ОБ ИСПОЛЬЗОВАНИИ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ ПРЕДОТВРАЩЕНИЯ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА К РАДИОКАНАЛУ АВТОМОБИЛЬНОЙ СИГНАЛИЗАЦИИ

Предлагается альтернативный способ защиты радиоканала автомобильной сигнализации от несанкционированного доступа за счет использования псевдослучайных последовательностей

Сегодня по статистике Россия занимает третье место в мире по угону автомобилей [1]. Одной из главных причин высокого числа краж является повсеместное использование беспроводных систем сигнализации и уменьшение цен на специальные технические средства для электронного взлома сигнализации. Одним из наиболее уязвимых мест современных беспроводных сигнализаций является радиоканал между мобильным брелоком управления (МБУ) и стационарным блоком управления (СБУ) автосигнализацией, по которому передаются управляющие команды.

Сегодня существует несколько базовых способов защиты радиоканала автомобильной сигнализации от несанкционированного доступа. Основным способом защиты является криптографическое шифрование передаваемого между брелоком и блоком управления трафика [2, 3]. Недостатками данного способа является дороговизна и время выполнения операций шифрования. Еще одним способом защиты является анализ длительности принимаемого импульса, позволяющий обнаруживать искажаемые области сигнала [4]. Недостатком данного способа защиты является необходимость наличия дополнительной защиты автомобильной сигнализации, направленной на замену уже скомпрометированной, но пока не использовавшейся злоумышленником ключевой информации. Еще одним возможным способом защиты является ослабление узкополосных помех, направленное на повышение помехозащищенности и достоверности приема сложных сигналов с фазовой манипуляцией [5]. Недостатком данного способа защиты является излишняя техническая сложность и дороговизна.

Авторами предлагается альтернативный способ защиты радиоканала автомобильной сигнализации от несанкционированного доступа. Данный способ основан по принципу «свой-чужой» с использованием псевдослучайных последовательностей (ПСП) и успешно опробован в системах охранно-пожарной и охранной сигнализации [6, 7, 8, 9, 10, 11].

Структурная схема предлагаемого устройства состоит из СБУ, в состав которого входит генератор случайных чисел (СЧ), генератор ПСП, устройство сравнения (УС) и МБУ, в состав которого входит генератор ПСП (рис. 1).

Суть предложенного способа защиты радиоканала автомобильной сигнализации от несанкционированного доступа состоит в следующем: при вхождении в режим синхронизации СБУ и МБУ на генератор СЧ СБУ поступает команда «старт», с помощью которой он вырабатывает случайное значение. Полученное случайное значение одновременно отправляется на генератор ПСП СБУ и через радиоканал на генератор ПСП МБУ. После генератор ПСП СБУ, функция генерации последовательности которого идентична функции генератора ПСП МБУ, вырабатывает последовательность, поступающую в УС. Одновременно с этим выработанная в генераторе ПСП МБУ, функция генерации последовательности которого идентична функции генератора ПСП СБУ, последовательность через радиоканал так же по-

ступает в УС. Далее в УС происходит сравнения откликов генераторов ПСП СБУ и МБУ. В случае совпадения значений, выдается сигнал «норма», служащий для оповещения о санкционированном доступе к автомобильной сигнализации. При несовпадении значений выдается сигнал «тревога», служащий для оповещения о несанкционированном доступе к автомобильной сигнализации.

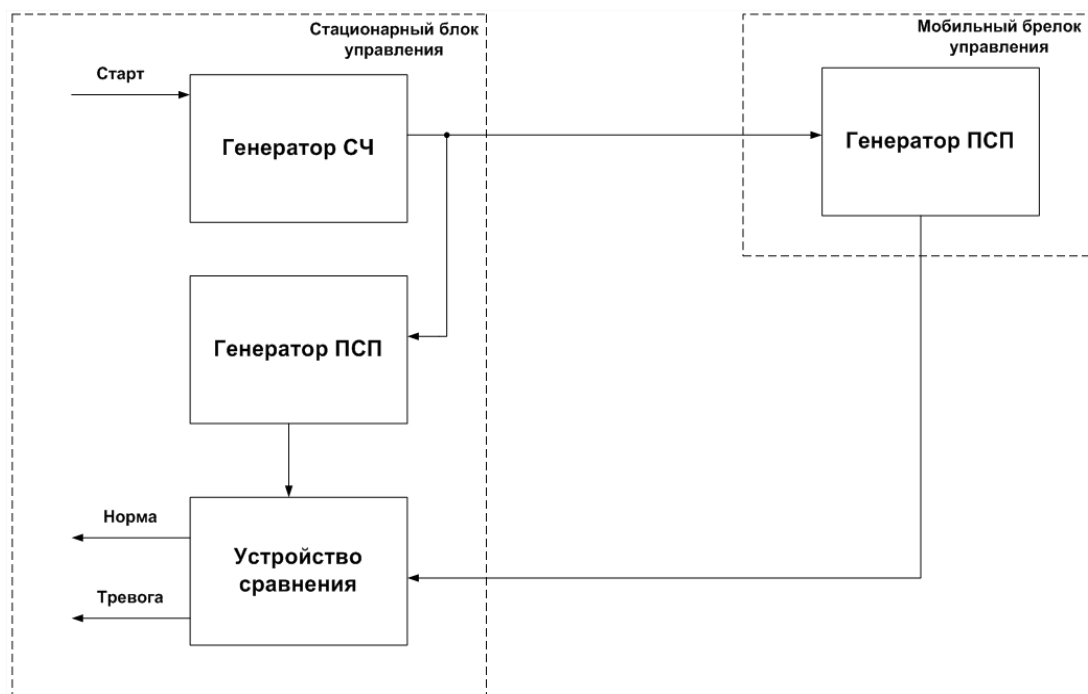


Рис. 1. Структурная схема

На основании данного способа авторами разработано устройство, которое подано в качестве заявки на предполагаемое изобретение [12].

Список литературы

1. Зикеева, Е. В. Противоугонные системы автомобиля / Е. В. Зикеева // Инновации в автомобилестроении : Междунар. науч.-практ. конф. – Вязьма : Филиал «МГИУ» в г. Вязьме, 2013 – С. 27–40.
2. Грязин, Д. С. Охранный комплекс автомобиля на базе стандарта Zigbee / Д. С. Грязин, А. А. Данилова // Наука вчера, сегодня, завтра : сб. ст. по материалам IX Междунар. науч.-практ. конф. (10 февраля 2014 г.). – URL: <http://sibac.info/13311> (дата обращения: 23.08.2014).
3. Шелованов, Е. Л. Защита данных в радиоканале при дистанционном управлении стационарными и движущимися объектами / Е. Л. Шелованов, С. Ф. Суков. – URL: <http://ea.donntu.edu.ua:8080/jspui/bitstream/123456789/14083/1/Шелованов%20Е.Л.,%20Суков%20С.Ф.pdf> (дата обращения: 23.08.2014).
4. Обеспечение целостности информации в автосигнализации мобильных объектов / С. С. Дубина, К. С. Козлов, Г. В. Сечко, А. М. Чернецкий // Международная научно-техническая конференция, приуроченная к 50-летию МРТИ-БГУИР (Минск, 18-19 марта 2014 г.): материалы конф. : в 2 ч. / редкол.: А. А. Кураев [и др.]. – Минск : БГУИР, 2014. – Ч. 1. – С. 384–385.
5. Пат на изобретение 2481978. Система защиты от несанкционированного доступа для транспортных средств / Калинин В. А., Шубарев В. А., Мельников В. А., Дикарев В. И. – Опубл. 20.05.2013.

6. Пат на изобретение 2310236. Устройство для имитозащиты контролируемых объектов / Лепешкин О. М., Осипов Д. Л., Савельев К. В., Жук А. П., Рожнов А. В., Снегирев Д. В. – Опубл. 10.11.2007.

7. Пат. 2387017 Российская Федерация. Система сопряжения устройств для имитозащиты контролируемых объектов / Осипов Д. Л., Лепешкин О. М., Харечкин П. В., Скубицкий А. В., Жук А. П. – Опубл. 20.04.2010.

8. Пат. 2417452 Российская Федерация. Устройство для имитозащиты группы контролируемых объектов / Осипов Д. Л., Лепешкин О. М., Тарасов Г. Д., Сухов А. Ю., Грунис С. И., Серебрянский С. Г. – Опубл. 27.04.2011.

9. Пат. 2447512 Российская Федерация. Устройство для имитозащиты группы контролируемых объектов / Осипов Д. Л., Лепешкин О. М., Грунис С. И., Олейников А. А. – Опубл. 10.04.2012.

10. Пат. 2455696 Российская Федерация / Осипов Д. Л., Олейников А. А., Порублев М. А. – Опубл. 10.07.2012.

11. Пат. 2520413 Российская Федерация. Устройство имитозащиты группы контролируемых объектов, основанное на логике XOR / Осипов Д. Л., Бурмистров В. А., Гавришев А. А., Порублев М. А. – Опубл. 27.06.2014.

12. Заявка на предполагаемое изобретение 2014117363. Устройство предотвращения несанкционированного использования транспортного средства за счет применения псевдослучайных последовательностей / Осипов Д. Л., Бурмистров В. А., Гавришев А. А. – Опубл. 29.04.2014.

Е. О. Голышевский

Россия, Пенза, Пензенский государственный университет

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ДЛЯ ПРОВЕДЕНИЯ СПЕЦИАЛЬНОЙ ПРОВЕРКИ
МЕТОДОМ АКТИВНОГО КОНТРОЛЯ ЦЕПЕЙ ПИТАНИЯ
С ИСПОЛЬЗОВАНИЕМ АППАРАТУРЫ ТЕСТОВОГО ПРИЕМНИКА
«RONDE&SCHWARZ»**

Представлены результаты исследования по разработке программного обеспечения для тестового приемника «Rohde&Schwarz», используемого при проведении специальной проверки методом активного контроля цепей питания.

Введение

Информационной безопасности, в том числе ее защите от утечки по различным техническим каналам, в последнее время уделяется большое внимание - ужесточаются требования, усиливается контроль. Не последнюю роль в добывании информации иностранными спецслужбами, а также службами, занимающимися промышленным шпионажем, играет применение различных средств разведки, использующих технические каналы утечки информации.

Целью специальной проверки технических средств является обнаружение специально внедренных устройств негласного съема информации, а также элементов этих технических средств, которые при воздействии на них тех или иных физических полей (например, высокочастотного электромагнитного излучения) могут быть использованы для несанкционированного получения информации (обрабатываемой или речевой).

Такой процесс является очень сложным, поэтому поиск "закладок" осуществляется по нескольким методикам и связан, как правило, с полной разборкой проверяемого технического средства. В ходе специальной проверки изымаются и возвращаются заказчику широко распространенные сейчас встроенные устройства беспроводной связи (Wi-Fi, Bluetooth, ИК-каналы), которые сами по себе уже являются каналами утечки обрабатываемой информации. Результатом специальной проверки является соответствующее Заключение (положительное или отрицательное) о возможности использования данного средства в составе объекта информатизации.

В ходе специальных исследований выявляются и измеряются параметры физических полей (излучений) технических средств, которые могут быть модулированы обрабатываемой ими информацией. Исследованиям подвергаются все устройства технического средства, в которых циркулирует информация ограниченного доступа. По результатам проведенных измерений и расчетов разрабатывается Предписание на эксплуатацию исследованного технического средства, содержащее условия и ограничения, которые необходимо выполнить для того, чтобы данное средство могло использоваться для обработки информации ограниченного распространения или устанавливаться в помещении, где такая информация обсуждается.

Радиозакладные устройства

Перехваченная информация может быть передана по воздуху (радиозакладки), по сетям питания, управления, связи (закладные устройства).

Для выявления излучающих в эфир радиозакладок необходимо определить возможный диапазон их работы и используемые виды модуляции и закрытия. Как следует из анализа существующих радиозакладных устройств, диапазон их работы достаточно широк и имеет тенденцию к продвижению в более высокие диапазоны, к использованию устройств с «прыгающими» частотами.

Основные диапазоны (по количеству известных образцов): 270...480 МГц, 115...200 МГц, 75...115 МГц.

За последнее время увеличилось количество радиозакладных устройств, работающих в диапазоне 640...1000 МГц и выше 1000 МГц.

После введения ограничений на специальные технические средства для радиозакладных устройств выделен диапазон 415...420 МГц. Однако в эксплуатации можно встретить большое количество ранее выпущенных устройств. Таким образом, радиозакладные устройства могут работать во всем диапазоне от 20 до 1000 МГц и выше.

Это существенно усложняет задачу поиска радиозакладных устройств по их излучениям. Серьезное усложнение в поиске закладных устройств вызывают изменяющиеся и совершенствующиеся виды модуляции и закрытия, используемые в закладных устройствах. Так, например, широко распространенные на начальном этапе радиозакладные устройства строились с использованием амплитудной модуляции, что позволяло использовать в качестве приемного устройства комплекса обычные бытовые приемные устройства. Однако это положительное качество часто превращалось в отрицательное, так как перехваченная и переданная в эфир информация легко обнаруживалась теми, кому она не предназначалась, обывателями, которые, прокручивая ручку своего бытового приемника, вдруг обнаруживали в эфире разговор своего соседа. Естественно, что такое обнаружение, как правило, приводило к последующему уничтожению иногда с очень большим трудом установленных закладных устройств. В радиозакладных устройствах в основном применяется модуляция несущей частоты передатчика, однако встречаются радио-

закладные устройства с модуляцией сигнала промежуточной частоты или двойной модуляции (например, радиозакладка РК-1970-SS). Прием таких сигналов на обычный супергетеродинный приемник невозможен (после детектирования прослушивается обычный шум). Для приема может быть использован только специальный приемник.

Акустоэлектрические каналы

Акустоэлектрические каналы утечки информации возникают за счет преобразований акустических сигналов в электрические. Некоторые элементы ВТСС, в том числе трансформаторы, катушки индуктивности, электромагниты вторичных электрочасов, звонков телефонных аппаратов и т.п., обладают свойством изменять свои параметры (емкость, индуктивность, сопротивление) под действием акустического поля, создаваемого источником речевого сигнала. Изменение параметров приводит либо к появлению на данных элементах электродвижущей силы (ЭДС), либо к модуляции токов, протекающих по этим элементам в соответствии с изменениями воздействующего акустического поля.

ВТСС, кроме указанных элементов, могут содержать непосредственно акустоэлектрические преобразователи. К таким ВТСС относятся некоторые типы датчиков охранной и пожарной сигнализации, громкоговорители ретрансляционной сети и т.д. Эффект акустоэлектрического преобразования в специальной литературе называют «микрофонным эффектом». Причем из ВТСС, обладающих «микрофонным эффектом», наибольшую чувствительность к акустическому полю имеют абонентские громкоговорители и некоторые датчики пожарной сигнализации.

Перехват акустоэлектрических колебаний в данном канале утечки информации осуществляется путем непосредственного подключения к соединительным линиям ВТСС специальных высокочувствительных низкочастотных усилителей. Например, подключая такие средства к соединительным линиям телефонных аппаратов с электромеханическими вызывными звонками, можно прослушивать разговоры, ведущиеся в помещениях, где установлены эти аппараты.

Технический канал утечки информации с использованием «высокочастотного навязывания» может быть осуществлен путем несанкционированного контактного введения токов высокой частоты от соответствующего генератора в линии, имеющей функциональные связи с нелинейными или параметрическими элементами ВТСС, на которых происходит модуляция высокочастотного сигнала информационным. Информационный сигнал в данных элементах ВТСС появляется вследствие акустоэлектрического преобразования акустических сигналов в электрические. Промодулированный сигнал отражается от указанных элементов и распространяется в обратном направлении по линии или излучается.

Наиболее часто такой канал используется для перехвата разговоров, ведущихся в помещении, через телефонный аппарат, имеющий выход за пределы контролируемой зоны.

Тестовый приемник «Rohde&Schwarz»ESIB 7

Тестовый приемник «Rohde&SchwarzESIB 7» используется для измерения очень малых уровней напряжения, например, электромагнитных помех на автомобильных антеннах согласно CISPR 25, семейство ESIB предлагает предусилитель с усилением 20 дБ и с диапазоном частот от 9 кГц до 7 ГГц. Во избежание перегрузок, предусилитель включается между ВЧ преселектором и входным смесителем. При использовании этого предусилителя уровень собственных шумов ESIB снижается до такой степени, что напряженность поля электромагнитных помех, получен-

ная в обзорном измерении с использованием пикового детектора, логопериодической антенны и 10 м соединительного кабеля, остается заметно ниже квазипикового предела EN55022.

Заключение

В ходе исследования было выявлено, что разработка программного обеспечения для тестового приемника «Rohde&Schwarz» ESIB 7 существенно сократит время проведения специальной проверки технического средства, содержащего радиозакладное устройство.

Список литературы

1. InformationSecurity / Информационная безопасность. – 2008. – № 1.
2. Бузов, Г. А. Защита от утечки информации по техническим каналам / Г. А. Бузов, С. В. Калинин, А. В. Кондратьев // Руководство пользователя тестового приемника «Rohde&Schwarz» ESIB 7.

Б. И. Ефимов

Россия, Омск, Омский государственный технический университет

АНАЛИЗ ЭФФЕКТИВНОСТИ УГРОЗ БЛОКИРОВАНИЯ И ПОДМЕНЫ ОТВЕТОВ ЭКСПЕРТОВ В СИСТЕМАХ ПРИНЯТИЯ РЕШЕНИЙ

Сформулированы основные требования к обеспечению информационной безопасности систем принятия решений с привлечением экспертов. Проведен анализ возможности использования средств анализа рисков в системах принятия решений с привлечением экспертов. Предложено решение задачи вычисления вероятности принятия ложного решения под воздействием угроз информационной безопасности.

В настоящее время одними из наиболее динамично развивающихся информационных систем являются системы принятия управленческих решений. Одним из видов данных систем являются системы, построенные на использовании знаний экспертов-аналитиков.

Требования к обеспечению информационной безопасности

При разработке указанных систем обеспечение информационной безопасности является одной из важнейших задач.

При рассмотрении вопроса защиты информационных систем для определения необходимого уровня информационной безопасности должны учитываться как формализованные требования, заданные соответствующими нормативными документами, так и дополнительные требования, зависящие от текущей статистики угроз и гипотетически возможных угроз для конкретного защищаемого объекта. Формализованные требования определяют базовый уровень информационной безопасности, рассматривается стандартный набор наиболее распространенных угроз безопасности и применяется типовая набор контрмер. При наличии дополнитель-

ных требований должен быть проведен полный вариант анализа рисков информационной системы.

Для различных информационных систем задачи обеспечения информационной безопасности бывают различными и варьируются в широких пределах. В системах принятия решения с привлечением экспертов для лица, принимающего решение (ЛПР), значимым является конечный результат – выбор экспертами одной из предложенных альтернатив.

Поэтому система защиты информации должна быть построена таким образом, чтобы выполнялась единственная задача – решение, принимаемое при условии реализации возможных угроз должно быть таким же, что и решение, которое было бы принято системой при полном отсутствии угроз информационной безопасности. При этом угрозы являются существенными только в том случае, если они приводят к изменению выбранной экспертами альтернативы. Будем считать, что для ЛПР требование, чтобы при возможных угрозах безопасности выбиралась та же альтернатива, что и в случае отсутствия угроз, является как необходимым, так и достаточным, т.е. значение «перевеса», с которым побеждает одна альтернатива над другой и процентное распределение голосов экспертов между альтернативами не имеют никакого значения.

Возможность применения существующих средств анализа

Существующие в настоящее время инструментальные средства анализа и управления рисками в зависимости от выполняемых задач можно разделить на средства базового уровня и средства полного анализа рисков.

Информационную систему принятия решений с привлечением экспертов нельзя назвать «типовой» информационной системой, так как задачи по обеспечению ИБ в данных системах сильно отличаются от «типовых» задач обеспечения ИБ распределенных информационных систем (обеспечение конфиденциальности, доступности и целостности информации). В системах принятия решения главной задачей системы в целом является недопущение изменения решения экспертов, при этом приведенные « типовые » требования могут быть в определенной степени нарушены. Поэтому использования средств базового уровня в системах принятия решений недостаточно.

Существующие программные средства обеспечения повышенного уровня безопасности можно классифицировать в зависимости от используемых в них методов и подходов к оценке рисков информационной безопасности [1]:

- методы оценки рисков с использованием качественных показателей;
- методы, оценивающие величину ущерба как стоимость восстановления ресурса;
- методы, оценивающие риск как математическое ожидание среднегодовых потерь;
- методы, рассматривающие информацию в качестве основного объекта защиты (модель информационных потоков, модель анализа угроз и уязвимостей).

Как показано в [1], в целом для оценки рисков систем принятия решений с привлечением экспертов не подходит ни один из существующих методов анализа рисков. Несмотря на все разнообразие указанных средств, все они исходят из предположения, что любые угрозы информационной безопасности, на какой бы элемент системы они не были направлены, приводят к тому или иному снижению уровня безопасности информационной системы в целом. Т.е. любое вложение в защиту информации снижает риски.

Однако применительно к системам принятия решений с привлечением экспертов должен использоваться другой подход, определяющий, что существенными являются лишь те угрозы, которые приводят к изменению выбранной экспертами альтернативы. Соответственно, и система защиты информации должна строиться таким образом, чтобы обеспечить выполнение указанного требования.

Воздействие угроз информационной безопасности

Рассмотрим, как влияют угрозы информационной безопасности по изменению и блокированию ответов экспертов на принятие решения по выбору одной из двух альтернатив.

Для вычисления вероятности принятия решения, отличного от решения, которое было бы принято по результатам опроса экспертов при отсутствии угроз информационной безопасности, определим следующие моменты:

- опрос проводится по выбору одной из двух альтернатив: «0» и «1»;
- действия злоумышленника направлены на принятие альтернативы «0»;
- угрозы по блокированию ответа эксперта являются случайными угрозами и могут быть реализованы в отношении любого эксперта.

Необходимо рассматривать только те случаи, когда эксперты в большинстве проголосовали за альтернативу «1» или поровну с альтернативой «0», но действия злоумышленника могут привести к принятию альтернативы «0» (в дальнейшем – наступление события A_0).

В [2] и [3] рассчитаны вероятности возникновения события A_0 , т.е. вероятности того, что действия злоумышленника приводят к изменению выбранной экспертами альтернативы для двух частных случаев:

- угрозы по блокированию ответов экспертов отсутствуют ($P_{\text{bloc } i} = 0$), вероятности изменения ответов экспертов $P_{\text{change } i}$ одинаковы для всех экспертов;
- угрозы по изменению ответов экспертов отсутствуют ($P_{\text{change } i} = 0$), вероятности блокирования ответов экспертов $P_{\text{bloc } i}$ одинаковы для всех экспертов.

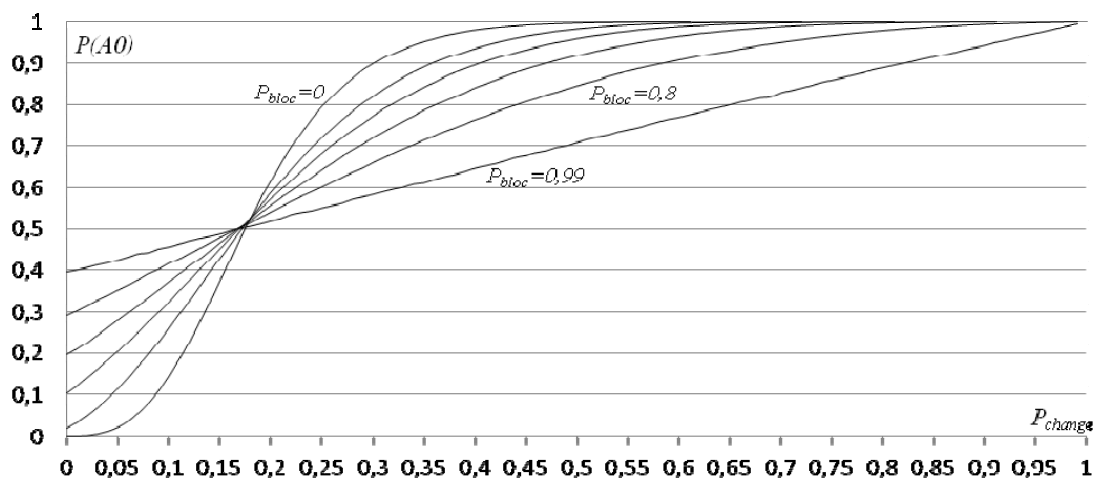
Указанные расчеты были проведены с использованием положений теории вероятностей.

Для вычисления вероятности принятия ЛПР ложного решения при одновременной реализации угроз по изменению и блокированию ответов экспертов использование только теории вероятности приводит к довольно сложным расчетам.

Будем полагать, что вероятности блокирования ответов экспертов $P_{\text{bloc } i}$ одинаковы для всех экспертов и вероятности изменения ответов эксперта под действием угроз информационной безопасности $P_{\text{change } i}$ также одинаковы для всех экспертов.

Для вычисления вероятности события A_0 в зависимости от вероятностей $P_{\text{bloc } i}$ и $P_{\text{change } i}$, количества экспертов m и доли экспертов n/m , проголосовавших за альтернативу «1», разработан модуль на языке программирования MATLAB. Данная программа моделирует работу системы принятия решений с привлечением экспертов. При каждом голосовании ответы экспертов блокируются (не доходят до ЛПР) с вероятностью $P_{\text{bloc } i}$ и изменяются с «1» на «0» с вероятностью $P_{\text{change } i}$, после чего определяется, какая из двух альтернатив выбирается ЛПР, т.е. принимается правильное («1») или ложное («0») решение. Для получения достоверного значения вероятности принятия ложного решения для каждого набора исходных данных проводится большое (около 10^6) количество вычислений.

На рис. 1 представлена зависимость вероятности $P(A_0)$ от вероятности изменения отдельного ответа эксперта P_{change} , при различных значениях вероятностей блокирования ответа эксперта P_{bloc} .



На графике приведены зависимости для различных $P_{bloc} = \{0; 0,2; 0,4; 0,6; 0,8; 0,99\}$ при $m = 20$, $n/m = 0,6$.

Значение $P_{bloc}=0,99$ взято для иллюстрации поведения системы при $P_{bloc} \rightarrow 1$.

Список литературы

1. Ефимов, Б. И. Возможность применения существующих средств анализа рисков в системах принятия решений с привлечением экспертов / Б. И. Ефимов // Омский научный вестник. Серия «Приборы, машины и технологии». – 2011. – № 3 (103). – С. 281–284.
2. Ефимов, Б. И. Вероятность принятия ложного решения под воздействием угроз информационной безопасности в системах принятия решений с привлечением экспертов / Б. И. Ефимов, Р. Т. Файзуллин // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2013. – № 1 (27). – С. 69–74.
3. Ефимов, Б. И. Устойчивость объективного решения экспертов при воздействии угроз по блокированию информации в системах принятия решений с привлечением экспертов / Б. И. Ефимов, Р. Т. Файзуллин // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2014. – № 2 (32). – С. 66–70.

С. Л. Зефилов, А. Ю. Щербакова

Россия, Пенза, Пензенский государственный университет

АНАЛИЗ СОСТОЯНИЙ ИНЦИДЕНТОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРИ ИХ ОБРАБОТКЕ

Работа посвящена анализу состояний инцидентов информационной безопасности при их обработке с помощью мер по предотвращению, обнаружению, сдерживанию, устранению инцидентов и восстановлению системы. Результатом анализа является дерево событий, универсальное для различных видов инцидентов информационной безопасности.

С ростом числа информационных систем и совершенствованием информационных технологий растет и число инцидентов информационной безопасности (ИБ), под которыми понимается одно или несколько нежелательных событий, которые

влиять на информационную безопасность активов систем и могут привести к негативным последствиям. Такими последствиями могут быть, например, нарушение конфиденциальности, целостности и доступности информационных активов, прерывания бизнес-процессов и другие. Прогнозирование инцидентов ИБ является ключом к планированию эффективной системы управления информационной безопасностью организации. Ввиду недостаточности статистических данных о нежелательных событиях ИБ, вызывающих негативные последствия для организаций, и неоднородности этих данных, закономерности реализации инцидентов ИБ выявить сложно. Поэтому для решения задач прогнозирования инцидентов ИБ должно использоваться моделирование их сценариев. Для этого должны быть определены состояния инцидентов ИБ, условия переходов в эти состояния, а также возможные пути развития инцидентов, находящихся в том или ином состоянии. В рамках настоящей статьи будут рассмотрены возможные состояния инцидента ИБ при его обработке с помощью различных защитных мер.

Для предотвращения инцидентов ИБ в организации реализуются защитные меры, которые выбираются по результатам оценки рисков ИБ. Однако все защитные меры имеют свою результативность [1], и всегда существует возможность того, что инциденты произойдут. Важную роль при обработке инцидентов ИБ играет своевременное реагирование на нежелательные события ИБ, которое может минимизировать затраты на восстановление системы после инцидента и снизить риск его повторного возникновения. Известно [2], что реагирование на инцидент ИБ подразумевает применение защитных мер по обнаружению инцидента ИБ, его сдерживанию, устранению и восстановлению системы после него, а также «антикризисные» действия. Алгоритм реагирования на инциденты ИБ приведен на рис. 1.

Рассмотрим возможные состояния инцидента ИБ при его обработке.

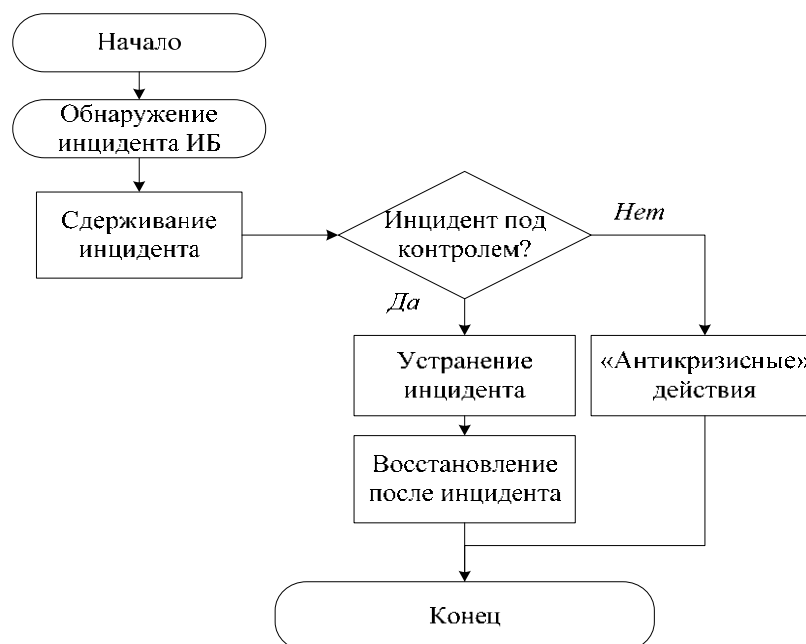


Рис. 1. Реагирование на инциденты ИБ

Первое состояние инцидента – возникновение инцидента ИБ и применение к нему защитных мер, выбранных по результатам оценки рисков. Такими защитными мерами могут быть, например, установка антивирусного программного обеспечения, контроль доступа в помещения, контроль сетевого доступа, обеспечение осведомленности персонала вопросам ИБ и др. В зависимости от результативности за-

щитных мер, выбранных по результатам оценки рисков, инцидент может быть предотвращен или перейти во второе возможное состояние.

Второе состояние инцидента – инцидент не предотвращен, применение защитных мер по обнаружению инцидента ИБ. Обнаружение инцидента ИБ может производиться с помощью технических средств, например, средств обнаружения вторжений, антивирусного программного обеспечения, а также персонала, например, с помощью анализа журналов мониторинга, сообщений пользователей об ошибках и др. Если инцидент был обнаружен, он переходит в состояние, характеризующееся применением к нему защитных мер по сдерживанию (третье возможное состояние). Если инцидент не был обнаружен, он переходит в состояние, когда его развитие происходит неконтролируемо и требуется применение «антикризисных» действий (пятое возможное состояние инцидента).

Третье состояние инцидента – инцидент обнаружен, применение защитных мер, направленных на сдерживание инцидента ИБ. Сдерживание инцидента ИБ производится для ограничения развития его последствий. Мерами по сдерживанию могут быть, например, изолирование узлов, систем, отключение серверов электронной почты и др. Если инцидент был сдержан, он переходит в состояние, характеризующееся применением к нему защитных мер по его устранению и восстановлению системы после него (четвертое возможное состояние инцидента). Если инцидент не удалось сдержать, его развитие становится неконтролируемым и требуется применение «антикризисных» действий (пятое возможное состояние инцидента).

Четвертое состояние инцидента – инцидент сдержан (находится под контролем), применение защитных мер, направленных на устранение инцидента ИБ и восстановление системы после него. Мерами по устранению инцидента и восстановлению после него являются, например, ликвидация вредоносного кода, восстановление программ или операционной системы из резервных копий, блокирование учетных записей пользователей, чьи данные были скомпрометированы и т.д. В случае полного устранения инцидента и восстановления системы, инцидент считается обработанным успешно. Если меры по устранению и восстановлению не были полностью выполнены, считается, что инцидент обработан с частичными последствиями (например, существует возможность его возобновления через неопределенное время или восстановление системы произошло позже требуемого для достижения цели определенной бизнес-операции времени).

Пятое состояние инцидента – инцидент не сдержан (является неконтролируемым), применение «антикризисных» действий. «Антикризисные» действия, включающие, например, активацию плана непрерывности бизнеса и восстановления его после прерываний, предпринимаются в случае неконтролируемого развития инцидента и проявления его последствий на уровне бизнес-процессов (например, прерывание бизнес-операции). Если предпринятые действия привели к устранению последствий инцидента и восстановлению системы, полагаем, что инцидент был обработан с минимальными потерями. Если устранить инцидент и восстановить систему для продолжения бизнес-деятельности с помощью «антикризисных» действий не удалось, то происходит эскалация инцидента, т.е. реализация негативных для организации последствий инцидента.

Таким образом, обработку инцидента ИБ можно представить в виде дерева событий [3], в узлах S_i которого отображаются состояния инцидента в соответствии с применением защитных мер на различных этапах его обработки. Исходами дерева решений S_k являются конечные состояния инцидента. Вероятности исходов дерева событий $P_{СК}$ могут быть вычислены, если известна результативность применяемых защитных мер.

На рис. 2 представлено дерево событий обработки инцидента ИБ, где:

- S_1 – возникновение инцидента ИБ, применение защитных мер по предотвращению инцидента ИБ;
- C_1 – инцидент предотвращен;
- S_2 – инцидент не предотвращен, применение защитных мер по обнаружению инцидента ИБ;
- S_3 – инцидент обнаружен, применение защитных мер, направленных на сдерживание инцидента ИБ;
- S_4 – инцидент сдержан, реализация защитных мер, направленных на устранение инцидента ИБ и восстановление системы после него;
- C_2 – инцидент успешно обработан;
- C_3 – инцидент обработан с частичными последствиями;
- S_5 – инцидент не сдержан, применение «антикризисных» действий;
- C_4 – инцидент обработан с минимальными потерями;
- C_5 – произошла эскалация инцидента.

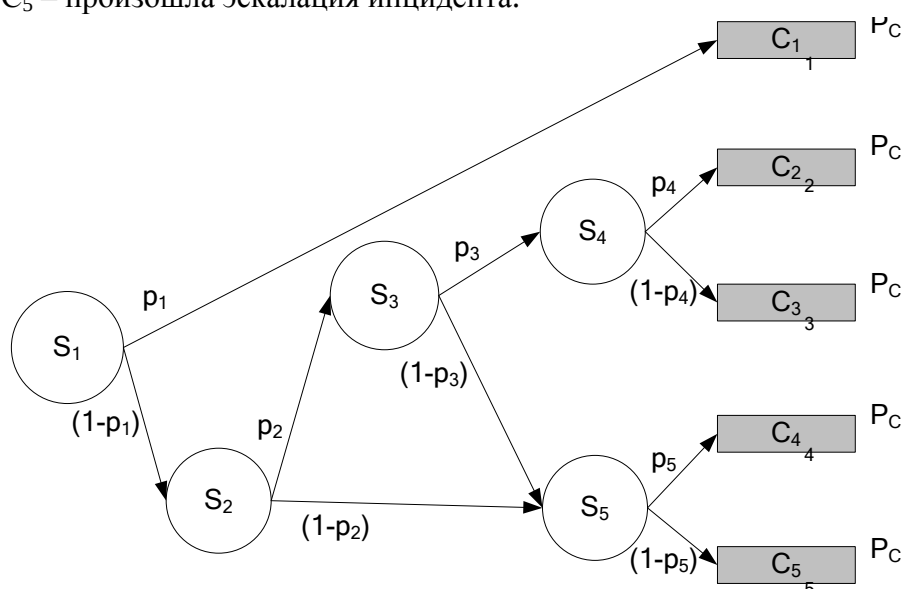


Рис. 2. Дерево событий обработки инцидента ИБ

Дерево событий показывает пути к каждому исходу обработки инцидентов ИБ и является универсальным для различных видов инцидентов ИБ. Идентификация результативности защитных мер по предотвращению, обнаружению, сдерживанию, устранению инцидентов и восстановлению после них позволит определить вероятность того или иного исхода обработки. Однако это возможно только при рассмотрении на основе разработанного дерева событий конкретных видов инцидентов ИБ. Построение имитационной модели на основе разработанного дерева событий даст возможность определить вероятностно-временные параметры исходов обработки инцидентов ИБ.

Список литературы

1. Щербакова, А. Ю. Модель сценариев инцидента внедрения вредоносного программного обеспечения / А. Ю. Щербакова // Информация и безопасность. – 2013. – № 3.
2. ISO/IEC 27035:2011 Information technology – Security techniques – Information security incident management.
3. ISO/IEC 31010:2009 Risk management – Risk assessment techniques.

В. В. Легошин, Н. Б. Жукова

Россия, Пенза, Пензенский государственный университет

**СРАВНЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ
АВТОРЕГРЕССИОННОГО СПЕКТРАЛЬНОГО ОЦЕНИВАНИЯ
ПРИМЕНИТЕЛЬНО К ЗАДАЧЕ СИНХРОНИЗАЦИИ
В СИСТЕМАХ СКРЫТОЙ ПЕРЕДАЧИ ИНФОРМАЦИИ**

Представлены результаты сравнения двух наиболее эффективных известных алгоритмов авторегрессионного спектрального оценивания с помощью программного средства, моделирующего данные алгоритмы.

В основе построения систем скрытой передачи информации лежит использование методов расширения спектра, обеспечивающих возможность распределения энергии передаваемого сигнала в широком диапазоне частот. При этом мощность полезного сигнала не превышает мощности шума эфира. Расширение спектра основывается на изменении тем или иным образом информационного сигнала в соответствии с законом, задаваемым псевдослучайной цифровой последовательностью, вид которой известен получателю (что позволяет ему восстановить информационный сигнал), но неизвестен противнику. В результате для последнего оказывается затруднительным не только перехватить передаваемую информацию, но и обнаружить сам факт передачи.

Существуют различные методы построения систем с расширенным спектром, из которых наибольшее распространение получили прямое расширение спектра, программная перестройка рабочей частоты (ППРЧ) и их комбинации. (В зарубежной литературе данные методы принято называть соответственно Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS)).

Современные технологии обеспечивают полосы ППРЧ порядка нескольких ГГц, что существенно превышает аналогичные показатели систем с прямым расширением спектра и обуславливает предпочтительность использования систем с ППРЧ.

В системах скрытой передачи информации вероятность обнаружения факта передачи зависит от продолжительности сеанса связи. В связи с этим актуальным является сокращение длительности служебных процедур и, в частности, процедуры синхронизации. Одним из наиболее перспективных методов быстрого установления первоначальной синхронизации в системах с ППРЧ является метод рекуррентного поиска [1], при котором в качестве датчика псевдослучайного кода, задающего закон изменения рабочей частоты, используется регистр сдвига с линейной обратной связью (РСЛОС). Код РСЛОС определяется на основе L (L – количество разрядов РСЛОС) последовательно выполняемых оценок частоты следующих друг за другом скачков, причем из каждого двоичного числа, соответствующего очередной оценке, используется лишь один старший или наиболее значимый бит (НЗБ).

Применение на этапе определения НЗБ параметрических методов спектрального оценивания по рабочему сигналу на основе авторегрессионной модели [2–5] позволяет при приемлемой сложности реализации получить высокую помехоустойчивость, что в конечном итоге уменьшает среднюю длительность поиска за счет сокращения неудачных попыток, связанных с неправильным определением отдельных НЗБ.

Согласно методу авторегрессионного спектрального оценивания принимаемый сигнал (представляющий собой передаваемый сигнал с наложенным на него

шумом) после дискретизации с частотой ω задается последовательностью отсчетов и рассматривается как авторегрессионный процесс, определяемым выражением:

$$y(k) = \sum_{m=1}^M \alpha_m y(k-m) + G n(k),$$

где $n(k)$ – последовательность отсчетов белого шума; $\alpha_m, m = \overline{1, M}$ и G – некоторые постоянные величины.

При этом энергетический спектр (ЭС) авторегрессионного процесса описывается выражением

$$Y(\omega) = 1 / \left| 1 - \sum_{m=1}^M \alpha_m e^{-j\omega m} \right|^2,$$

т.е. полностью определяется набором коэффициентов $\alpha_m, m = \overline{1, M}$. Таким образом, данный метод спектрального анализа сводится к определению коэффициентов авторегрессионной модели.

Наиболее известными алгоритмами определения коэффициентов авторегрессионной модели являются стохастический градиентный метод и метод идентификации параметров решетчатой структуры.

При использовании первого метода линейный предсказатель моделируется адаптивным трансверсальным фильтром, коэффициенты которого подстраиваются стохастическим градиентным методом.

Второй возможный подход к реализации АСО основывается на моделировании линейного предсказателя в виде решетчатой (лестничной) структуры.

При этом в процессе адаптации определяются не сами коэффициенты фильтра линейного предсказания α_m , а так называемые коэффициенты частной корреляции (коэффициенты отражения) p_m .

Для упрощения реализации метода идентификации параметров решетчатой структуры применяется блочная обработка данных.

Для исследования эффективности указанных методов было разработано программное средство, моделирующее часть системы скрытой передачи информации, в том числе: источник сигнала, модулированного с помощью ППРЧ; канал связи с аддитивным гауссовским шумом; блоки быстрой синхронизации, проводящие оценку несущей частоты на основе двух указанных выше методов.

Моделирование проводилось при следующих параметрах:

- 1) частота дискретизации – 20480000 Гц;
- 2) частота перестройки рабочей частоты – 10000 Гц;
- 3) количество перестроек во время передачи одного бита равно – 10;
- 4) количество передаваемых информационных бит – 1000;
- 5) уровень сигнал/шум в канале равен – 12 дБ;
- 6) порядок исследуемых фильтров – от 2 до 50 с шагом 1;
- 7) разница между соседними несущими частотами для ППРЧ 14000 Гц;
- 8) разница частот между '0' и '1' при частотной модуляции 3000 Гц;
- 9) частота, соответствующая нулевому состоянию РСЛОС равна 1025000 Гц.

Были построены графики зависимости вероятности ошибки в определении наиболее значимого бита (НЗБ) Р от порядка фильтра М, представленные на рис. 1, 2.

Кривая 1 соответствует стохастическому градиентному методу, кривая 2 – для алгоритма идентификации параметров решетчатой структуры на основе блочной обработки данных.

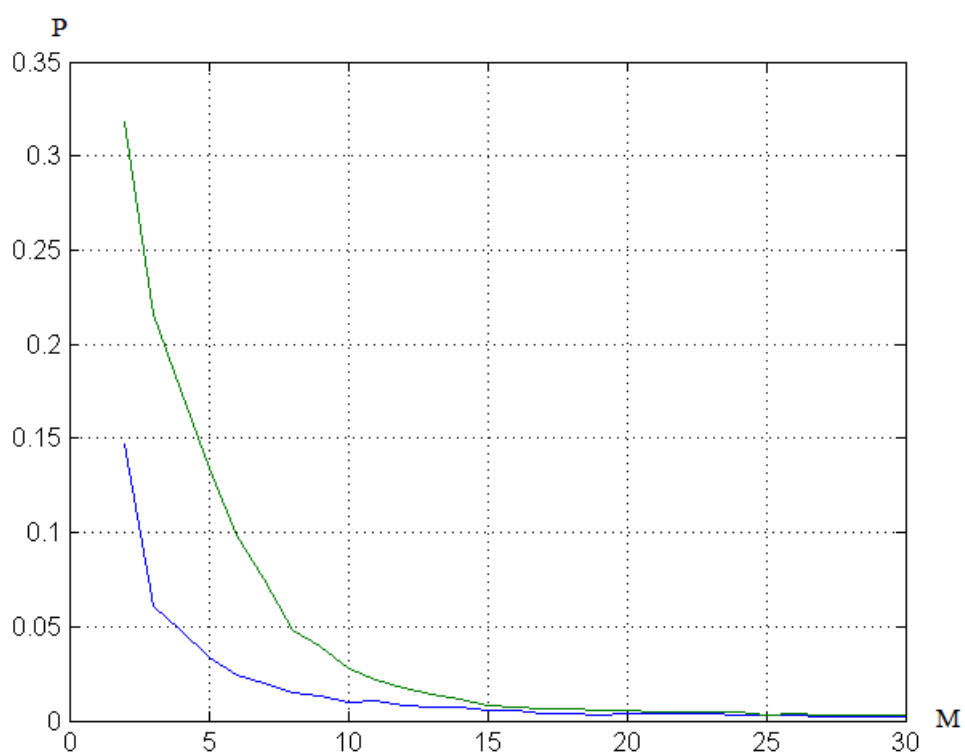


Рис. 1. Сравнение алгоритмов для порядков фильтра от 2 до 30

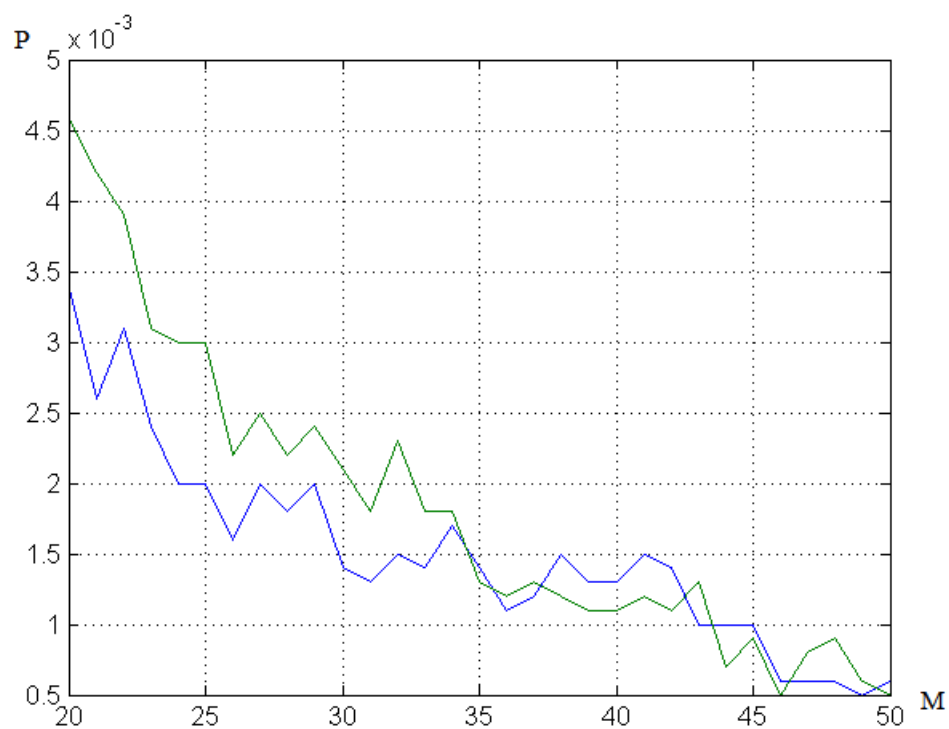


Рис. 2. Сравнение алгоритмов для порядков фильтра от 20 до 50

Из графиков видно, что для порядков фильтра от 2 до 35 алгоритм идентификации параметров решетчатой структуры обладает лучшими характеристиками.

Для фильтров больших порядков (до 50) оба метода показывают одинаковые результаты, однако градиентный метод значительно проще в реализации и обладает большим быстродействием, что делает его более привлекательным для реализации в системах синхронизации.

Список литературы

1. Журавлев, В. И. Поиск и синхронизация в широкополосных системах / В. И. Журавлев. – М. : Радио и связь, 1986. – 240 с.
2. Elhakeem, A. K. New Code Acquisition Techniques in Spread-Spectrum Communication / A. K. Elhakeem, G. S. Takhar, S. C. Gupta // IEEE Transactions on Communications, February 1980. – 1980. – Vol. COM-28, № 2. – P. 249–257.
3. Ponnusami, J. A. Acquisition of Pseudonoise Codes in FH Systems / J. A. Ponnusami, M. D. Srinath // IEEE Transactions on Aerospace and Electronic Systems, May 1981. – 1981. – Vol. AES-17, no. 3. – P. 335–341.
4. Chau, Y. A. Spectral-Estimation-Based Acquisition for Frequency-Hopping Spread-Spectrum Communications in Nonfading or Rayleigh-Fading Channel / Y. A. Chau, J.-K. Wang // IEEE Transactions on Communications, April 1997. – 1997. – Vol. 45, no. 4. – P. 44–455.
5. Марпл-мл., С. Л. Цифровой спектральный анализ и его приложения / С. Л. Марпл-мл. ; пер. с англ. под ред. И. С. Рыжака. – М. : Мир. 1990. – 585 с.

С. В. Вовк

Россия, Пенза, Пензенский государственный университет

МОДЕЛИРОВАНИЕ И ФИЛЬТРАЦИЯ СИГНАЛОВ В СРЕДЕ SCILAB

Методы моделирования, исследования спектра и фильтрации сигналов в среде SciLab.

В настоящее время цифровая обработка сигналов является магистральным направлением развития современной информационной техники. К числу наиболее важных и распространенных задач, решаемых при цифровой обработке сигналов, являются цифровая фильтрация и спектральный анализ. Именно поэтому было решено разработать лабораторную работу предназначена для изучения студентами методов моделирования, исследования спектра и фильтрации сигналов с использованием среды SciLab. Среда SciLab выбрана по причине того, что она является аналогом лицензированной системы MathLab, но распространяется бесплатно.

В начале работы студентам предлагается сформировать сигнал заданной формы и построить его спектр. Для каждого типа сигнала это осуществляется благодаря собственному выражению для одного периода этого сигнала. Например, один период непрерывного сигнала типа «меандр» можно описать выражением:

$$x_1(t) = \begin{cases} E & 0 \leq t < T/2 \\ -E & T/2 \leq t < T \end{cases}. \quad (1)$$

В дискретном времени равенство (1) трансформируется к виду:

$$x_i(k \Delta t) = \begin{cases} E & 0 \leq k \leq N_i/2 - 1 \\ -E & N_i/2 \leq k \leq N_i - 1 \end{cases} \quad (2)$$

где $\Delta t = 1/f_d$ – шаг дискретизации.

Далее необходимо сформировать вектор сигнала, удовлетворяющего соотношению (2). Прежде всего необходимо по заданным значениям частоты дискретизации f_d и периода сигнала T определить число отсчетов N_i , содержащихся в одном периоде сигнала. Затем построить вектора отсчетов, принадлежащих одному периоду сигнала:

$$x\_meandr\_1 = e * [ones(1, np/2), -ones(1, np/2)].$$

После того, как были сформированы вектора, соответствующие одному периоду сигнала, необходимо создать вектор последовательности, содержащей заданное число N отсчетов этих сигналов

$$: x\_meandr = [x\_meandr\_1, x\_meandr\_1, x\_meandr\_1].$$

После чего выполняется построение графиков сформированных таким образом сигналов. Для построения графиков целесообразно воспользоваться командами *subplot* и *plot2d*.

С помощью команды *subplot* графическое окно разделяется на «матрицу» из отдельных областей (подокон), в каждой из которых можно изобразить отдельный график. Синтаксис функции: *subplot(m, n, p)*, где параметры m, n, p – положительные целые числа. По этой команде графическое окно разбивается на матрицу размером $m \times n$ подокон и текущим выбирается окно с номером p . Непосредственно для построения графиков в каждом окне используется команда *plot2d*. В простейшем случае для этого достаточно записать: *plot2d(fr, hf)*, указав векторы значений аргумента (fr) и графически отображаемой функции (hf). При этом диапазон значений величин, откладываемых по осям координат графика, выбирается автоматически, исходя из значений fr и hf .

Для проведения анализа спектра этих сигналов необходимо провести вычисление дискретных спектров амплитуд (модулей дискретного преобразования Фурье (ДПФ) $|X(nf)|$) сформированных сигналов и построение их графиков. Вычисление ДПФ в среде Scilab осуществляется с использованием команды *fft*.

Расчет нерекурсивного фильтра, позволяющего выделить заданную гармонику сигнала, следует провести с использованием функции *eqfir(pf, bedge, des, wate)*. При этом рассчитываемый фильтр должен быть однородным (т.е. все элементы вектора *wate* должны быть равны 1) и иметь порядок $pf = 52$. Параметр *bedge* задает характерные частоты желаемой АЧХ фильтра и определяется как матрица (размером 3×2 для ППФ и ПЗФ, или 2×2 для ФВЧ и ФНЧ), параметр *des* определяет тип рассчитываемого фильтра и задается в виде 3-х (для ППФ и ПЗФ) или 2-х (для ФВЧ и ФНЧ) компонентного вектора, каждая составляющая которого представляет собой значение идеальной АЧХ в соответствующей полосе частот. Определив вектор коэффициентов нерекурсивного фильтра, нужно вычислить его АЧХ и оценить неравномерность АЧХ в полосах пропускания δ_{pr} и непропускания δ_{npr} . Величины δ_{pr} и δ_{npr} будут использоваться в качестве исходных данных при расчете каскадных рекурсивных фильтров (Баттерворта и эллиптического), выполняемом с использованием функции *eqiir(ftype, approx, om, deltap, deltas)*. При

этом значения составляющих вектора om вычисляются на основе соотношений для расчета частот среза, и принимается $deltap = delta\_pr$, $deltas = delta\_npr$. В результате расчета рекурсивных фильтров должны быть сформированы векторы коэффициентов полиномов передаточных функций эллиптического фильтра $H_{el}(z)$ и фильтра Баттерворта $H_{but}(z)$.

Затем проводится реализация процедуры цифровой фильтрации. Как известно, в общем случае выходной сигнал $y(k)$ каузального линейного цифрового фильтра связан с входным сигналом $x(k)$ и импульсной характеристикой фильтра $hn(k)$ соотношением линейной свертки, которая в случае нерекурсивного фильтра трансформируется в конечную свертку. В среде Scilab существует команда *flts*, которая реализует вычисления, предписываемые данным алгоритмом. При использовании этой команды свойства линейной системы (фильтра) задаются вектором коэффициентов полиномов передаточной функции, который получается в результате применения методов расчета рекурсивных фильтров.

Однако при расчете нерекурсивных фильтров с использованием команды *eqfir* возвращаемый этой командой вектор коэффициентов фильтра *hn* представляет собой набор отсчетов импульсной характеристики фильтра и не может быть непосредственно использован в качестве аргумента функции *flts*. Для того, чтобы получить возможность использования этой команды в данном случае, необходимо преобразовать *hn* в вектор коэффициентов полинома передаточной функции *hz*, характеризующий передаточную функцию системы $H(z)$. В среде scilab это можно сделать с помощью *syslin*. Заданию аргументов команды *syslin* должно предшествовать определение базового элемента полинома z , осуществляемое посредством выполнения команды $z = poly(0, 'z')$. Смысл этой команды заключается в формировании полинома по степеням z , корнем которого (об этом говорит запись $'z'$), является значение $z = 0$. Этим полиномом и является моном z . Определив таким образом базовый элемент полинома, можно сформировать основной аргумент команды *syslin* в виде: $poly(hn, 'z', 'c') ./ z^{pf}$. Последнее выражение описывает создание полинома по степеням z (об этом говорит запись $'z'$), коэффициентами которого (что следует из записи $'c'$) являются значения составляющих вектора *hn*, причем каждое слагаемое полинома делится на z в степени pf . Полностью команда *syslin*, задающая вектор *hz*, записывается в виде $hz = syslin('d', poly(hn, 'z', 'c') ./ z^{pf})$, причем аргумент $'d'$ указывает на то, что рассматривается фильтр, работающий в дискретном (а не непрерывном) времени.

В итоге процедура фильтрации может быть осуществлена при помощи команд:

$$z = poly(0, 'z')$$

$$hz = syslin('d', poly(hn, 'z', 'c') ./ z^{pf})$$

$$y = flts(x, hz)$$

В разрабатываемой лабораторной работе процедура фильтрации используется для выделения заданной гармоник из полигармонического сигнала, сформированного в начале работы.

Таким образом, предложенное задание позволяет студентам приобрести практические навыки выполнения процедур моделирования, исследования спектра и фильтрации сигналов, которые являются базовыми в процессе цифровой обработки сигналов.

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ТЕХНОЛОГИИ IP-ТЕЛЕФОНИИ ПРИ ИСПОЛЬЗОВАНИИ IP-АТС «ASTERISK»

Рассмотрен комплексный подход к анализу обеспечения информационной безопасности использования технологии VoIP-телефонии на базе программной IP-АТС «Asterisk».

Нарастающая в последнее время популярность цифровой телефонии в виду ее неоспоримых преимуществ перед аналоговой влечет за собой ряд требований к безопасности осуществляемых с ее помощью вызовов. Практическая реализация защитных мер по осуществлению безопасных VoIP-соединений внутри организации и за ее пределами подразумевает комплексный подход к обеспечению безопасности технологии IP-телефонии. В данной работе проанализировано три составляющих этого подхода: обеспечение конфиденциальности потока данных, защита от несанкционированного доступа (НСД) к узлам сети и применение вспомогательных организационных защитных мер.

VoIP-трафик представляет собой поток оцифрованных медиа-данных, разбитых на пакеты и передаваемых по IP-сетям с помощью транспортного протокола RTP (Real-time Transport Protocol – транспортный протокол в реальном времени). Для установления соединения между узлами сети, синхронизации параметров сессии и сигнализации используется протокол прикладного уровня SIP (Session Initiation Protocol – протокол установления сессии). Эти протоколы не предоставляют средств по обеспечению конфиденциальности передаваемых с их помощью данных, поэтому очевидным решением проблемы защиты от прослушивания VoIP-трафика является его шифрование, а шифрование управляющих сеансом связи пакетов позволит скрыть от злоумышленника идентификационные данные абонентов. Средствами программной IP-АТС «Asterisk» такое шифрование осуществляется путем использования криптографического протокола TLS (Transport Layer Security – безопасный протокол транспортного уровня) в связке с SIP для сигнализации между двумя абонентами и использования защищенного протокола SRTP (Secure RTP – защищенный RTP) вместо RTP для передачи данных. Для этого генерируется сертификат сервера, который подписывается с помощью полученных ключа и сертификата удостоверяющего центра (или сгенерированных на стороне сервера самостоятельно в случае использования самоподписанных сертификатов), сертификат сервера соединяется с ключом сервера для получения нового сертификата сервера [1]. В результате при перехвате злоумышленником VoIP-трафика между двумя абонентами, установившими сеанс связи с использованием протоколов SIP+TLS и SRTP, вместо речевой информации он услышит шум.

Кроме использования шифрования VoIP-трафика средствами программной IP-АТС «Asterisk» также может быть применено шифрование в VPN-сетях с помощью VPN-серверов и шлюзов. Такое шифрование осуществляется на базе протокола IP-Sec, оно эффективно и снижает нагрузку на VoIP-сервер («Asterisk»), однако подразумевает наличие определенного рода ограничений структуры сети и требований к производительности ее узлов, с помощью которых непосредственно осуществляется шифрование-расшифрование пакетов [2].

Также возможно произвести многоуровневое шифрование и анонимизацию VoIP-трафика с помощью сети I2P.

Обеспечение защиты от НСД к узлам сети подразумевает надлежащую программную конфигурацию узлов сети (VoIP-серверов, прокси-серверов, программных и физических IP-телефонов) и их безопасное физическое размещение в локальной сети. Могут быть предъявлены следующие требования к безопасной программной конфигурации IP-АТС «Asterisk»:

- использование опции «alwaysauthreject=yes» при конфигурации SIP-канала, позволяющей серверу отклонять запросы аутентификации без сообщения подробностей. Таким образом в случае перебора злоумышленником учетных записей на сервере он не узнает, удалось ли ему обнаружить существующий;
- использование опции «allowguest=no» при конфигурации SIP-канала, запрещающей подключения к серверу в режиме прямых SIP-звонков или так называемых «гостевых вызовов»;
- использование опций «bindaddr=IP-адрес» и «bindport=порт», позволяющих задать статический адрес IP-АТС, по которому «Asterisk» будет принимать подключения;
- назначение абоненту добавочного номера, отличного от его имени. Большинство сканеров уязвимостей IP-АТС первоначально определяют имя зарегистрированного на сервере пользователя по такому же обнаруженному добавочному номеру;
- использование опций «deny=IP-адрес(a)» и «permit=IP-адрес(a)» при конфигурировании существующего абонента, что позволяет ограничить подключения к VoIP-серверу для конкретного пользователя только с выделенного диапазона адресов;
- использование опции «call-limit=n», позволяющей ограничить число одновременных сетевых линий для конфигурируемого абонента;
- использование опций «useragent=...» и/или «sdp session=...» при конфигурации SIP-канала, позволяющее идентифицировать VoIP-сервер «Asterisk» в сети не как IP-АТС, а в качестве какого-либо другого сетевого устройства;
- применение защиты планом маршрутизации звонков (dialplan), разделив при этом всех существующих абонентов VoIP-сервера на группы, каждой из которых разрешить исходящие звонки только по требуемым им для решения задач номерам. Для тех пользователей, вызовы которых по тем или иным причинам обрабатываются контекстом «default» (контекст по умолчанию), необходимо завершить вызов;
- отмена загрузки неиспользуемых модулей сервера «Asterisk».

Поскольку IP-АТС «Asterisk» зачастую выполняется в ОС семейства Linux, конфигурирование сетевых параметров непосредственно ОС также позволяет снизить вероятность НСД к выполняемой на ней IP-АТС. В данном случае под конфигурацией подразумевается использование нестандартных системных портов вместо широкоизвестных, отключение неиспользуемых служб (модулей), использование дискреционного разграничения доступа пользователей к запущенным службам (в том числе к службе «Asterisk»), применение аутентификации по ключу для удаленного доступа администратора к «Asterisk», например, по протоколу SSH, установка и настройка службы «fail2ban» в паре с «iptables».

Безопасное физическое размещение IP-АТС в локальной сети также защищает VoIP-сервер от НСД и, как следствие, ведет к обеспечению безопасности проходящих через него вызовов. Пример безопасного размещения IP-АТС «Asterisk» в локальной сети предприятия представлен на рис. 1.

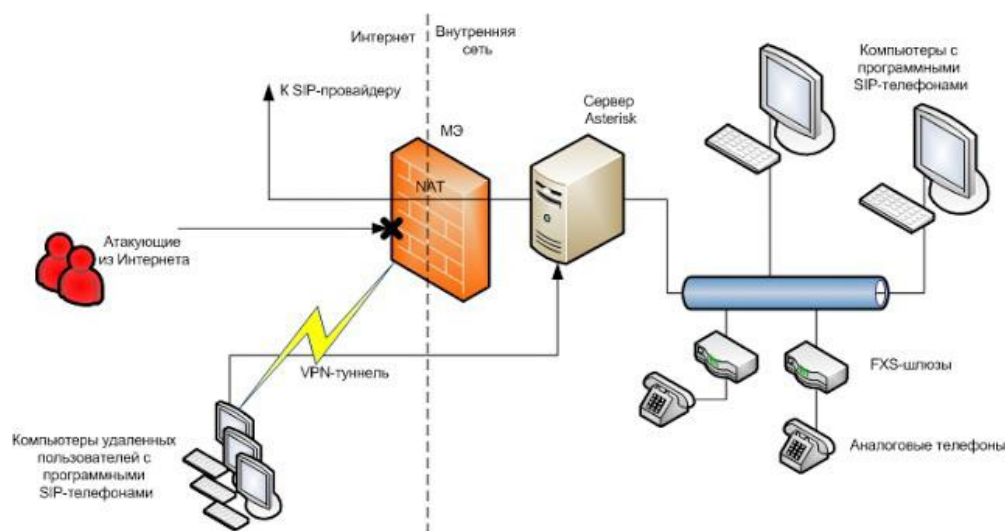


Рис. 1. Пример безопасного размещения IP-АТС «Asterisk» в локальной сети предприятия

Согласно рис. 1, используется сеть на базе межсетевого экрана с NAT (Network Address Translation – преобразование сетевых адресов), за которым размещен сервер «Asterisk». Также не рекомендуется использовать сети, построенные на концентраторах, заменив их на коммутируемые.

Применение технологии VLAN с ACL (Access Control List – список управления доступом) используется для виртуального разделения устройств различных групп в сети предприятия: абонентские устройства имеют адресацию в рамках одной подсети, тогда как серверные решения, в том числе и IP-АТС, другой. Такое разделение уменьшит вероятность НСД к последним.

Применение вспомогательных организационных защитных мер по обеспечению безопасности технологии IP-телефонии может представлять собой реализацию следующих пунктов:

- использование стойких паролей для всех учетных записей абонентов сервера;
- периодическое обновление программного обеспечения всех узлов сети;
- анализ журналов системы;
- повышение осведомленности персонала;
- смена паролей при смене администратора сети и прочее.

Таким образом, комплексная организация информационной безопасности технологии IP-телефонии, заключающаяся в обеспечении конфиденциальности потока медиа-данных, защите от НСД к узлам сети и применении организационных защитных мер, позволяет организовать защищенную VoIP-сеть на предприятии (в учреждении, организации) на базе некоммерческого программного обеспечения «Asterisk».

Список литературы

1. SIPS on Asterisk – SIP security with TLS. – URL: <http://remiphilippe.fr/sips-on-asterisk-sip-security-with-tls/>
2. Защита голосовых соединений VoIP от прослушивания. – URL: <http://www.osp.ru/lan/2003/12/138398/>

Д. В. Арефьев

Россия, Пенза, Пензенский государственный университет

СРЕДСТВА ЗАЩИТЫ АППАРАТУРЫ ПЕРЕДАЧИ ДАННЫХ ОТ УГРОЗЫ ПРОНИКНОВЕНИЯ НЕЗАШИФРОВАННОЙ ИНФОРМАЦИИ В КАНАЛ СВЯЗИ

В работе применен комплексный подход к анализу обеспечения информационной безопасности системы, в состав которой входит оконечное оборудование данных, блок сопряжения, сопрягаемая аппаратура и выход в канал связи, с применением пассивных средств защиты.

При работе с комплексом, в состав которого входит аппаратура передачи данных, наиболее острой проблемой является появление помех, которые представляют собой информативный, опасный сигнал. Информативный сигнал в данном случае представляет собой некий электрический сигнал, по параметрам которого возможно восстановить исходную конфиденциальную информацию. Особую опасность появление таких сигналов представляет для информационной системы, в которой используется шифрование.

Участком такой информационной системы является подсистема, в состав которой входит оконечное оборудование данных (далее по тексту ООД), блок сопряжения (далее по тексту БС), и сопрягаемая аппаратура, проводящая процедуру шифрования. Данная подсистема имеет выход в канал связи.

При этом следует отметить, что блок сопряжения состоит из двух частей: канальной, имеющей вывод в канал связи, и абонентской, имеющей стык с ООД. Между собой эти части взаимосвязаны.

Использование БС, позволяющего связать в единую систему ООД, сопрягаемую аппаратуру и канал связи доказало свою эффективность на протяжении долго времени. БС позволяет минимизировать денежные затраты на построение информационной системы, позволяет избавиться от построения отдельных линий связи для передачи зашифрованной и открытой информации, сэкономить рабочее пространство в силу своих размеров и габаритов, уменьшить зону контроля за информационной безопасностью информационной подсистемы, состоящей из ООД, БС и сопрягаемой аппаратуры.

Однако, выполняя столь значимую функцию, БС является наиболее уязвимым компонентом информационной системы, так как имеет в составе ряд внутренних соединений между канальной и абонентской частью, а также канал, который позволяет связать ООД и выход в канал связи, минуя сопрягаемую аппаратуру. Данный канал используется для передачи и приема незашифрованной информации. Также в составе БС имеются каналы связи, которые позволяют связать ООД и сопрягаемую аппаратуру, и сопрягаемую аппаратуру с выводом в канал связи. Данные каналы используются для передачи, как зашифрованной информации, так и незашифрованной информации.

На рис. 1 представлено движение информационных потоков в данной системе, при этом белыми стрелками показана передача открытой информации, а черными передача зашифрованной информации.

По рис. 1 можно определить, что достаточным условием проникновения незашифрованного сигнала в канал связи является либо неисправность коммутирующих устройств в составе БС в месте соединения канальной и абонентской части, что приведет к проникновению незашифрованной информации в канал связи в виде информационного сигнала, либо возникновение информативной помехи, которая

может возникнуть в результате взаимных наводок между цепями, по которым передается незашифрованный сигнал и цепями, по которым передается зашифрованный сигнал. В результате по параметрам наведенного информативного сигнала можно восстановить исходный информационный сигнал и получить доступ к конфиденциальной информации.

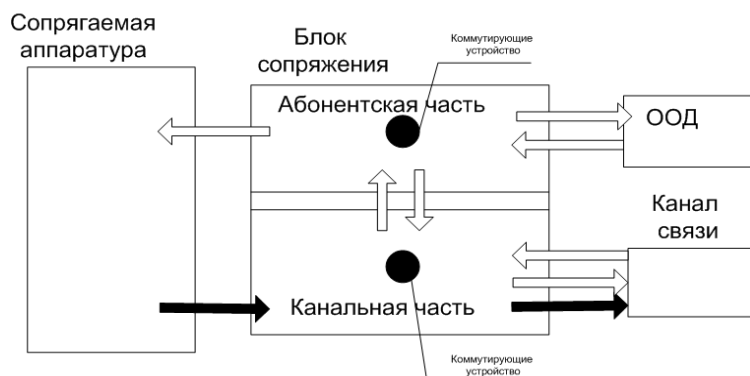


Рис. 1. Движение информационных потоков в информационной системе

Таким образом, основная задача по обеспечению безопасности данной информационной системы должна быть связана с борьбой с возникающими информативными помехами и предотвращении передачи незашифрованной информации между частями БС, в момент использования шифрования.

Средствами защиты от проникновения высокочастотной информативной помехи могут выступать фильтры нижних частот.

Фильтры предназначены для работы в качестве встроенных элементов внутреннего монтажа в радиоэлектронной аппаратуре в электрических цепях постоянного, переменного, пульсирующего, импульсного токов для подавления высокочастотных помех в определенном диапазоне. Фильтры должны соответствовать установленным в технических условиях (далее ТУ) характеристикам, в противном случае, выносится вердикт о несоответствии требованиям, обозначенным при производстве изделия БС.

Например, фильтр при проведении контрольно-измерительных испытаний должен иметь сопротивление изоляции между выводами и корпусом не менее 20 МОм, индуктивность должна быть не менее 0,05 мкГн, емкость, соответствующую значениям, установленным в инструкции по проведению измерения, тангенс угла потерь не более 0,035. Только при соответствии данным параметрам фильтр обеспечит должное затухание и как следствие подавит информативную помеху, приведет ее к виду, по которому восстановить исходный информационный сигнал будет крайне затруднительно.

При этом надо учесть, что отдельно должна быть проведена проверка по обеспечению затухания на заданных в технических требованиях частотах. Для этого может использоваться селективный микровольтметр и радиогерметичный контейнер, со встроенным в него фильтром, выводы которого подключены к выводам микровольтметра. Посредством изменения частоты на панели микровольтметра можно получить значения затухания, обеспечиваемого фильтром, на соответствующих частотах.

Если фильтры выдержали испытания и соответствуют установленным в ТУ техническим требованиям, то считается, что фильтры способны выполнять свою основную функцию, т.е. предотвращать попадание помехи, информативного сигнала в канал связи в соответствии со своими характеристиками.

Требования по борьбе с помехами требуют располагать фильтры в местах стыка между частями информационной системы, в данном случае между ООД и БС, БС и выводом в канал связи, а также между канальной и абонентской частью БС, так как именно в этих местах конструкции возможно возникновение или навод-

ка информативного сигнала, по которому можно раскрыть исходную конфиденциальную информацию.

Места стыка аппаратуры так же рекомендуется экранировать с целью уменьшения области паразитных излучений, вследствие которых возникает информативная помеха.

Для предотвращения передачи незашифрованной информации между частями БС, в момент использования шифрования, необходимо рассмотреть место в конструкции БС, где обеспечивается связь между канальной и абонентской частью. Если использовать стандартную линию связи, то в канальной и абонентской части БС потребуется использовать коммутирующие устройства, которые позволят изменить направление передачи сигнала в зависимости от использования шифрования при передаче информации. При этом вероятность отказа коммутирующих устройств должна быть крайне мала.

Однако коммутирующие устройство не является достаточным решением, так как не позволяет справиться с информативной помехой в месте коммутации, вследствие чего информативный сигнал, по параметрам которого можно восстановить исходный незашифрованный информационный сигнал, может попасть в канал связи. Поэтому, и в цепи с коммутирующим устройством используются фильтры, описанные ранее, а в месте соединения канальной и абонентской частей БС было решено включить в состав линии связи узел гальванической развязки, который позволяет исключить прямой электрический контакт за счет использования интегральных микросхем – оптронов и как следствие минимизировать возникновение информативной помехи. При этом элементы, входящие в состав оптрона, а именно излучатель – светодиод и приемник – фотодиод должны соответствовать техническим требованиям, указанным в ТУ.

Однако такие схемные решения имеют недостаток, так как при работе БС за счет паразитных излучений, возникающих на элементах, входящих в состав оптрона, может произойти наводка исходного незашифрованного сигнала на цепь, по которой передается зашифрованная информация и как следствие попадание конфиденциальной информации, в виде наведенного информативного сигнала, в канал связи. Поэтому помимо узла гальванической развязки в месте соединения канальной и абонентской части располагается система фильтров.

Источником паразитных излучений может являться цепь питания БС, которая реагирует на изменения при передаче конфиденциальной информации от ООД к сопрягаемой аппаратуре и как следствие может стать источником наведенного сигнала на цепь, по которой передается зашифрованная информация. Решением данной проблемы стало разделение источников питания для канальной и абонентской частей БС.

Полученная система обеспечения информационной безопасности с использование пассивных средств защиты, представлена на рис. 2.

Таким образом, проведено описание использования пассивных средств защиты аппаратуры передачи данных от угрозы проникновения незашифрованной информации в канал связи.

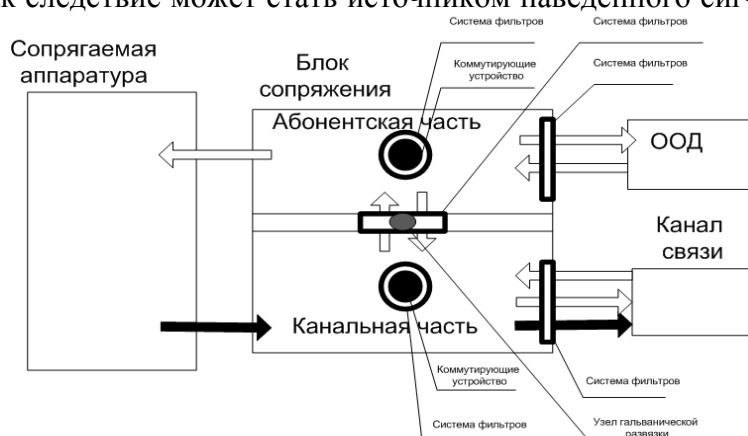


Рис. 2. Система обеспечения информационной безопасности с использованием пассивных средств защиты

Д. А. Карпушин

Россия, Пенза, Пензенский государственный университет

ПОВЫШЕНИЕ ДРУЖЕСТВЕННОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ, ИСПОЛЗУЮЩИХ СТРОГУЮ ИДЕНТИФИКАЦИЮ И АУТЕНТИФИКАЦИЮ ПОЛЬЗОВАТЕЛЯ

Рассматривается проблема снижения дружелюбности информационных систем при ужесточении политик безопасности. Предлагается использовать искусственные нейронные сети для идентификации и аутентификации по данным, допускающим незначительные ошибки.

В любой современной информационной системе (ИС), предоставляющей некоторый набор услуг, обязательно предусмотрен механизм авторизации пользователя. Под авторизацией, в широком смысле этого слова, подразумевается комплекс мероприятий по идентификации и аутентификации пользователя, с последующим предоставлением пользователю набора полномочий в данной системе.

В процессе идентификации системой выделяется конкретный пользователь с целью проведения дальнейших мероприятий по его авторизации. Идентификаторами служат: имя или псевдоним пользователя, номер карты (например, банковской), биометрические данные, номер телефона, уникальный набор букв и цифр и т.д.

После успешной идентификации начинается процесс аутентификации – проверки того, что пользователь является именно тем, за кого себя выдает, а не кем-то другим. Аутентификация, как правило, проводится с помощью одноразового или многократного пароля, с помощью токена, PIN-кода или личной биометрии аутентифицируемого. В случае успеха пользователю предоставляются определенные политики безопасности полномочия по взаимодействию с ИС.

Следует отметить, что во время ввода идентифицирующей (аутентифицирующей) информации, зарегистрированным ранее пользователем, могут возникать непреднамеренные ошибки (например, набор имени в неправильной раскладке, перепутывание соседних букв и цифр на клавиатуре и т.д.). В существующих системах контроля доступа обнаружение даже одной ошибки в предъявленных пользователем идентификационных данных служит поводом к отказу в доступе к ИС. Заметим, что если политика безопасности компании допускает использование псевдонима и пароля длиной, например, пять символов, то авторизация в ИС такой компании, несомненно, не доставит неудобств пользователю, поскольку и короткие псевдонимы, и пароли легко запоминаются, а вероятность допущения ошибок в них минимальна.

С другой стороны, при ужесточении политик информационной безопасности (генерация длинных, случайных, труднозапоминаемых паролей, еженедельная смена паролей) удобство пользования ИС снижается. Проблема усугубляется, в случае, когда пользователю необходим доступ к нескольким ИС (авторизованный доступ к 10–20 «личным кабинетам» в современном мире является нормой), а число непреднамеренных ошибок – резко возрастает.

Причем эта проблема затрагивает не только парольные системы аутентификации. Ей подвержены любые системы, где аутентификационные данные достаточно сложные и (или) большие по объему. Например, системы с биометрической аутентификацией. Причиной ошибок в случае с биометрией могут быть: естественная нестабильность образа при вводе, шумы, изменения в окружающей обстановке и чувствительность биометрического датчика.

Поэтому при разработке любой информационной системы перед ее разработчиками остро встает вопрос о выборе «рабочей точки» – компромисса между удобством пользования системой и уровнем безопасности [1].

Одно из возможных решений проблемы – использование корректирующих кодов [2]. В этом случае в идентификационные (аутентификационные) данные добавляется некоторая избыточная информация, позволяющая обнаруживать и исправлять незначительное число ошибок в данных. Однако пользователям вряд ли понравится приписывать к своему адресу электронной почты несколько проверочных символов (по аналогии с QR-кодами и штрих-кодами), а существенное удлинение случайного пароля для нахождения в нем одной единственной ошибки вряд ли сделает пароль более легким в запоминании.

Значит, следует искать другое решение, например, организовать данные на стороне сервера ИС таким образом, чтобы имелась возможность при небольшом проценте искажений входных данных как идентифицировать, так и аутентифицировать пользователя. В общем случае, когда от ИС не требуется корректировать ошибки ввода, для хранения больших наборов аутентификационных данных используют одну из нескольких общеизвестных структур данных: таблицу, хэш-таблицу, бинарное дерево поиска, красно-черные деревья и В-деревья. Эти структуры данных детально изучены и прекрасно решают возложенные на них задачи по хранению данных (рис. 1, а, б и 2).

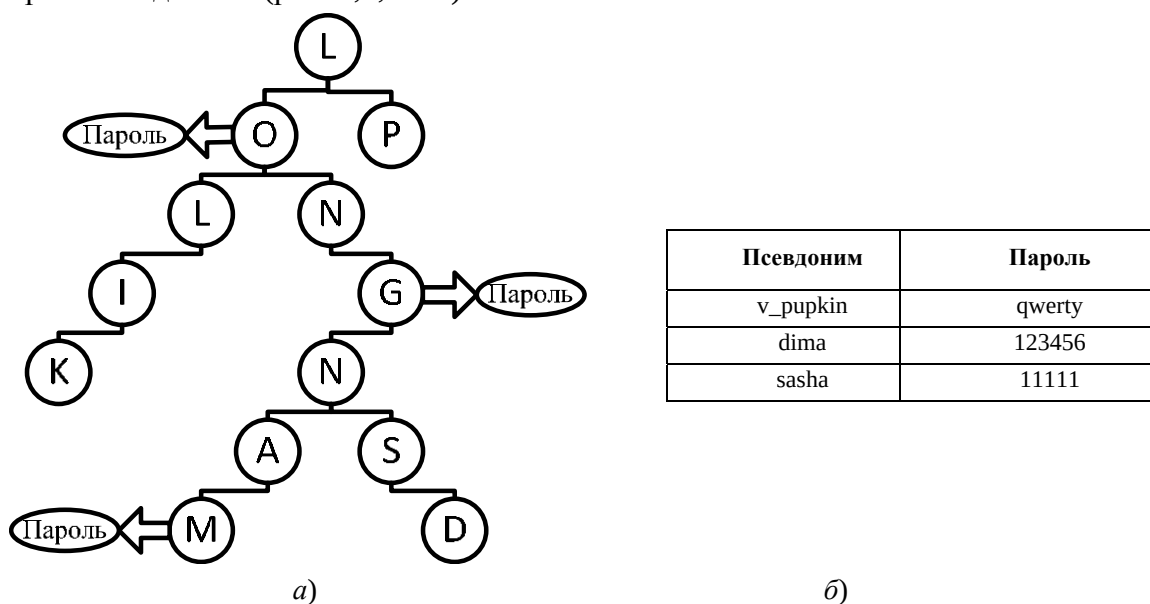


Рис. 1. Фрагмент бинарного дерева поиска (а) и таблицы учетных записей (б)

Рассмотрим следующий пример. Пусть в качестве псевдонима в нашей информационной системе будет использоваться адрес электронной почты. Допустим, что в качестве структуры хранения данных используется бинарное дерево. Обратимся к рис. 1, а. На нем представлен фрагмент бинарного дерева поиска, содержащий в себе псевдонимы пользователей рассматриваемой ИС.

Допустим, что псевдоним пользователя, пытающегося пройти авторизацию, содержит следующие символы: «LONGNAM». Выполняя операцию поиска, система будет передвигаться по дереву по одному символу за шаг, и в случае успеха найдет совпадение. Допустим, что пользователь опечатался и написал «LONGNSM». Вплоть до последнего символа алгоритм поиска будет работать успешно. Но, встретив на своем пути последний символ, алгоритм не найдет соответствующей ветви для осуществления перехода и вернет ошибку «пользователь не

найден». Аналогично можно показать, что и применение других указанных выше структур данных не позволит организовать эффективную проверку вводимых пользователем идентификационных данных, если они предполагают наличие искажений. То есть классические структуры данных могут применяться в «недружественных» к пользователю ИС, либо должны быть дополнены нетривиальными алгоритмами поиска с использованием частично измененных или неполных данных. Более того, в случае с паролями и другими аутентификационными данными прямое сравнение введенного пароля с эталонным значением зачастую не только невозможно (например, в ИС хранится только значение хэш-функции от пароля, а незначительное изменение пароля приведет к существенному изменению его хэш-значения), но и недопустимо (в силу внутренних и внешних угроз).

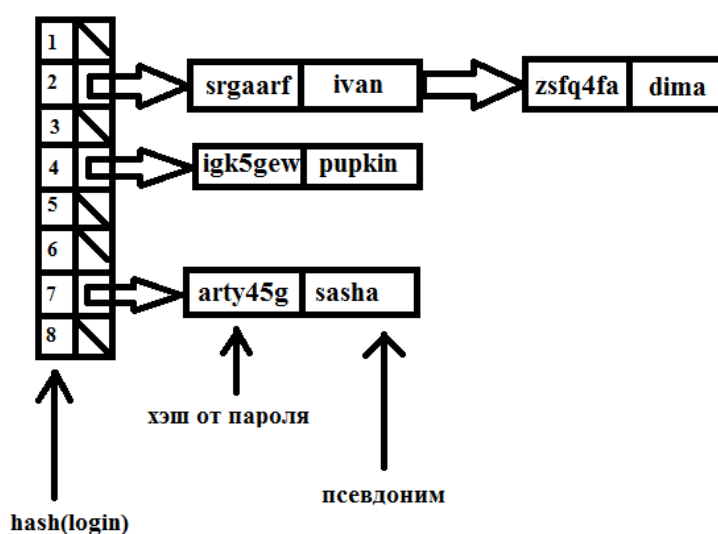


Рис. 2. Фрагмент хэш-таблицы, содержащей учетные записи

В этой связи интересным представляется подход, в котором используются отечественные алгоритмы быстрого обучения больших искусственных нейронных сетей, позволяющие обучить нейронную сеть стабильно выдавать один и тот же выходной код для близких биометрических образов типа «Свой». При этом для других биометрических образов (типа «Чужой») на выходе будет получен существенно отличающийся от проверочного по мере Хэмминга код [3]. При незначительном проценте искажений или пропусков в аутентификационных данных, искусственная нейронная сеть сможет с большой вероятностью выбрать несколько записей в таблице учетных записей ИС, наиболее соответствующих аутентифицируемому пользователю ИС, и выполнить его аутентификацию по одной из них.

Следовательно, необходимо дальнейшее исследование механизма искусственных нейронных сетей при организации безопасного поиска в больших наборах биометрических (и других) данных, допускающих ошибки или неполноту при вводе.

Список литературы

1. Руководство по биометрии / Р. Болл, Дж. Коннел. – М. : Изд-во Техносфера, 2007.
2. Обнаружение и исправление ошибок. – URL: <https://ru.wikipedia.org/wiki/>
3. Волчихин, В. И. Быстрые алгоритмы обучения нейросетевых механизмов биометрико-криптографической защиты информации : моногр. / В. И. Волчихин, А. И. Иванов, В. А. Фунтиков. – Пенза : Изд-во Пенз. гос. ун-та, 2005.

СЕТИ ЭВМ И ТЕЛЕКОММУНИКАЦИИ

Д. А. Перепёлкин, М. А. Иванчикова

Россия, Рязань, Рязанский государственный радиотехнический университет

ЗАДАЧА ФОРМИРОВАНИЯ ОПТИМАЛЬНЫХ МАРШРУТОВ ПЕРЕДАЧИ ДАННЫХ КОРПОРАТИВНЫХ СЕТЕЙ В УСЛОВИЯХ ДЕЙСТВИЯ НА ТЕРРИТОРИИ РЕГИОНА НЕСКОЛЬКИХ ПРОВАЙДЕРОВ СВЯЗИ С РАЗЛИЧНЫМИ ЗОНАМИ ПОКРЫТИЯ АБОНЕНТОВ

Рассмотрена задача формирования оптимальных маршрутов передачи данных в корпоративных сетях при действии на территории региона нескольких провайдеров связи с различными зонами покрытия абонентов.

В настоящее время большой класс распределенных сетевых структур оптимально строить на основе базовых (опорных) сетей. На сегодняшний день такие задачи актуальны при формировании региональной телекоммуникационной инфраструктуры с учетом особенностей действующих на территории региона операторов связи [1-3]. Это дает возможность сократить общие затраты на создание высокоскоростных каналов связи, построить устойчивую сетевую структуру, к которой легко подключать новые сегменты с учетом специфики предприятия. В связи с этим основное внимание уделяется эффективности применяемых в корпоративных сетях процессов маршрутизации.

Задача маршрутизации в корпоративных сетях решается при условии, что кратчайший маршрут, обеспечивающий передачу пакета на минимальное время, зависит от топологии сети, пропускной способности и нагрузки на линии связи. На практике загрузка и пропускная способность линий связи корпоративной сети динамически меняются, что может приводить к частой рассылке служебной информации об изменениях маршрутов.

В общем случае корпоративная сеть состоит из множества базовых (коммуникационных) узлов, к которым подключаются пользователи сети, соединенных скоростными каналами связи. Стоимость, или цена, маршрута складывается из стоимости используемых каналов связи и базовых узлов.

Задача формирования и построения оптимальных маршрутов передачи данных часто формулируется как задача поиска кратчайших путей на графе. При этом вершинами графа являются базовые узлы, а ребрами – каналы связи, соединяющие эти базовые узлы. В настоящее время известно большое число алгоритмов поиска оптимальных маршрутов.

Для решения задачи поиска оптимальных маршрутов в корпоративных сетях широко используется алгоритм Дейкстры. Данный алгоритм применяется для построения таблиц маршрутизации в протоколе OSPF (OpenShortestPathFirst). Характеристики и параметры качества обслуживания протокола OSPF подробно рассматриваются в работах [4, 5]. Трудоемкость построения таблиц маршрутизации с использованием классического алгоритма Дейкстры составляет величину $O(N^2)$, где N – число маршрутизаторов корпоративной сети.

В работе [6] предложен алгоритм парных переходов, позволяющий за счет сбора дополнительной информации учесть возможные изменения параметров линий связи корпоративной сети и не производить полный пересчет маршрутных таб-

лиц. Это позволило снизить трудоемкость расчета таблиц маршрутизации до величины порядка $O(k \cdot N)$, где k – число фактически выполненных парных переходов. В работах [7–10] предложены эффективные алгоритмы адаптивной ускоренной маршрутизации при динамических изменениях в структуре корпоративной сети, которые также позволили снизить трудоемкость построения оптимальных маршрутов передачи данных до величины $O(k \cdot N)$.

Наиболее эффективный алгоритм для решения задачи поиска оптимальных маршрутов в корпоративных сетях нескольких провайдеров связи предложен в работе [11]. Данный алгоритм адаптивной маршрутизации позволяет повысить эффективность функционирования корпоративных сетей нескольких провайдеров связи за счет оптимального построения маршрутов передачи данных и исключения из общей стоимости маршрута – стоимости неиспользуемых каналов и узлов связи. Однако в случае динамического изменения нагрузки на линиях связи в корпоративной сети нескольких провайдеров связи необходимо производить полный перерасчет таблиц маршрутизации.

Применение новых перспективных подходов для решения задачи адаптивной маршрутизации позволяет повысить эффективность функционирования корпоративных сетей нескольких провайдеров связи с различными зонами покрытия за счет уменьшения трудоемкости построения оптимальных маршрутов передачи данных.

Математическую модель корпоративной сети нескольких провайдеров связи с различными зонами покрытия абонентов представим в виде неориентированного взвешенного связного мультиграфа $G = (V, E, W, Z)$, где V – множество вершин (узлов связи или маршрутизаторов), $|V| = N$, E – множество ребер (каналов или линий связи), $|E| = M$, W – множество весов ребер (стоимость каналов связи между узлами), Z – множество провайдеров связи, $|Z| = m$.

Каждый узел связи v_i мультиграфа корпоративной сети также имеет свой вес $s(v_i)$. Вес узла может определяться, например, числом пользователей, которые подключаются к данному узлу. Однако в случае подключения к узлу каналов связи нескольких провайдеров он выполняет функции по коммутации этих каналов, что, как правило, требует дополнительных затрат. Поэтому вес узла должен учитывать этот фактор, что в нашем случае будет выглядеть следующим образом:

$$s(v_i) = \sum_{x=1}^m s_x(v_j),$$

где $s_x(v_j)$ – вес узла связи с номером j , связанный с обслуживанием каналов провайдера связи с номером x .

Пусть на мультиграфе G в некоторый момент времени уже решена задача поиска оптимальных маршрутов до всех узлов множества $V_s = V \setminus \{v_s\}$ из начального узла v_s , т. е. построено дерево оптимальных маршрутов с корнем в узле v_s . Обозначим это дерево как T_g .

Рассмотрим множество каналов E мультиграфа G . По признаку вхождения каналов в дерево T_g можно разделить исходное множество E на два подмножества: $E_T \in T_g$ и $E_R \notin T_g, E_T \cup E_R = E$.

Множество каналов дерева E_T – множество каналов дерева T_g для мультиграфа G . Для заданного мульти графа G согласно свойству дерева мощность множества E_T будет равняться мощности множества V минус единица $|E_T| = |V| - 1$.

Множество каналов замены для дерева E_R – множество каналов мультиграфа G , не вошедших в дерево T_g . При соответствующих условиях некоторый канал связи $e_{ij}^x \in E_R$, инцидентный узлам v_i и v_j и обслуживаемый провайдером $x \in Z$, может пе-

рейти в множество каналов дерева E_T , заменив собой некоторый канал $e_{k,p}^x \in E_T$. При этом инцидентность канала $e_{k,p}^x$ узлу v_i или v_j является обязательным условием. В свою очередь канал $e_{i,j}^x$ перейдет в множество E_R .

Будем называть такие переходы парными переходами и обозначать $e_{i,j}^x - e_{k,p}^x$.

Парные переходы возможны в двух случаях: при уменьшении веса канала связи $e_{k,p}^x$ до некоторого порогового значения и при увеличении веса канала $e_{i,j}^x \in E_T$. При этом возможен такой случай, при котором изменение веса канала связи $e_{i,j}^x \in E_T$ повлечет за собой исключение некоторого канала $e_{o,u}^x \in E_T$ и включение в это множество канала $e_{k,p}^x \in E_R$. Изменение веса канала связи $e_{k,p}^x \in E$ может привести к одному или нескольким парным переходам, но может и не привести к перестановкам, вследствие чего конфигурация дерева оптимальных маршрутов не изменится. Количество парных переходов зависит от величины изменения и положения канала связи в дереве оптимальных маршрутов (для каналов $e_{i,j}^x \in E_T$). Любой канал связи $e_{i,j}^x \in E$ может попасть в дерево T_g только посредством срабатывания парного перехода.

Для каждого канала связи $e_{i,j}^x \in E$ на шкале значений весов определены точка вхождения в дерево $w_{i,j}^{x,t}$ и точка вхождения в множество замены $w_{i,j}^{x,s}$, причем $w_{i,j}^{x,t} \leq w_{i,j}^{x,s}$, под которыми понимается максимально возможный вес канала $e_{i,j}^x$ при его вхождении в множество каналов дерева $E_T \in T_g$ и в множество каналов замены для дерева $E_S \notin T_g$ соответственно.

Маршрутная степень узла связи $ms(v_i)$ – число неповторяющихся каналов связи $e_{i,j}^x \in E$, инцидентных узлу v_i , через каждый из которых можно построить маршрут между узлами v_i и v_s .

Множество оптимальных маршрутов до узлов мультиграфа G из исходного узла $v_s - \pi_j (j = 1...N)$ определяет множество оценок оптимальных маршрутов до этих узлов – $d_j \in D_j (j = 1...N)$. Оценку длины оптимального маршрута d_j можно определить из следующего выражения:

$$d_j = \min_{x=1..m} \left\{ d_p^x \right\} + w_{p,j}^x + s_j^x,$$

где d_p^x – оценка длины маршрута до узла связи v_p , $w_{p,j}^x$ – вес канала связи, соединяющего узел v_p с узлом v_j , s_j^x – стоимость обслуживания каналов связи провайдера $x \in Z$ для узла связи с номером j .

Для разработки алгоритма адаптивной ускоренной маршрутизации в корпоративных сетях нескольких провайдеров связи с различными зонами покрытия абонентов в условиях возможных динамических изменений параметров линий связи были сформулированы и доказаны теоремы.

Применение разработанного алгоритма позволило снизить трудоемкость расчета таблиц маршрутизации до значения порядка $O(k \cdot m \cdot N)$, где k – число выполненных парных переходов, m – число провайдеров связи, N – число маршрутизаторов в сети. Таким образом, разработанный алгоритм парных переходов каналов связи позволяет сформировать оптимальные маршруты передачи данных в корпоративных сетях нескольких провайдеров связи с различными зонами покрытия в условиях динамических изменений нагрузки на их линиях связи и не производить полный пересчет маршрутных таблиц.

Работа выполнена при финансовой поддержке гранта Президента РФ для молодых ученых-кандидатов наук МК-819.2014.9.

Список литературы

1. Бурков, С. М. Алгоритмы и методы поэтапного формирования телекоммуникационных сетей региона. Математическая модель / С. М. Бурков // Вестник Тихоокеанского государственного университета. – 2008. – № 1. – С. 91–100.
2. Бурков, С. М. Постановка задачи формирования базовой сети регионального уровня / С. М. Бурков, В. А. Бертенев // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. – 2009. – Т. 4, № 82. – С. 22–27.
3. Горшков, С. Г. Задача формирования структуры базовой сети / С. Г. Горшков, Е. В. Никитин, Е. А. Саксонов // Вестник Тихоокеанского государственного университета. – 2010. – № 2. – С. 59–66.
4. Корячко, В. П. Корпоративные сети: технологии, протоколы, алгоритмы / В. П. Корячко, Д. А. Перепёлкин. – М. : Горячая линия – Телеком, 2011. – 219 с.
5. Корячко, В. П. Анализ и проектирование маршрутов передачи данных в корпоративных сетях / В. П. Корячко, Д. А. Перепёлкин. – М. : Горячая линия – Телеком, 2012. – 235 с.
6. Уваров, Д. В. Построение дерева кратчайших путей в графе на основе данных о парных переходах / Д. В. Уваров, А. И. Перепёлкин, В. П. Корячко // Системы управления и информационные технологии. – 2004. – № 4 (16). – С. 93–96.
7. Перепёлкин, Д. А. Разработка алгоритмов адаптивной маршрутизации в корпоративных вычислительных сетях / Д. А. Перепёлкин, А. И. Перепёлкин // Вестник Рязанского государственного радиотехнического университета. – 2006. – № 19. – С. 114–116.
8. Перепёлкин, А. И. Разработка алгоритма динамической маршрутизации на базе протокола OSPF в корпоративных вычислительных сетях / Д. А. Перепёлкин, А. И. Перепёлкин // Вестник Рязанского государственного радиотехнического университета. – 2009. – № 28. – С. 68–72.
9. Перепёлкин, Д. А. Алгоритм адаптивной ускоренной маршрутизации на базе протокола OSPF при динамическом добавлении элементов корпоративной сети / Д. А. Перепёлкин // Вестник Рязанского государственного радиотехнического университета. – 2010. – № 34. – С. 65–71.
10. Перепёлкин, Д. А. Алгоритм адаптивной ускоренной маршрутизации на базе протокола OSPF при динамическом отказе элементов корпоративной сети / Д. А. Перепёлкин // Вестник Рязанского государственного радиотехнического университета. – 2011. – № 37. – С. 53–58.
11. Корячко, В. П. Алгоритм адаптивной маршрутизации в корпоративных сетях нескольких провайдеров связи / В. П. Корячко, Д. А. Перепёлкин, М. А. Иванчикова // Вестник Рязанского государственного радиотехнического университета. – 2013. – № 2 (44). – С. 52–56.

Д. А. Перепёлкин

Россия, Рязань, Рязанский государственный радиотехнический университет

**МАТЕМАТИЧЕСКАЯ МОДЕЛЬ МНОГОПУТЕВОЙ АДАПТИВНОЙ
МАРШРУТИЗАЦИИ В КОРПОРАТИВНЫХ СЕТЯХ
С БАЛАНСИРОВКОЙ НАГРУЗКИ НА ОСНОВЕ ДАННЫХ
О ПАРНЫХ ПЕРЕСТАНОВКАХ МАРШРУТОВ**

Предложена математическая модель многопутевой адаптивной маршрутизации с балансировкой нагрузки, повышающая качество обслуживания пользовательских запросов и приложений в корпоративных сетях при передаче по каналам связи разнородного трафика за счет использования дополнительной информации о возможных маршрутах замены.

Развитие современных корпоративных сетей сопровождается непрерывной сменой сетевых технологий, направленной на повышение быстродействия и надежности сетей, возможностей интегрированной передачи данных, голоса и видеoinформации. Обеспечение высокоскоростного и надежного обмена информацией между узлами корпоративной сети при жестких требованиях к задержкам передачи информации в условиях возможных всплесков трафика в каналах связи является одной из важнейших проблем.

Для повышения качества обслуживания (Quality of Service, QoS) запросов различных сетевых приложений и отдельных пользователей необходимо использовать эффективные модели и алгоритмы адаптивной маршрутизации, которые непосредственно влияют на значения ключевых QoS-параметров и показателей: средней задержки передачи, джиттера, скорости передачи и значения отклонения от длины оптимального маршрута. В связи с этим особое внимание уделяется внедрению и поддержке решений многопутевой стратегии маршрутизации, использованию композитных метрик линий связи, максимально учитывающих численные значения различных QoS-показателей, а также обеспечению масштабируемости маршрутных решений, т. е. способности сохранить в заданных пределах свою эффективность в условиях роста территориальной распределенности корпоративной сети, числа и типов обслуживаемых трафиков пользователей.

Развитие в последнее время технологии программно-конфигурируемых сетей [1–3] позволяет формулировать различные задачи оптимальной маршрутизации и балансировки сетевого трафика [4, 5]. В настоящее время ведутся активные научные исследования по развитию технологии программно-конфигурируемых сетей с применением стратегии многопутевой маршрутизации. Так в работах [6–8] предложена потоковая модель многопутевой маршрутизации с балансировкой нагрузки на основе теории GERT-сетей. К недостаткам предложенной модели стоит отнести то, что для определения множества оптимальных маршрутов в данной модели необходимо использовать комбинаторные алгоритмы построения схем маршрутизации. Также в этой модели не учитываются возможности динамических изменений в структуре телекоммуникационной сети и ее параметров линий связи. В работе [9] разработан алгоритм парных переходов, позволяющий за счет сбора дополнительной информации учесть возможные изменения параметров линий связи корпоративной сети и не производить полный пересчет маршрутных таблиц. Это позволило снизить трудоемкость расчета таблиц маршрутизации до значения порядка $O(k \cdot N)$, где k – число фактически выполненных парных переходов. В работах [10–12] на основе алгоритма парных переходов предложены эффективные алгоритмы адап-

тивной ускоренной маршрутизации при динамических изменениях в структуре корпоративной сети, которые также позволили снизить трудоемкость построения оптимальных маршрутов передачи данных до значения $O(k \cdot N)$. Недостатком предложенных алгоритмов является то, что при динамических изменениях параметров линий связи и в структуре корпоративной сети после каждого парного перехода необходимо рассчитывать дополнительную информацию для того, чтобы определить оптимальный маршрут до других узлов в сети. В работах [13–18] предложен алгоритм парных перестановок маршрутов в условиях динамических изменений структуры и параметров линий связи корпоративной сети, что позволило снизить трудоемкость построения оптимальных маршрутов передачи данных до значения $O(N)$.

Разработка новых перспективных подходов для решения задачи многопутевой адаптивной маршрутизации на основе данных о парных перестановках маршрутов позволяет повысить качество обслуживания сетевых приложений и запросов пользователей корпоративных сетей с балансировкой нагрузки за счет уменьшения трудоемкости построения оптимальных маршрутов и обеспечить надежность и быстродействие передачи данных.

Математическую модель корпоративной сети представим в виде неориентированного взвешенного связного графа $G = (V, E, W, Z)$, где V – множество вершин (узлов связи), $|V| = N$, E – множество ребер (каналов или линий связи), $|E| = M$, W – множество весов ребер (стоимость каналов или линий связи), Z – множество трафиков (пользовательских приложений, служебных данных и т.д.).

Пусть на графе G в некоторый момент времени уже решена задача поиска оптимальных маршрутов до всех узлов множества $V_s = V \setminus \{v_s\}$ из начального узла v_s , т. е. построено дерево оптимальных маршрутов с корнем в узле v_s . Обозначим это дерево как T_g . Рассмотрим множество каналов E графа G . По признаку вхождения каналов в дерево T_g можно разделить исходное множество E на два подмножества: $E_T \in T_g$ и $E_R \notin T_g, E_T \cup E_R = E$.

Множество каналов дерева E_T – множество каналов дерева T_g для графа G . Для заданного графа G согласно свойству дерева мощность множества E_T будет равняться мощности множества V минус единица $|E_T| = |V| - 1$.

Множество каналов замены для дерева E_R – множество каналов графа G , не вошедших в дерево T_g . При соответствующих условиях некоторый канал $e_{ij} \in E_R$, инцидентный узлам v_i и v_j , может перейти в множество каналов дерева E_T , заменив собой некоторый канал $e_{kp} \in E_T$. При этом инцидентность канала e_{kp} узлу v_i или v_j является обязательным условием. В свою очередь, канал e_{ij} перейдет в множество E_R .

Будем называть такие переходы парными переходами и обозначать $e_{ij} - e_{kp}$.

Для каждого канала связи $e_{ij} \in E$ на шкале значений весов определены точка вхождения в дерево w_{ij}^t и точка вхождения в множество замены w_{ij}^s , причем $w_{ij}^t \leq w_{ij}^s$, под которыми понимается максимально возможный вес канала e_{ij} при его вхождении в множество каналов дерева $E_T \in T_g$ и в множество каналов замены для дерева $E_S \notin T_g$ соответственно.

Обозначим w_{ij} – маршрутная метрика канала, соединяющего узлы v_i и v_j . Узел v_i располагается ниже по иерархии в дереве оптимальных маршрутов относительно v_j . Множество E_T – множество каналов, каждый элемент которого входит, по крайней мере, в один оптимальный маршрут из начального узла связи, E_R – множество остальных каналов. $E_R \cup E_T = E$, $E_R \cap E_T = \emptyset$. Обозначим V_T – множество узлов, до которых найден оптимальный маршрут из начального узла связи, V_R – множество остальных узлов. $V_R \cup V_T = V$, $V_R \cap V_T = \emptyset$.

Обозначим множество маршрутов до узла v_i из исходного узла v_s через Π_i , где элемент множества $\pi_{i,k} \in \Pi_i$ будет множеством не повторяющихся каналов $e_{i,j} \in E$, образующих вместе маршрут, соединяющий v_s и v_i . Каждому $\pi_{i,k} \in \Pi_i$ поставим в соответствие число, равное сумме весов входящих в него каналов, т. е. длину или маршрутную метрику маршрута $d_{i,k} \in D_i$, где D_i представляет собой множество оценок оптимальных маршрутов до узла v_i из исходного узла v_s . На множестве Π_i задан селектор H , возвращающий оптимальный маршрут из множества Π_i . Оптимальный маршрут до узла v_i будем обозначать $\pi_i = H(\Pi_i)$, оценку его длины – d_i .

Каждому трафику из множества Z сопоставим ряд параметров: $w_{i,j}^z$ – маршрутная метрика z -го трафика в канале связи $(i, j) \in E$, s_z – маршрутизатор-отправитель, t_z – маршрутизатор-получатель. Для оценки доли z -го трафика, протекающего в канале $(i, j) \in E$, используем управляющую переменную $x_{i,j}^z$. В соответствии с физикой решаемой задачи на переменные $x_{i,j}^z$ накладываем следующие ограничения:

$$0 \leq x_{i,j}^z \leq 1.$$

Маршрутную метрику d_t оптимального маршрута π_t между маршрутизатором-источником и маршрутизатором-получателем пакетов данных z -го трафика определим по выражению:

$$d_t = \min_{(i,j) \in E} \sum_{z \in Z} \sum_{(i,j) \in \pi_t} w_{i,j}^z x_{i,j}^z.$$

Тогда маршрутную метрику k -го маршрута из имеющихся маршрутов между маршрутизатором-источником и маршрутизатором-получателем определим по выражению:

$$d_{k,t} = \sum_{z \in Z} \sum_{(i,j) \in \pi_{k,t}} w_{i,j}^k x_{i,j}^z, \text{ для всех } k \in K,$$

где $w_{i,j}^k$ – маршрутная метрика канала связи, соединяющего узлы с номерами i и j , и входящего в маршрут с номером k к узлу связи с номером t ; K – общее число маршрутов между маршрутизатором-источником и маршрутизатором-получателем.

Суммарную маршрутную метрику всех имеющихся маршрутов между маршрутизатором-источником и маршрутизатором-получателем определим следующим образом:

$$D_t = \sum_{k=1}^K d_{k,t}, \text{ для всех } k \in K.$$

Для выполнения балансировки нагрузки между имеющимися маршрутами определим долю информации $y_{k,t}$, проходящей по каждому из маршрутов:

$$y_{k,t} = \frac{p_{k,t}}{P_t}, \text{ где } p_{k,t} = \frac{D_t}{d_{k,t}}; P_t = \sum_{k=1}^K p_{k,t}, \text{ для всех } k \in K.$$

На переменные $y_{k,t}$ наложим дополнительные ограничения:

$$\sum_{k=1}^K y_{k,t} = 1; 0 \leq y_{k,t} \leq 1.$$

Для управления процессом балансировки трафика в корпоративной сети необходимо выполнить условие:

$$\sum_{(i,j) \in E} w_{i,j} x_{i,j}^z \leq \alpha c_{i,j}; 0 \leq \alpha \leq 1; (i,j) \in E,$$

где $w_{i,j}$ – маршрутная метрика канала связи, $c_{i,j}$ – максимально-допустимая нагрузка канала связи; α – коэффициент балансировки трафика или динамически управляемый порог максимальной загрузки каналов связи в корпоративной сети.

Для обеспечения оптимальности процесса балансировки нагрузки между разными маршрутами необходимо обеспечить минимизацию максимальной загрузки каналов связи, входящих в маршрут:

$$\min \alpha = \min_{z \in Z} \max_{(i,j) \in E} \frac{\sum_{z \in Z} \sum_{(i,j) \in E} w_{i,j} x_{i,j}^z}{c_{i,j}}.$$

При оценке качества конечных решений воспользуемся такими показателями QoS как величина отклонения от длины оптимального маршрута и средняя многопутевая задержка передачи.

В первом случае сначала определяется допустимое значение отклонения от длины оптимального маршрута, а затем перед балансировкой отбрасываются все маршруты не соответствующие данному показателю. Для этого используем следующее выражение:

$$Q(M) = \max_{\pi_k \in \Pi} \{d(\pi_k)\} - \min_{\pi_k \in \Pi} \{d(\pi_k)\}$$

Во втором случае определяется средняя задержка пакетов $\tau(\pi_k)$ вдоль маршрута π_k из множества маршрутов $\Pi = \{\pi_1; \pi_2; \dots; \pi_k; \dots; \pi_K\}$, где K – количество маршрутов между маршрутизатором-источником и маршрутизатором-получателем:

$$\tau(\pi_k) = \sum_{(i,j) \in \pi_k} \tau_{i,j}.$$

Далее выполняется расчет межконцевоймногопутевой задержки с помощью выражения:

$$T(M) = \max_{\pi_k \in \Pi} \{\tau(\pi_k)\}.$$

Оценку джиттера пакетов, вызванного реализацией многопутевой стратегии маршрутизации, т.е. разница в значениях средних задержек пакетов вдоль разных маршрутов, произведем при помощи формулы:

$$J(M) = \max_{\pi_k \in \Pi} \{\tau(\pi_k)\} - \min_{\pi_k \in \Pi} \{\tau(\pi_k)\}.$$

Для подтверждения правильности предложенной математической модели многопутевой адаптивной маршрутизации в корпоративных сетях с балансировкой нагрузки в условиях динамических изменений параметров линий связи, разработаны алгоритм и программное обеспечение моделирования процессов маршрутизации. Были проведены исследования графовых моделей корпоративных сетей, состоящих из 10, 100 и 200 узлов связи. Исследование разработанного алгоритма показало, что максимальное значение числа изменений существенно меньше размерности для каждой из рассмотренных моделей сети, а значение оценки математического ожидания не превышает единицы. Более того, обнаружена тенденция уменьшения значения оценки математического ожидания числа изменений дерева оптимальных маршрутов с увеличением количества узлов связи в сети.

Таким образом, можно сделать вывод, что предложенная математическая модель многопутевой адаптивной маршрутизации позволяет повысить качество обслуживания пользовательских приложений и различных типов трафика в корпоративных сетях с балансировкой нагрузки в условиях динамических изменений параметров линий связи за счет использования дополнительной информации о парных перестановках маршрутов.

Работа выполнена при финансовой поддержке гранта Президента РФ для молодых ученых-кандидатов наук МК-819.2014.9.

Список литературы

1. Openflow: Enabling Innovation in Campus Networks / McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J. // ACM SIGCOMM Computer Communication Review. – 2008. – Vol. 38, no. 2. – P. 69–74.
2. Смелянский, Р. Л. Программно-конфигурируемые сети / Р. Л. Смелянский // Открытые системы. – СУБД. – 2012. – № 9. – С. 18.
3. Сравнительный анализ архитектур и протоколов программно-конфигурируемых сетей / В. А. Ефимушкин, Т. В. Ледовских, Д. М. Корабельников, Д. Н. Языков // Электро-связь. – 2014. – № 8. – С. 9–14.
4. Ижванов, Ю. Л. Динамическая оценка состояния компьютерных сетей на основе метода сетевой томографии и задачи балансировки трафика / Ю. Л. Ижванов // Информатизация образования и науки. – 2013. – № 3 (19). – С. 35–40.
5. Никульчев, Е. В. Динамическое управление трафиком программно-конфигурируемых сетей в облачной инфраструктуре / Е. В. Никульчев, С. В. Паяин, Е. В. Плужник // Вестник Рязанского государственного радиотехнического университета. – 2013. – № 3 (45). – С. 54–57.
6. Шибанов, А. П. Моделирование агрегированного телекоммуникационного канала с технологией открытых потоков / А. П. Шибанов, В. П. Корячко, Ю. Л. Ижванов // Радиотехника. – 2012. – № 3. – С. 109–112.
7. Оптимизация сетей с дозированной балансировкой нагрузки и пиринговыми каналами / Ю. Л. Ижванов, В. П. Корячко, А. П. Шибанов, А. Н. Сапрыкин, О. В. Лукьянов // Вестник Рязанского государственного радиотехнического университета. – 2013. – № 1 (43). – С. 67–74.
8. Корячко, В. П. Нахождение скрытого параллелизма протоколов для улучшения характеристик сети передачи данных полигонного измерительного комплекса / В. П. Корячко, О. В. Лукьянов, А. П. Шибанов // Вестник Рязанского государственного радиотехнического университета. – 2014. – № 47. – С. 68–75.
9. Уваров, Д. В. Построение дерева кратчайших путей в графе на основе данных о парных переходах / Д. В. Уваров, А. И. Перепёлкин, В. П. Корячко // Системы управления и информационные технологии. – 2004. – № 4 (16). – С. 93–96.
10. Перепёлкин, Д. А. Разработка алгоритмов адаптивной маршрутизации в корпоративных вычислительных сетях / Д. А. Перепёлкин, А. И. Перепёлкин // Вестник Рязанского государственного радиотехнического университета. – 2006. – № 19. – С. 114–116.
11. Перепёлкин, Д. А. Алгоритм адаптивной ускоренной маршрутизации на базе протокола OSPF при динамическом добавлении элементов корпоративной сети / Д. А. Перепёлкин // Вестник Рязанского государственного радиотехнического университета. – 2010. – № 34. – С. 65–71.
12. Перепёлкин Д. А. Алгоритм адаптивной ускоренной маршрутизации на базе протокола OSPF при динамическом отказе элементов корпоративной сети / Д. А. Перепёлкин // Вестник Рязанского государственного радиотехнического университета. – 2011. – № 37. – С. 53–58.
13. Корячко, В. П. Алгоритм парных перестановок маршрутов в корпоративных сетях / В. П. Корячко, Д. А. Перепёлкин, А. И. Перепёлкин // Системы управления и информационные технологии. – 2010. – Т. 40, № 2. – С. 51–56.

14. Корячко, В. П. Повышение эффективности функционирования корпоративных сетей при динамических изменениях в их структуре и нагрузках на линии связи / В. П. Корячко, Д. А. Перепёлкин, А. И. Перепёлкин // Вестник Рязанского государственного радиотехнического университета. – 2010. – № 33. – С. 49–55.

15. Перепёлкин, Д. А. Алгоритм адаптивной ускоренной маршрутизации в условиях динамически изменяющихся нагрузок на линиях связи в корпоративной сети / Д. А. Перепёлкин, А. И. Перепёлкин // Информационные технологии. – 2011. – № 3. – С. 2–7.

16. Перепёлкин, Д. А. Алгоритм парных перестановок маршрутов на базе протокола OSPF при динамическом подключении узлов и линий связи корпоративной сети / Д. А. Перепёлкин // Вестник Рязанского государственного радиотехнического университета. – 2013. – № 4–1 (46). – С. 67–75.

17. Перепёлкин, Д. А. Алгоритм парных перестановок маршрутов на базе протокола OSPF при динамическом отказе узлов и линий связи корпоративной сети / Д. А. Перепёлкин // Вестник Рязанского государственного радиотехнического университета. – 2014. – № 47. – С. 84–91.

18. Перепёлкин, Д. А. Динамическое формирование структуры и параметров линий связи корпоративной сети на основе данных о парных перестановках маршрутов / Д. А. Перепёлкин // Информационные технологии. – 2014. – № 4. – С. 52–60.

А. О. Семёнов, Н. Н. Коннов

Россия, Пенза, Пензенский государственный университет

ФОРМАЛЬНОЕ ПРЕДСТАВЛЕНИЕ СТОХАСТИЧЕСКОГО АЛГОРИТМА ДИСПЕТЧЕРИЗАЦИИ ОЧЕРЕДЕЙ В ТЕЛЕКОММУНИКАЦИОННОМ ОБОРУДОВАНИИ

Приводится формальное описание стохастического алгоритма диспетчеризации очередей в интеллектуальных коммутаторах и маршрутизаторах с поддержкой качества обслуживания.

Современные телекоммуникационные устройства (коммутаторы и маршрутизаторы) обладают комплексом механизмов, обеспечивающих поддержку требуемого качества обслуживания (QoS) различных типов трафика, которое описывается такими параметрами, как пропускная способность (полоса пропускания), задержка при передаче пакета и ее вариация, потери пакетов [1].

Среди механизмов обеспечения качества (маркировка и классификация трафика, обслуживание очередей, управление перегрузкой, управление полосой пропускания) важнейшим является алгоритм диспетчеризации очередей, т.е. правила выбора очереди для считывания из нее данных и объем считываемых данных на одно обращение, которые непосредственно определяют задержку пакетов и вероятность их потери [2,3].

Классические циклические (WRR и DRR) и «справедливые» обслуживания (WFQ) алгоритмы обеспечивают гарантируемую полосу пропускания и предсказуемый уровень средней задержки пакетов. Однако эти алгоритмы чисто детерминированы, что приводит к неэффективному использованию полосы, если реальный трафик не соответствует принятому шаблону полосы пропускания.

В настоящей работе предлагается стохастический алгоритм диспетчеризации, основанный на стохастическом подходе к выбору очередей на обслуживание с адаптацией к длинам очередей и текущему трафику в отдельных классах QoS.

В [4] был предложен вариант стохастического алгоритма, при котором каждая очередь имеет пакеты одинаковой длины.

Предложим другой вариант стохастического алгоритма, при котором классификатором пакеты в телекоммуникационном устройстве (коммутатор или маршрутизатор) пакеты из входящего трафика распределяются классификатором в соответствии с их классом по n очередям. Устройство выборки пакета осуществляет коммутацию между i -й очередью на основании решений управляющего блока. Управляющий блок принимает решения о выборе очереди на обслуживание, исходя из некоей информации о состоянии очередей, например об их длине, и состоянии генератора псевдослучайных чисел. Обобщенная структура алгоритма отображена на рис. 1.

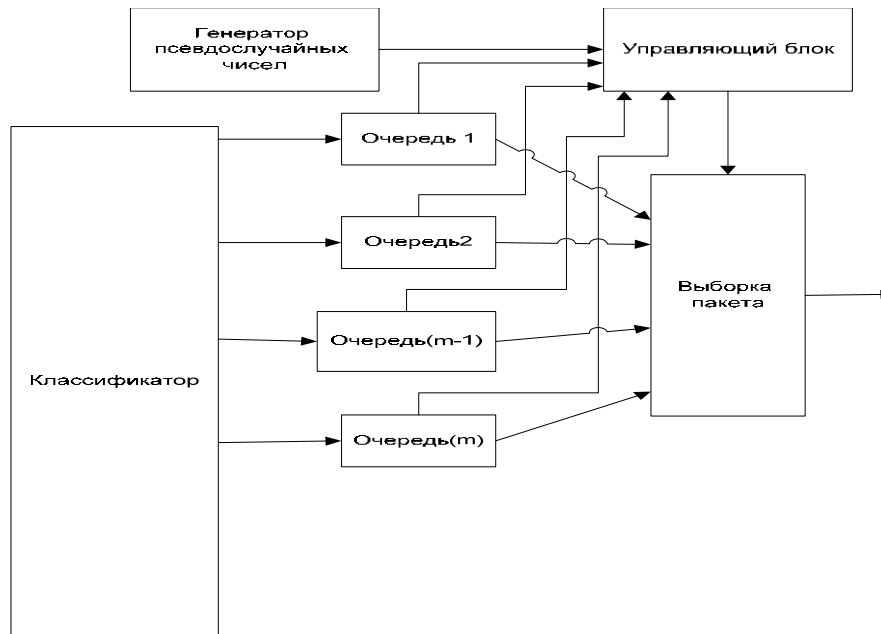


Рис. 1. Обобщенная структура алгоритма

Что формально может быть описано в виде высказывания:

$$\left. \begin{aligned} & \forall \left(t \in NN; NN \equiv (0, 1, \dots, \infty); 0 \leq i \leq m-1; i, m, n \in N; N \equiv (1, 2, \dots, \infty); n \leq m; l_i(t) \geq 0; \right. \\ & \left. p_{por_i}, p_i \in [0; 1] \right) \exists \end{aligned} \quad (1)$$

$$\left. \begin{aligned} & \exists \left(\sum_{i=0}^{m-1} \left(n p_{por_i} k_i \frac{l_i(t)}{\sum_{j=0}^{m-1} l_j(t)} \right) = 1; \sum_{i=0}^{m-1} p_i = 1; \sum_{i=0, t}^{m-1} p_i = 1; \lim_{t \rightarrow \infty} \frac{\sum p_i(t)}{t} = p_{por_i} \right), \end{aligned} \right\}$$

где t – автоматное время; NN – множество неотрицательных натуральных чисел (non-negative integers set); i – номер текущей очереди; m – количество очередей; N – множество натуральных чисел (natural numbers set).

Рассмотрим теперь данный алгоритм с точки зрения основных характеристик (параметров) обслуживания, а именно полоса пропускания (bandwidth(BW)), задержка при передаче пакета (delay), колебание (вариация задержки) при передаче пакетов, потеря пакетов (packetloss) и другие, с классической теорией обеспечения качества обслуживания не связанные, такие как вычислительная сложность алгоритма обслуживания очередей и т.п.

Следует также ввести соглашение по единицам измерения. Так, если в этой и (или) последующих статьях не указана единица измерения полосы пропускания, то в качестве эталона возьмем бит/с, входного или выходного относительно сетевого устройства (если иное не указано в тексте) трафика, а также длины пакетов, очередей и т. д. – в битах, время рассматривать как безразмерную величину, характеризующуюся номером такта основного синхросигнала.

Пусть имеется средняя и мгновенная полосы пропускания канала, равные B и b соответственно. Пусть имеется необходимость диспетчеризации потоков данных, идущих через m очередей. Физически каждая очередь представляет собой оную некоего сетевого устройства (очередь на выход из порта конечного клиента, маршрутизатора и т.д.), разделенную также по классам трафика. Каждый класс выходного трафика R_{out_i} одного устройства требует обустройства отдельных очередей на этапе классификации.

$$R_{out_i} = \sum_t b_i(t), \quad (2)$$

$$\left((E_i = 0 \wedge E_i \in BoolSet) \vee (l_i(t) = 0) \right) \supset (b_i(t) = 0), \quad (3)$$

$$(E_i = 1 \wedge E_i \in BoolSet \wedge (l_i(t) > 0)) \supset (b_i(t) \neq 0; b_i(t) \in const), \quad (4)$$

$$\left((E_i = 1 \wedge E_i \in BoolSet \wedge (l_i(t) = 0)) \vee ((q_i(t) - d_i(t) \leq 0) \wedge q_i(t) \in N \wedge d_i(t) \in N) \right) \supset (E_i(t+1) = 0), \quad (5)$$

$$\exists (p_i(t)) \left(E_j \left(t E_i(t+1) = \overline{E_i(t)} \right) = 1 \right), \quad (6)$$

где R_{out_i} – объем выходного относительно сетевого устройства трафика; b_i – мгновенная полоса пропускания i -го канала; E_i – сигнал разрешения выхода пакетов из сетевого устройства в канал; $BoolSet$ – множество булевых значений (содержит только 0 и 1, однако его можно и расширить, например, за счет введения Z-состояния, неопределенного (undef)состояния и т. д.), t – текущий момент времени, q_i – текущая величина квоты (не путать с квотой в DRR), d_i – длина пакета, находящегося в i -й очереди вторым от ее головы (queuehead), $t E_i(t+1) = \overline{E_i(t)}$ – момент времени, соответствующий намечаемой смене активной (с атрибутом E_i , равным единице или другому активному сигналу) очереди, $const$ – множество целых неотрицательных неизменяемых арифметическими и иными операциями значений. Стоит сделать замечание по формуле (4). При передаче каждого из Ethernet-пакетов тратится на сторонние нужды двадцать тактов процессорного времени, однако с точки зрения модели такого рода затраты включаются в длину пакета.

Стоит отметить, что число m в случае масштабирования становится достаточно большим. При большом числе m требуется большее количество вычислений в управляющем блоке. Подробности построения управляющего блока, а также полная структура устройства, моделирующего алгоритм будут представлены далее. Однако некоторые детали сего действия могут быть высказаны здесь:

1) С увеличением m увеличивается и количество умножителей в управляющем блоке (их количество $m - 1$).

2) Существенная часть вычислений в управляющем блоке параллельна, следовательно масштабирование системы не создаст существенных проблем, связанных с уменьшением производительности управляющего блока при увеличении m .

3) Пересчет вероятностей (и коэффициента k_i) производится либо каждые $d \leq d_{\min}$ тактов, где, например, $d_{\min} = 64$ для Ethernet-протокола, либо за не-

сколько тактов до завершения какой-либо очередью рассматриваемого устройства передачи пакетов на вход выходного канала, причем пакет, передаваемый в момент времени $t - C_r$.

$$\exists t \left(\left(\forall (k \in NN) \exists (t_r = k d_{\min}) \right) \supset \neg \left(\forall (C_r \in \text{const}) \exists \left(\left(t_r = t_{E_i}(t+1) = \overline{E_i(t)} - C_r \right) \wedge \left(\bigwedge_{j \neq i} (E_j(t) = 1) \supset \neg (E_j(t) = 0) \right) \right) \right) \right), \quad (7)$$

где k – номер такта пересчета при пересчете через фиксированное основное автоматное время; t_r – автоматное время пересчета; C_r – константа пересчета (количество тактов, необходимых для пересчета управляющим блоком вероятностей обслуживания той или иной очереди). Достоинство применения левой части выражения при моделировании и реализации алгоритма заключается в относительной простоте реализации и несущественном выигрыше в цене конечного оборудования. Потому считаю нужным настоятельно рекомендовать применение правой части выражения в реализации алгоритма.

4) Ни одно из вышеизложенных в виде формул высказываний не обеспечивает отсутствия монополизации трафика. Потому внесем изменения в (1) с учетом ограничений по исходящему трафику.

$$\left(\forall \left(t \in NN; t < t_{\max}; NN \equiv (0, 1, \dots, \infty); 0 \leq i \leq m-1; i, m, n \in N; N \equiv (1, 2, \dots, \infty); n \leq m; \right. \right. \\ \left. \left. l_i(t) \geq 0; p_{por_i}, p_i \in [0; 1] \right) \right) \exists \left(\begin{aligned} & \sum_{i=0}^{m-1} \left(n p_{por_i} k_i \frac{l_i(t) (R_{\max\_out_i} - R_{out_i})}{\sum_{j=0}^{m-1} (l_j(t) (R_{\max\_out_i} - R_{out_i}))} \right) = 1; \sum_{i=0}^{m-1} p_i = 1; \sum_{i=0, t}^{m-1} p_i = 1; \\ & \lim_{t \rightarrow \infty} \frac{\sum p_i(t)}{t} = p_{por_i} \end{aligned} \right), \quad (8)$$

где $t_{\max}$ – максимальное время работы системы.

Время относительно периодической перезагрузки системы:

$t = t_{\Sigma} \bmod t_{\max}$, где t_{Σ} – автоматное время без учета периодического сброса.

Вышеприведенное формальное представление стохастического алгоритма диспетчеризации было использовано при разработке модели алгоритма в системе Simulink. Результаты моделирования показали работоспособность стохастического алгоритма диспетчеризации и его преимущества по сравнению с известным алгоритмом DRR.

Список литературы

1. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб. : Питер, 2010.
2. Вегешна, Ш. Качество обслуживания в сетях IP / Ш. Вегешна. – М. : Вильямс, 2003.

3. Кучерявый, Е. А. Управление трафиком и качество обслуживания в сети Интернет / Е. А. Кучерявый. – СПб. : Наука и Техника, 2004.
4. Конов, Н. Н. Исследование проблемы эффективного использования ресурсов вычислительной сети и сравнительный анализ алгоритмов диспетчеризации очередей обслуживания / Н. Н. Конов, А. О. Семёнов. – Уфа, 2014.
5. Феллер, В. Введение в теорию вероятностей и ее приложения / В. Феллер. – М. : Финансы и статистика, 1984.

А. О. Семёнов

Россия, Пенза, Пензенский государственный университет

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СТОХАСТИЧЕСКОГО АЛГОРИТМА ДИСПЕТЧЕРИЗАЦИИ ОЧЕРЕДЕЙ В ТЕЛЕКОММУНИКАЦИОННОМ ОБОРУДОВАНИИ

Приводится обобщенное описание процесса моделирования стохастического алгоритма диспетчеризации с использованием системы имитационного моделирования Simulink.

Одним из важных компонентов обеспечения качества обслуживания (QoS) в современных компьютерных сетях является механизм управления очередями, который во многом определяет задержку пакетов и вероятность их потери [2]. Нами было предложено применять стохастический алгоритм, который использует специальные процедуры для отправки пакетов.

Для изучения и разработки методов обеспечения QoS в телекоммуникациях находят применение методы имитационного моделирования [1]. В качестве системы имитационного моделирования в работе была выбрана система Simulink из программного математического пакета MATLAB.

Система Simulink изначально предназначена для моделирования динамических систем, модели которых представляются из отдельных блоков. Стохастический алгоритм диспетчеризации был изначально представлен в виде набора соединенных между собой блоков.

Обобщенная структура модели такова:

- 1) Управляющий блок, включающий в себя:
 - умножители;
 - делители;
 - сумматоры;
 - устройства, обеспечивающие условное поведение алгоритма (согласно условиям);
 - компараторы;
 - другие логические элементы.
- 2) Операционные блоки, включающие в себя:
 - очереди;
 - устройства для сбора статистики;
 - часть средств синхронизации хранится также операционных блоках.

- 3) Генератор случайных чисел.
- 4) Средства синхронизации, устраняющие конфликты при попытках доступа к разделяемому ресурсу.
- 5) Блок пользовательских констант (настройки).
- 6) Устройства вывода пользовательской информации (Score).

На рис. 1 приведен внешний вид получившейся модели.

Помимо разработки модели каркаса была также проделана работа по анализам результатов моделирования.

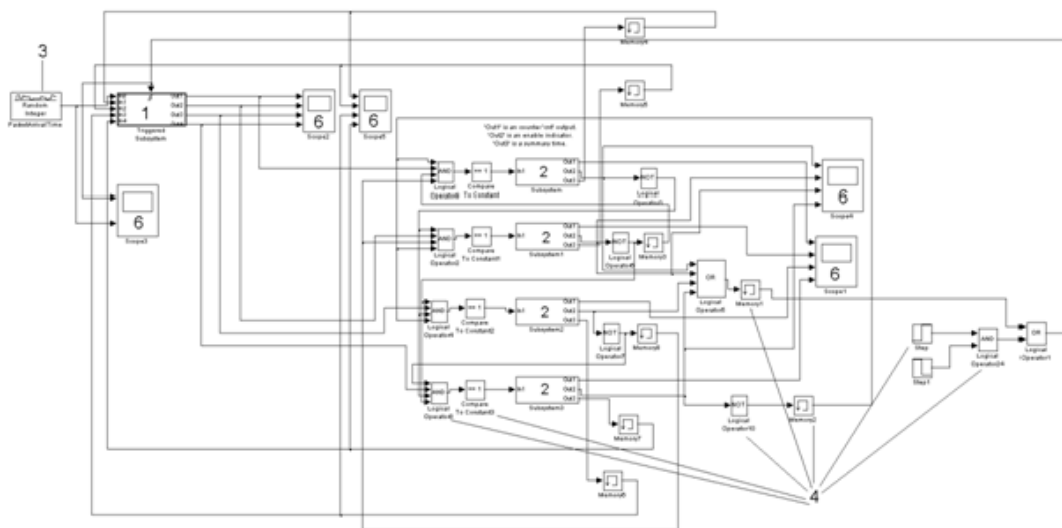


Рис. 1. Обобщенная структура стохастического алгоритма диспетчеризации

Моделирование стохастического алгоритма производилось со следующими параметрами:

- 1) Размер максимального трафика в период между двумя соседними по времени сбросами определен как равный для всех четырех классов трафика (для всех очередей) (25000/25000/25000/25000 бит на первый класс трафика, 25000 на второй и т.д.).
- 2) Количество очередей в системе равно четырем;
- 3) Приоритеты очередей 0,5/0,35/0,1/0,05 для первой, второй, третьей и четвертой очередей, соответственно.
- 4) Распределение входящего по каждой из четырех очередей трафика – равномерное, с равными для каждой очереди семенами (seed = 37) [1, 4].
- 5) Результаты моделирования такой системы таковы:
- 6) Максимальная загрузка очередей увеличивается с номером очереди.
- 7) Максимальная задержка пакетов увеличивается с номером очереди и с течением времени скачкообразно не меняется.

Для сравнения эффективности также проведено моделирование с алгоритмом классического диспетчеризации очередей DRR[2].

Анализ результатов моделирования приведен ниже.

- 1) Средняя (за большое время моделирования) полоса пропускания при использовании стохастического алгоритма существенно не отличается от алгоритма DRR [2]. Разница между значениями полос пропускания за отдельные промежутки

времени (за небольшие, но и немалые) меньше при использовании стохастического алгоритма диспетчеризации, чем при использовании алгоритма DRR.

2) Сумма длин очередей при использовании стохастического алгоритма диспетчеризации существенно не отличается от случая использования алгоритма DRR. Длины отдельных очередей в случае использования стохастического алгоритма диспетчеризации существенно отличаются в зависимости от заданного приоритета. Длины очередей с меньшим приоритетом имеют в большинстве случаев большую длину. Длины очередей с большим приоритетом имеют в большинстве случаев меньшую длину. Циклический дефицитный алгоритм, в свою очередь, в зависимости от приоритета не имеет вовсе.

3) В стохастическом алгоритме диспетчеризации косвенно предусмотрена зависимость длины очереди от величины исходящего трафика очереди. Циклический дефицитный алгоритм, в свою очередь, в зависимости от величины исходящего трафика не имеет вовсе.

4) Стохастический алгоритм диспетчеризации не изменяет длины пакетов. Алгоритм DRR также не изменяет длины пакетов.

5) Стохастический алгоритм диспетчеризации обеспечивает, как правило, более частую смену номера обслуживаемой очереди, чем при использовании циклического дефицитного алгоритма. Исходя из сказанного, как правило, энергопотребление управляющего блока существенно выше при использовании стохастического алгоритма диспетчеризации, чем при использовании циклического дефицитного алгоритма.

Стоит отметить, что моделирование производилось при одинаковых входных наборах данных, одинаковых распределениях и т.д.

Достоинства стохастического алгоритма:

- адаптивность к изменяющимся параметрам трафика (длинам очередей, предельно допустимому трафику, приоритетам очередей) подтвердилась;
- колебания величин оказываются меньшими по сравнению с аналогами;
- механизм задания приоритетов обеспечивает относительную приоритезацию, не абсолютную.

Таким образом, был промоделирован стохастический алгоритм диспетчеризации и проведено сравнительное моделирование алгоритма с алгоритмом DRR. Достоинства и недостатки алгоритма перечислены.

Список литературы

1. Дьяконов, В. Simulink4. Специальный справочник / В. Дьяконов. – СПб. : Питер, 2002.
2. Вегешна, Ш. Качество обслуживания в сетях IP / Ш. Вегешна. – М. : Вильямс, 2003.
3. Кучерявый, Е. А. Управление трафиком и качество обслуживания в сети Интернет / Е. А. Кучерявый. – СПб. : Наука и Техника, 2004.
4. Феллер, В. Введение в теорию вероятностей и ее приложения / В. Феллер. – М. : Финансы и статистика, 1984.

А. В. Сакерин, А. С. Родионов, Д. А. Мигов

Россия, Новосибирск, Сибирский государственный университет
телекоммуникаций и информатики

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ В РАСЧЕТЕ ПОКАЗАТЕЛЕЙ НАДЕЖНОСТИ СЕТЕЙ

Рассматривается задача эффективного распараллеливания алгоритма полного перебора, применяемого при расчете некоторых характеристик надежности сетей с ненадежными элементами, а также в задачах комбинаторной оптимизации. Описываются общие подходы и конкретные алгоритмы, позволяющие эффективно использовать вычислительные ресурсы при решении задач указанных классов.

Многие алгоритмы анализа и оптимизации структур, моделируемых случайными графами (СГ), в частности при рассмотрении их надежности, требуют полного перебора возможных реализаций этих случайных графов. В большинстве случаев, вычисление целевого функционала возможно независимо для каждой реализации, что позволяет естественное распараллеливание процесса вычислений. В работе представлены обобщения, позволяющие выбор наилучшего способа распараллеливания с учетом мощности и архитектуры инструментальных вычислительных средств. В качестве примеров рассматриваются задачи вычисления: вероятности связности подмножества вершин (k -терминальная связность); средней вероятности парной связности; k -терминальной связности при ограничении количества ребер в соединяющих путях (на диаметр графа).

Вычисление характеристик случайных графов

Будем говорить о вычислении математического ожидания произвольной метрической функции графа G . Рассматривается следующая модель случайного графа.

$G(n; m) = (X; U)$ – неориентированный граф с множеством вершин X и множеством ребер U .

$n = |X|$ – число вершин графа.

$m = |U|$ – число ребер графа.

p_{ij} – вероятность того, что ребро e_{ij} исправно.

$F(G)$ – метрическая функция графа.

$\bar{F}(G)$ – математическое ожидание функции $F(G)$.

G/e – граф, полученный из G стягиванием конечных вершин ребра e .

$G \setminus e$ – граф, полученный из G удалением ребра e .

Ребра выходят из строя (отказывают) независимо друг от друга.

Если число вершин и ребер графа ясно из контекста, граф обозначается как просто G . Множество вершин (ребер) конкретного графа G обозначается как $X(G)$ ($U(G)$).

Согласно этой модели, в общем случае имеем:

$$\bar{F}(G) = \sum_{H \in \Gamma} P(H) F(H)$$

В этой формуле Γ – множество всех возможных реализаций графа G .

При вычислении характеристик связности (вероятность связности всех или подмножества вершин, средней вероятности парной связности, среднего размера максимального связного подграфа и др.) большей эффективностью часто обладает метод факторизации [1, 2], при котором для произвольно выбранного ребра e_i :

$$\bar{F}(G) = p_i \bar{F}(G/e_i) + (1 - p_i) \bar{F}(G \setminus e_i)$$

При вычислении средней вероятности парной связности можно использовать и формулу получения взаимнооднозначно связанного с ней математического ожидания числа несвязных пар вершин:

$$N(G) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n a_{ij} w_i w_j$$

где w_i – вес вершины, ожидаемое число стянутых в нее вершин при применении на предварительных этапах редукции висячих вершин, факторизации по формуле (2) или ветвления по цепям [3],

$$\bar{R}(G) = \frac{C_n^2 - N(G)}{C_n^2}$$

Существенное ускорение вычисления некоторых характеристик связности (при сохранении уровня алгоритмической сложности) можно достичь заменой перебора вариантов разрушения графа перебором вариантов разрушения его покрывающего дерева [4]. Основная формула получения вероятности связности графа в этом алгоритме имеет вид:

$$R(G) = R(T_0) + \sum_{i=1}^{N-1} \sum_{j=1}^{C_{N-1}^i} P_s(E_i^j) I(H_i^j),$$

где $P_s(E_i^j)$ есть вероятность удаления из T_0 ребер j -м способом; H_i^j – соответствующий псевдограф связей компонент разрушенного дерева; $I(G)$ – индикаторная функция связности графа. Рассматриваются все возможные сочетания i ребер.

Удаление из T_0 ребер приводит к его распаду на $i + 1$ часть (блок). Каждый из этих блоков содержит подмножество вершин графа, связанных между собой. Связность графа в целом может осуществиться только за счет ребер, не входящих в T_0 . Ребро между двумя вершинами графа компонент не существует только если множество ребер графа $G \setminus T_0$, соединяющих вершины соответствующих блоков, не пусто. Пусть между парой блоков существуют k ребер с вероятностями присутствия $p_1, p_2, \dots, p_k$. Тогда вероятность присутствия соответствующего ребра графа H есть $p = 1 - \prod_{i=1}^k (1 - p_i)$.

При использовании этого подхода для вычисления математического ожидания числа несвязных пар вершин СГ имеем:

$$N(G) = \sum_{i=0}^{N-1} \sum_{j=1}^{C_{N-1}^i} P_s(E_i^j) N(H_i^j)$$

Вес вершины псевдографа равен сумме весов вершин соответствующих поддереьев.

Организация вычислений

При всей кажущейся простоте и «естественности» распараллеливания приведенных выше формул, в основном связанного с получением частичных сумм независимо вычисляемых значений, существует немало «подводных камней», скрывающихся за этой простотой.

При использовании рекурсивной формулы (2) фактически строится дерево разбора, листья которого соответствуют различным вариантам разрушения графа. Преимущество использования метода факторизации заключается в возможности прекращения ветвления по отдельным ветвям в случае существования метода непосредственного вычисления функционала.

«Естественная» реализация метода, когда при рекурсии одна из ветвей обрабатывается процессом-мастером на текущем ядре, тогда как делается попытка отправить альтернативный вариант на свободный процесс на другом ядре с последующим ожиданием результата, очевидно неэффективна в силу существенного разброса времени вычисления по ветвям. От ожидания легко избавиться передавая подчиненному процессу не только граф, но и вероятность его получения с последующей отправкой результата, в случае окончания ветвления, процессу, осуществляющему суммирование. В любом случае, на начальном этапе невозможно загрузить все ядра и чем больше ядер выделено для задачи, тем больше времени пройдет до их полной загрузки.

При использовании формулы (1) проблема загрузки вычислительных ядер не существует, достаточно распределить между ними варианты разрушения графа. Однако, тут также необходимо решить несколько задач.

«Естественное» решение заключается в разделении поровну вариантов с удалением 1,2, 3,... ребер. Это требует построения систематического генератора расстановок k единиц по m местам в маске разрушения. Генератор должен обеспечить равномерность распределения масок по вычислительным ядрам.

Вместе с тем, общее число вариантов разрушения графа $G(n, m)$ равно $N = 2^m$ и каждое из чисел $0, 1, \dots, N$ в двоичном m -разрядном представлении соответствует одной из возможных масок разрушения. Таким образом, достаточно равномерно распределять между ядрами эти числа.

При вычислении некоторых характеристик СГ, например математического ожидания числа несвязных пар вершин, происходит накопление сумм значений, которые сами получаются с помощью параллельных алгоритмов.

Распределение пар вершин или вариантов разрушения происходит не между ядрами, а между группами ядер, которые используются для применения параллельного алгоритма вычисления соответствующего функционала. Очевидно, настройка такого двойного параллельного алгоритма является нетривиальной задачей, требующей проведения большого количества численных экспериментов.

Заключение

Показаны основные проблемы организации параллельного исполнения полного перебора в некоторых комбинаторных задачах теории графов. Предложены некоторые общие подходы, применимые к различным задачам и позволяющие балансировать нагрузку при использовании большого количества вычислительных ядер при минимизации обменов и синхронизации между ними. Дальнейшие исследования связаны с выбором алгоритмов и их параметров при использовании различных архитектур инструментальных ЭВМ, а также с построением имитационных моделей параллельных программ решения рассматриваемых задач для целей анализа их масштабируемости.

Стоит ожидать, что подобно моделям, рассмотренным в [5], такое моделирование приведет к выявлению узких мест алгоритмов при их отображении на ЭВМ с очень большим числом вычислительных ядер (сотни тысяч и миллионы).

Список литературы

1. Colbourn, Ch. J. The combinatorics of network reliability / Ch. J. Colbourn. – N.Y. : Oxford University press, –1987. – 160 p.
2. Satyanarayana, A. Network reliability and the factoring theorem / A. Satyanarayana, M. K. Chang // Networks. – 1983. – Vol. 13. – P. 107–120.
3. Rodionov, A. S. On the Expected Value of a Number of Disconnected Pairs of Nodes in Unreliable Network / A. S. Rodionov, O. K. Rodionova, H. Choo // ICCSA 2007. LNCS. – Heidelberg : Springer, 2007. – Vol. 4707. – P. 534–543.
4. Chen, Y. A new algorithm for network probabilistic connectivity / Y. Chen, J. Li, J. Chen // Military Communications Conference Proceedings, 1999, MILCOM 1999. IEEE. – 1999. – Vol. 2. – P. 920–923.
5. Scaling the Distributed Stochastic Simulation to Exaflop Supercomputers / B. Glinsky, A. Rodionov, M. Marchenko, D. Podkorytov and D. Weins // High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICES), 2012 IEEE 14th International Conference on Proceedings. – P. 1131–1136.

Н. Б. Жукова

Россия, Пенза, Пензенский государственный университет

МЕТОД СОКРАЩЕНИЯ ДЛИТЕЛЬНОСТИ ПРОЦЕДУРЫ ПЕРВОНАЧАЛЬНОЙ СИНХРОНИЗАЦИИ В СИСТЕМАХ С ПРОГРАММНОЙ ПЕРЕСТРОЙКОЙ РАБОЧЕЙ ЧАСТОТЫ

Исследованы варианты реализации первоначальной синхронизации в системах с псевдослучайной перестройкой рабочей частоты, основанные на использовании рекуррентного поиска с применением авторегрессионного спектрального оценивания. Проанализированы известные алгоритмы, предложены варианты их модификации. Построены имитационные модели рассматриваемых решений, получены количественные данные, характеризующие их помехоустойчивость.

Поскольку перехват передаваемых объектом сообщений возможен лишь при установлении его активности в эфире, весьма привлекательным для систем военной связи и спецслужб является подход, при котором обеспечивается низкая вероятность обнаружения самого факта передачи. Возможность построения такой аппаратуры базируется на использовании систем с расширенным спектром, в которых энергия передаваемого сигнала оказывается распределенной в широком диапазоне частот, при этом мощность полезного сигнала не превышает мощности шума эфира. В подобных системах вероятность обнаружения факта передачи тем меньше, чем шире диапазон используемых частот и короче длительность сеанса связи [1]. В связи с этим предпочтительным подходом к их построению является расширение спектра с применением метода быстрой псевдослучайной перестройки рабочей частоты (ППРЧ), поскольку при его использовании обеспечивается возможность рас-

пределения энергии передаваемого сигнала в значительно более широком диапазоне частот по сравнению с другими известными методами расширения спектра [1].

При использовании в качестве датчика псевдослучайного кода, задающего закон изменения рабочей частоты, регистра сдвига с линейной обратной связью (РСЛОС) потенциально высоким быстродействием обладает рекуррентный поиск, основанный на авторегрессионном спектральном оценивании (АСО). (В зарубежной литературе данный метод поиска принято называть autoregressivespectralestimationacquisitiontechnique (ASEAT)).

Согласно методу авторегрессионного спектрального оценивания принимаемый сигнал (представляющий собой передаваемый сигнал с наложенным на него шумом) после дискретизации с частотой f_d задается последовательностью отсчетов $y(k)$ и рассматривается как авторегрессионный процесс, определяемым выражением:

$$y(k) = \sum_{m=1}^M \alpha_m y(k-m) + G n(k), \quad (1.1)$$

где $n(k)$ – последовательность отсчетов белого шума; $\alpha_m, m = \overline{1, M}$ и G – некоторые постоянные величины.

Полагая в (1.1) $G = 1$, авторегрессионный процесс можно трактовать как выходной сигнал фильтра с передаточной функцией

$$H(z) = \frac{1}{1 - \sum_{m=1}^M \alpha_m z^{-m}}, \quad (1.2)$$

на вход которого поступает белый шум $n(k)$ с единичной дисперсией $\sigma_n^2 = 1$.

При этом энергетический спектр (ЭС) такого процесса описывается выражением

$$Y(\omega) = 1 / \left| 1 - \sum_{m=1}^M \alpha_m e^{-j\omega m} \right|^2, \quad (1.3)$$

т.е. полностью определяется набором коэффициентов $\alpha_m, m = \overline{1, M}$. Таким образом, данный метод спектрального анализа сводится к определению коэффициентов $\alpha_m, m = \overline{1, M}$ авторегрессионной модели.

Для определения этих коэффициентов производится минимизация среднеквадратической ошибки линейного предсказания сигнала. Сигнал $y(k)$ пропускается через нерекурсивный фильтр с коэффициентами $[1, -\beta_1, -\beta_2, \dots, -\beta_M]$.

Линейным предсказанием текущего отсчета входного сигнала $y_{\text{лп}}(k)$ называют взвешенную сумму предыдущих отсчетов входного сигнала, вычисляемую в соответствии с разностным уравнением вида:

$$y_{\text{лп}}(k) = \sum_{m=1}^M \beta_m y(k-m). \quad (1.4)$$

Ошибкой линейного предсказания называют разность

$$e(k) = y(k) - y_{\text{лп}}(k). \quad (1.5)$$

С учетом выражений (2.2), (2.4) и (2.5) z -преобразование $E(z)$ ошибки предсказания $e(k)$ можно представить в виде:

$$E(z) = N(z) \frac{1 - \beta_1 z^{-1} - \beta_2 z^{-2} - \dots - \beta_M z^{-M}}{1 - \alpha_1 z^{-1} - \alpha_2 z^{-2} - \dots - \alpha_M z^{-M}},$$

где $N(z)$ – z -преобразование шума $n(k)$.

Отношение двух полиномов в последнем выражении в общем случае можно записать в виде бесконечного полинома $P(z)$ по отрицательным степеням переменной z , причем целая часть этого отношения в данном случае равна 1, т.е.

$$E(z) = N(z) (1 + p_1 z^{-1} + p_2 z^{-2} + \dots).$$

Сигнал ошибки $e(k)$ будет представлять собой результат преобразования белого шума $n(k)$ эквивалентным фильтром с передаточной функцией $P(z)$:

$$e(k) = n(k) + p_1 n(k-1) + p_2 n(k-2) + \dots.$$

Поскольку отсчеты белого шума являются независимыми, дисперсия сигнала $e(k)$ определяется как:

$$\overline{e^2} = \sigma_n^2 (1 + p_1^2 + p_2^2 + \dots).$$

Отсюда видно, что $\overline{e^2} \geq \sigma_n^2$ и равенство, соответствующее минимальному значению $\overline{e^2}$, достигается в том случае, когда $p_i = 0$ для всех i . Это возможно, если наборы коэффициентов $\{\alpha_i\}$ и $\{\beta_i\}$ совпадают.

Таким образом, задача определения ЭС вида (1.3) сводится к оценке оптимальных с точки зрения минимума СКО коэффициентов нерекурсивного линейного предсказателя, описываемого уравнением (1.4).

При известных значениях коэффициентов $\alpha_m, m = \overline{1, M}$ знаменатель выражения (1.3) легко вычислить с использованием алгоритма быстрого преобразования Фурье как дискретное преобразование Фурье последовательности $[1, -\alpha_1, \dots, -\alpha_M]$. Для получения любой требуемой разрешающей способности при оценке спектра достаточно дополнить указанную последовательность соответствующим количеством нулей. При условии, что принимаемый сигнал подается на вход предсказателя в течение интервала времени $kT_c \leq t < (k+1)T_c$, его частоту можно определить по полюсу выражения (1.3), максимизирующему значение $Y(\omega)$.

Список литературы

1. Spread Spectrum Communication Handbook [Electronic Edition] / M. K. Simon, J. K. Omura, R. A. Scholtz, B. K. Levitt. – N. Y. : McGraw-Hill, Inc., 2002. – 1229 с.
2. Журавлев, В. И. Поиск и синхронизация в широкополосных системах / В. И. Журавлев. – М. : Радио и связь. 1986. – 240 с.
3. Диксон, Р. К. Широкополосные системы / Р. К. Диксон : пер. с англ. под ред. В. И. Журавлева. – М. : Связь. 1979. – 304 с.
4. Rappaport, S. S. A two-level coarse code acquisition scheme for spread spectrum radio / S. S. Rappaport, D. L. Shilling // IEEE Transactions on Communications, September 1980. – 2008. – Vol. COM-28, – P. 1734–1748.
5. Putman, C. A. A Comparison of Schemes for Coarse Acquisition of Frequency-Hopped Spread-Spectrum Signals / C. A. Putman, S. S. Rappaport, D. L. Shilling // IEEE Transactions on Communications, February 1983. – 1983. – Vol. COM-31– P. 183–189.

Н. Э. Мельников

Россия, Пенза, Пензенский государственный университет

СИСТЕМА ПЕРЕДАЧИ ДАННЫХ НА ПРИМЕРЕ ВЫСОКОЧАСТОТНЫХ РАДИОМОДЕМОВ

Рассматривается организация низкоскоростной сети обмена данными. Организация низкоскоростной сети обмена осуществляется на основе модуля трансивера, входящего в состав базовой станции и абонентских станций комплекта.

Современные системы вооружения требуют создания средств высокоскоростного и защищенного обмена данными между несколькими (до нескольких десятков) абонентами в связной сети. В ряде случаев радиосвязь должна осуществляться на расстояниях, значительно превышающих дальность радиогоризонта, что требует трансляции сигналов между узлами сети. Средства обмена данными при этом должны надежно функционировать в условиях воздействия естественных и искусственных помех, быть устойчивыми к применению средств радиоэлектронной борьбы. Важным условием является взаимозаменяемость составных частей системы. Выход из строя одного или нескольких узлов не должен приводить к нарушению работы системы в целом.

Для реализации обмена данными используется совокупность станций, размещаемых на местности и образующих сеть обмена данными с динамически изменяемой топологией. Основные характеристики сети:

- Количество абонентов (направлений), задействованных в сети не менее 10;
- Количество каналов каждого абонента сети не менее 5;
- Скорость передачи данных при загрузке сети не менее 10 абонентов и при 5 каналах каждого абонента не менее 10 Мбит/с; суммарно со всех направлений к абонентскому устройству и не менее 20 Мбит/с суммарно со всех направлений к базовой станции;
- Дальность связи, при выполнении условий скорости передачи данных не менее 80 км при наличии прямой радиовидимости;
- Возможность расширения количества абонентов до 30 при выполнении условия скорости передачи данных и сохранении дальности связи не менее 80 км;
- Интерфейс, реализованный по протоколу типа Ethernet 10/100 Мбит/с;
- Диапазон рабочих частот – S.
- Организацию обмена засекреченной и открытой информации. Шифрование и расшифровывание осуществляется внешними по отношению к устройствами;
- Техническое маскирование передаваемых данных;
- Непрерывную работу в течение 72 ч.

Сеть обмена данными представляет собой совокупность образованную базовыми станциями, абонентскими станциями и каналами связи, образованными вышеперечисленными станциями.

Совокупность станций, размещенная на местности, образует сеть обмена данными с динамически изменяемой топологией. Сеть работает в синхронном режиме под управлением одной или нескольких станций. Станция синхронизирует обмен в группе ведомых станций, которые в текущей топологии подчинены ей. В качестве ведущей станции может выступать любая станция сети. Изменение топологии сети производится в ручном или полуавтоматическом режиме из любого узла сети, обладающего соответствующими правами. Сеть одновременно и независимо работает в двух диапазонах: 915-922 МГц и 2300-2550 МГц. Каждая станция включает, как

минимум, один приемопередатчик диапазона 900 МГц и один – диапазона 2400 МГц. В диапазоне 2400 МГц организуется высокоскоростная сеть обмена данными. В диапазоне 900 МГц – низкоскоростная сеть обмена данными.

Дублирование сетей позволяет повысить устойчивость системы к преднамеренным и непреднамеренным помехам и упростить управление сетью.

В данной статье представлена к рассмотрению организация низкоскоростной сети обмена данными. Для организации низкоскоростной сети обмена данными в составе базовой и абонентских станций используется модуль трансивера.

Модуль трансивера обеспечивает реализацию трех функций. Прием данных от абонентской аппаратуры и передачу их другим абонентам. Прием данных и передачу и на абонентскую аппаратуру. Ретрансляцию данных между абонентами.

Диапазон рабочих частот 915–922 МГц, разбит на 5 групп по 7 несущих частот в каждой. Каждая из пяти групп называется *каналом*. В пределах канала смена несущей частоты происходит по псевдослучайному закону. Смена несущей частоты производится каждые 50 мс. Период повторения последовательности несущих частот приблизительно равен 50 мин. В радиолинии применяется сигнал с частотной модуляцией. Сигнал является простым. Бодовая скорость может принимать следующие значения: 19200 бод/с, 38400 бод/с, 76800 бод/с, 154200 бод/с. Система синхронизации, входящая в состав сети, является замкнутой. В радиолинии применяется специальный синхросигнал.

Трансиверы низкоскоростного обмена данных сети разделяют на два типа: ведущий и ведомый. Все бортовые модемы являются ведомыми. Наземные модемы могут быть как ведущими, так и ведомыми (определяется пользователем). Синхронизацию в канале задает ведущий радиомодем. Таким образом, в одном частотном канале должен присутствовать только один ведущий модем. Если необходимо присутствие в канале более одной наземной станции, то только одна из них должна быть ведущей, а остальные необходимо перевести в режим ведомого модема. Ведомый радиомодем, приняв сигнал от ведущего модема, корректирует временное окно и заполнение генератора частот. Каждый радиомодем имеет свой уникальный идентификатор, и обращение к модемам производится посредством них. При совпадении идентификатора в пакете с личным адресом, ведомый производит ответ по ответному адресу. При этом ответным адресом вполне может быть адрес другой станции, и как раз за счет этого реализуется режим ретрансляции. При обмене применяются алгоритмы перезапроса и повторения в случае отсутствия ответа или наличия соответствующих флагов в заголовке пакета, что понижает вероятность потери полезной информации.

Вся передаваемая информация подвергается скремблированию, шифрованию, помехоустойчивому кодированию и перемежению. Применяется блочный код со скоростью $1/2$. Кодированию подвергаются полубайты, с получением в результате байта (8–4). Перемежение является побитным и осуществляется над 8-байтными блоками. Такое сочетание кода и перемежения позволяет исправить два ошибочных байта из каждых восьми. Заголовок пакета шифруется отдельным ключом. На каждой из 35 частот ключ к заголовку разный. Ключ к информации задается случайно и данные о его номере передаются в зашифрованном заголовке. В сети процедура смены ключей предусмотрена. Так как система использует принцип временного разделения, ввиду этого одна радиостанция находится в режиме излучения не более 50 % времени.

Описанные выше трансиверы низкоскоростного обмена данных применяются для организации радиосвязи между различными клиентами. Максимальное количество клиентов, имеющих возможность обмена информацией может достигать 21 (1 – ведущий + 20 – ведомые). Максимальное количество радиостанций используе-

мых для организации связи может быть выше 20 если используется ретрансляция при обмене между клиентами. Одна ведущая радиостанция способна вести обмен не более чем с 19 ведомыми, при этом с каждой из ведомых радиостанций схема организации радиосвязи может быть различной. Данная ситуация изображена на рис. 1.

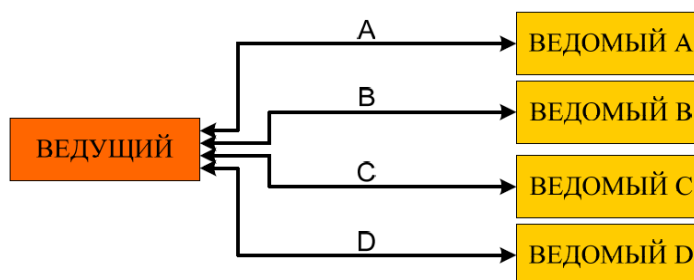


Рис. 1. Организация связи между ведущими и ведомыми

Каждая из веток А, В, С, D может иметь различную схему построения. Например ветка А может иметь один ретранслятор, ветка В два ретранслятора, а ветки С и D могут и вовсе не иметь ретрансляторов. Данный пример изображен на рис. 2.

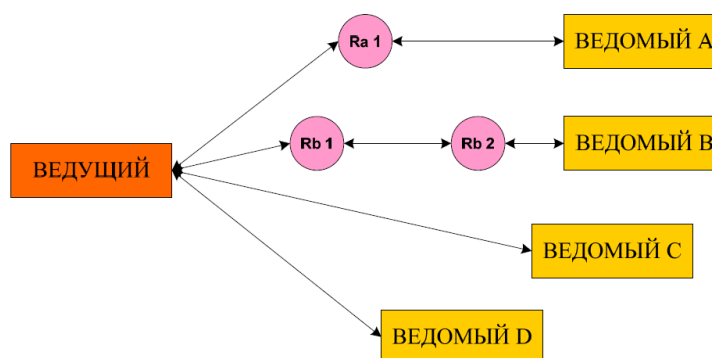


Рис. 2. Пример организации радиосвязи

Используемые твердотельная элементная база, неподвижные многолучевые антенны с электронным сканированием, современные цифровые методы формирования и обработки когерентных «шумоподобных» сигналов должны обеспечить получение повышенной скрытности излучения, малой массы, низкого энергопотребления и высокую надежность работы [1].

Список литературы

1. Конахович, Г. Ф. Сети передачи пакетных данных / Г. Ф. Конахович, В. М. Чуприн. – М. : Пресс, 2006. – 272 с.

Р. А. Ерёмина

Россия, Пенза, Пензенский Государственный университет

ОРГАНИЗАЦИЯ КОММУТАТОРА FAST ETHERNET С ПОДДЕРЖКОЙ QOS

Представлено описание структуры встраиваемого коммутатора Fast Ethernet с поддержкой качества обслуживания, ориентированного на встроенную реализацию

Технология коммутируемого Ethernet с поддержкой качества обслуживания (Quality of Service – QoS) в соответствии со стандартом IEEE 802.1q/p, нашли широкое применение в телекоммуникационных сетях, где необходимо обеспечить передачу трафика, чувствительного к задержкам, в том числе в системах в автоматического управления различного назначения [1,2]. Основным компонентом коммутируемого Ethernet является коммутатор, который обычно реализуется на базе быстродействующих процессов и заказных интегральных схем (ASIC) и конструктивно выполняется в виде отдельного устройства. Однако в последние годы появился новый класс коммутаторов – встраиваемые, реализация которых базируется на ПЛИС. Данные устройства в начале разрабатывались для интеграции на базовую плату бортовых систем, но в дальнейшем они станут неотъемлемой частью средств промышленной автоматизации, где требуется высокая скорость доставки пакета (приложения критичные к задержке передачи данных); работа в сложных климатических условиях; работы на подвижных объектах с условиях воздействия вибраций, однократных ударов и т.д. [3,4].

В работе поведена оценочная проработка проекта восьми портового встраиваемого коммутатора Fast Ethernet.

В результате анализа различных способов структурной организации коммутаторов [5, 6] можно сделать вывод о преимуществе архитектурного решения на основе разделяемой среды, реализуемой скоростной межпортовой магистралью используемой в режиме «мультиплексирования с разделением по времени» (Time Division Multiplexing – TDM), то каждому временному слоту соответствовал определенный выходной порт. Буферизация коммутируемых кадров выполняется в выходных портах.

Достоинства этого решения:

- Уменьшение количества буферов в входных портах и сосредоточении его в одном уменьшает аппаратные затраты и упрощает работу с памятью при реализации QoS.
- Простота организации коммутирующей среды, т.к. пространственная коммутация заменяется временной (режимом временного разделения одной магистрали). Требования к пропускной способности магистрали не могут быть велики в связи с небольшим количеством портов ввода-вывода.
- Уменьшение общей задержки коммутации, т.к. уменьшается буферизация на входном порту и уровней передачи при продвижении кадров по магистралям.
- Возможность организации конвейерной обработки кадров при их продвижении.

Структурная схема коммутатора на 8 портов, реализующего архитектуру с разделяемой средой передачи представлена на рис. 1.

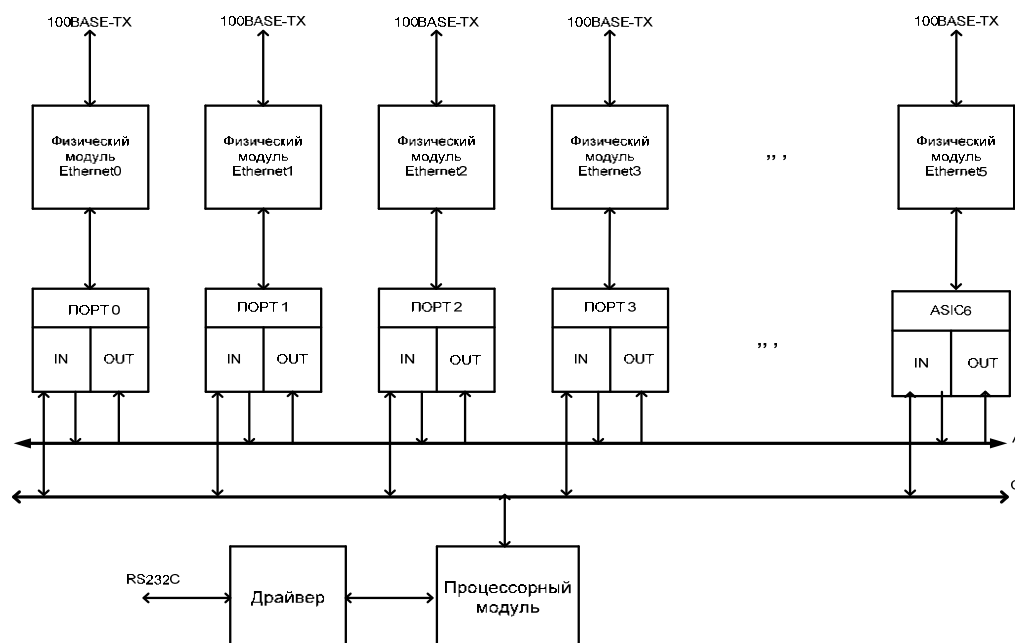


Рис. 1. Структурная схема коммутатора на 8 портов

Каждый из 8 портов ввода-вывода, работающих в дуплексном режиме, включает модули:

- физического уровня Ethernet, который обеспечивает интерфейс средой передачи по спецификации 100BASE-TX;
- порта, которых состоит из аппаратуры входного порта и аппаратуры выходного порта;
- процессорный, выполняющий функции системного управления

Входной порт принимает кадры с физического контроллера и выполняет их временную буферизацию на 64 байт, в случае длины < 64 байт уничтожает принятый кадр, формирует запрос к маршрутным таблицам, передает кадр в порт назначения через межпортовую магистраль А.

Выходной порт принимает кадры из межпортовой магистрали А, обеспечивает классификацию кадра по уровню QoS, буферизирует кадр в соответствующей очереди, обеспечивает обслуживание очередей по дисциплинам жесткого или циклического приоритета, обеспечивает передачу кадров в физический интерфейс.

Процессорный модуль составляет маршрутные таблицы, обращается к маршрутным таблицам за номером порта назначения по запросу от входных портов, управляет конфигурацией входных и выходных портов.

Процессорный модуль строится на базе быстродействующего RISC-процессора, например с архитектурой MicroBlaze [7]. Для настройки и контроля работы коммутатора используется стандартный последовательный порт RS-232C.

Связь между модулями обеспечивают две межпортовые магистрали – А и С.

Магистраль А обеспечивает продвижение кадров во входной порт по неблокируемой технологии TDM – это означает, что каждому абоненту (порту) выделяется строго определенный тайм-слот (TS), во время которого выполняется обмен ин-

Магистраль А ряд шин, предназначенных для передачи слов в выходной порт назначения, указатель на заполнение слова, номера активного тайм-слота, номера порта назначения (в том числе широковещательной рассылки), метка типа данных передаваемых в TS (пустой цикл, первое слово в кадре, слово в кадре, последнее слово в кадре).

Структура выходного порта показана на рис. 2. Он включает в себя:

- интерфейсный блок, через который порт подключается к магистралям А и С;
- блоки предварительной буферизации, которые обеспечивают временную буферизацию слов кадров, сохраняя их на 5 тактов для выявления значений полей приоритета класса трафика (Priority) и длины кадра (Length/Type) в соответствии с его форматом [2];
- буфера данных, каждый из которых предназначен для формирования и хранения очереди кадров, принадлежащих одному классу QoS (предполагается разбивать трафик на 4 класса обслуживания);
- блок обслуживания очередей, реализующий диспетчеризацию по алгоритмам строгого или циклического приоритета.
- контроллер внутренней шины В, поддерживающий обмен между шиной А и буферами данных.
- блок интерфейса физического уровня с внешним модулем физического уровня Ethernet.

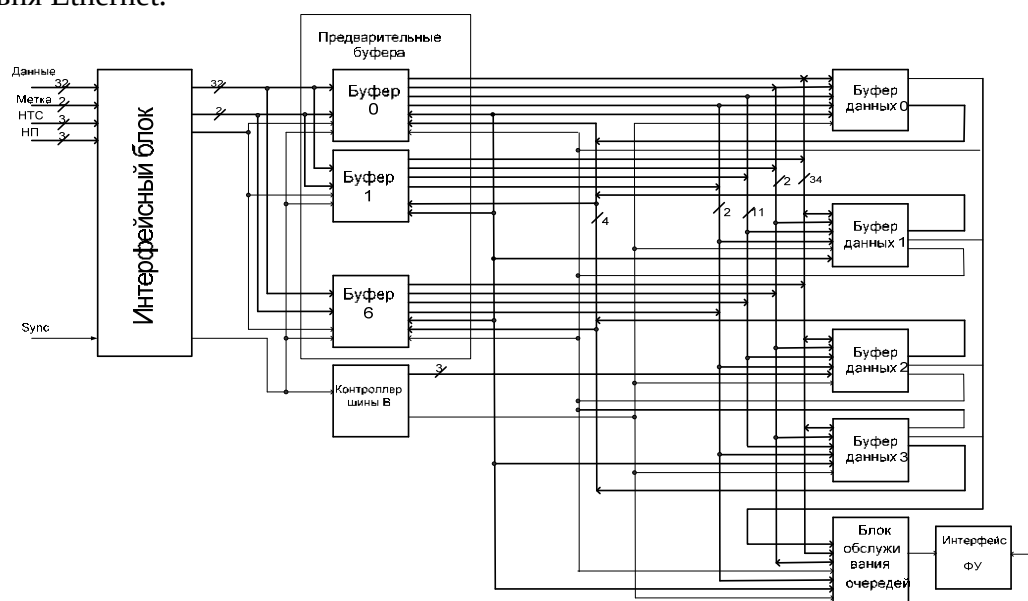


Рис. 2. Структурная схема модуля выходного пори

Внутренняя магистраль В работает синхронно и синфазно с внешней магистралью А. Из восьми тайм-слотов В магистральной В семь используются для записи кадров в буфера данных, а один, номер которого совпадает с номером выходного порта – для чтения данных из этих буферов блоком обслуживания очередей.

Память каждого буфера данных разбита на две части.

Память кадров, предназначенная для хранения непосредственно кадров, которая реализована в виде RAM емкостью 4096 34-разрядных слов. Каждое слово RAM состоит из 32-разрядного слова кадра и соответствующего ему двухразрядного кода счетчика байтов. В RAM информация помещается записями, содержащими 1 кадр. Размеры кадра от 64 до 1522 байт (от 16 до 380 слов).

Для контроля заполнения RAM введена память заголовков, описывающих каждый кадр, помещенный в RAM, которая реализована в виде регистрового ЗУ емкостью 16 заголовков. Таким образом, в очередь максимально может быть записано 16 кадров со средним размером кадра 1024 байта.

С целью оценки сложности реализации предлагаемых структурных решений было разработано описание на языке VHDL части выходного порта (интерфейсный блок, предварительный буфер и буфер данных) и выполнено моделирование средствами пакета ModelSim. Пример временной диаграммы, иллюстрирующей работу предварительного буфера при трансляции кадра приведены на рис. 3.

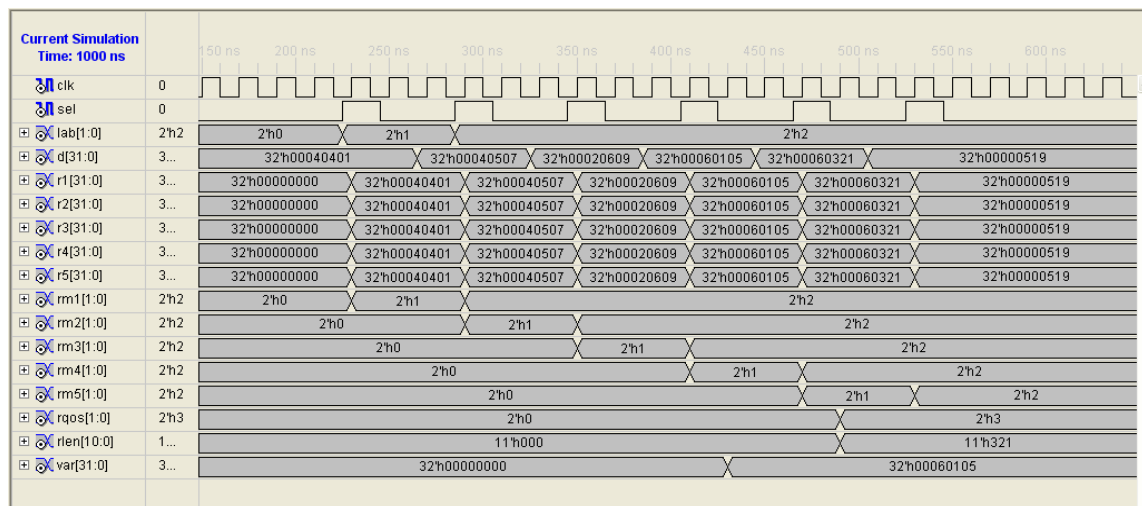


Рис. 3. Результат моделирования

Список литературы

- Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы : учеб. для вузов / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб. : Питер, 2010.
- EkNiclas. Department of Electrical Engineering, Helsinki University of Technology. IEEE 802.1 P,Q-QoS on the MAC level. – URL: <http://www.tml.tkk.fi/Opinnot/Tik-110.551/1999/papers/08IEEE802.1QosInMAC/qos.html>
- Зоммер, Рольф-Дитер. Встраиваемые коммутаторы Ethernet: время пришло / Рольф-Дитер Зоммер, Франк Вай // СТА. – 2012. – № 5. – С. 92–95.
- Марена, Т. Компактный, недорогой и экономичный процессорный Ethernet-мост с использованием ПЛИС / Т. Марена // Электронные компоненты. – 2008. – № 1. – С. 49–51
- Построение коммутируемых компьютерных сетей. – URL: <http://www.intuit.ru/studies/courses/3591/833/info>
- Коммутаторы локальных сетей D-Link : учеб. пособие. – М., 2006.
- Гурин, Е. И. Построение систем на кристалле с подчиненным процессорным ядром Microblaze на ПЛИС фирмы Xilinx / Е. И. Гурин // Компоненты и технологии. – 2007. – № 9. – С. 115–118.

МОДЕЛИРОВАНИЕ КОММУТАТОРОВ ETHERNET В ПРОМЫШЛЕННЫХ СЕТЯХ

Рассмотрены проблемы моделирования промышленных сетей Ethernet цветными временными сетями Петри.

В телекоммуникационной инфраструктуре современных АСУ ТП начала широко внедряться технология коммутируемого Ethernet [1–3], чему способствуют такие ее качества, как:

- высокая скорость передачи, обеспечивающая соответствие требованиям жесткого реального времени;
- простота интеграции с Internet/Intranet, в том числе по протоколам прикладного уровня;
- возможность по организации сетей самых разнообразных топологий;
- появление недорогих коммутаторов, решивших проблему недетерминированности Ethernet;
- эффективная техническая поддержка технологии со стороны международных организаций по стандартизации;
- наличие большого числа специалистов по обслуживанию Ethernet.

Обязательным требованием к оборудованию промышленного Ethernet есть поддержка службы обеспечения качества обслуживания (Quality of Service – QoS), позволяющей различным сетевым приложениям сосуществовать в одной сети, не уменьшая при этом пропускную способность других приложений. Реализуемые в коммутаторах Ethernet стандарта IEEE 802.1 q/r механизмы QoS обеспечивают выделение гарантированной полосы и приоритизацию трафика отдельного приложения [4].

Выбор оптимальных параметров настройки коммутаторов QoS представляет сложную задачу, которая может быть решена путем имитационного моделирования. Авторами предлагается использовать для моделирования с телекоммуникаций с поддержкой QoS аппарат цветных, иерархических, временных сетей Петри, реализованный в пакете CPN Tools [5].

Общие принципы проектирования традиционных корпоративных IP сетей представлены в работе [6]. Поэтому авторы основное внимание уделили моделированию механизмов QoS в телекоммуникационном коммутаторе (маршрутизаторе), используя при этом методологию нисходящего проектирования [7]. Модель на верхнем уровне иерархии включает (рис. 1):

- подсети «InPort» и «OutPort», которые моделируют порты коммутатора, обеспечивающие доступ к линиям связи в полнодуплексном режиме;
- подсеть «Switch», моделирующая продвижение кадра с входного порта на определенный выходной порт в соответствии с таблицей коммутации (поддерживаются статический и динамический режимы ведения таблицы);
- подсети «Buf» которые моделируют буфера кадров, направленных в соответствующий выходной порт.

В подсетях «Buf» собственно и реализуется механизм QoS: введение классов обслуживания для различных видов трафика, контроль перегрузок, поддержка различных алгоритмов управления очередями.

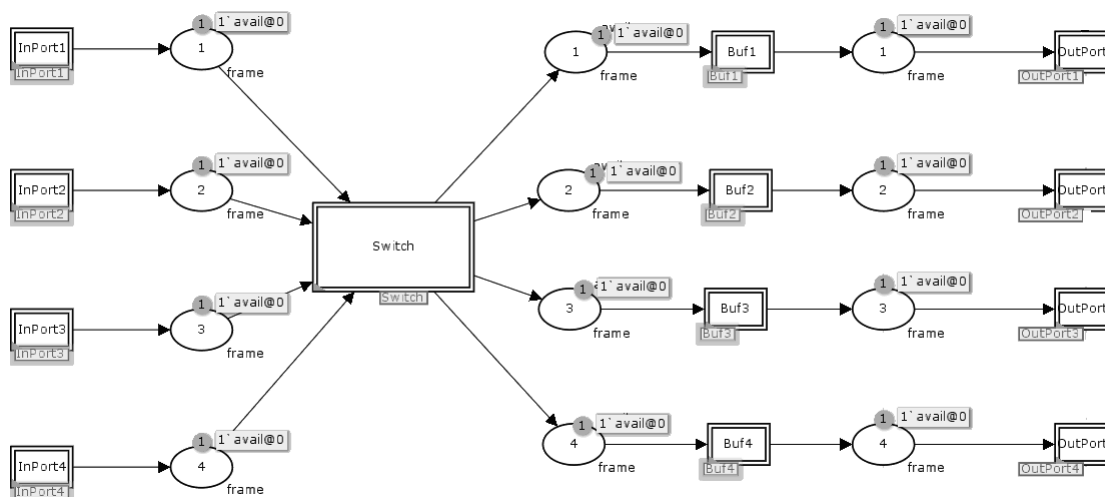


Рис. 1. Общая структура модель 4-х портового коммутатора

В отличие от традиционного для телекоммуникаций применения сетей Петри для моделирования протокола взаимодействия сетевых устройств, в нашем случае моделируется поток кадров, продвижение которых в сети определяется содержанием самих кадров, поэтому очень важно эффективно использовать аппарат цветности маркеров. Было введено два условных типа маркеров:

1. «Информационные» маркеры, продвижение которых имитирует обработку кадра в коммутаторе:

- множество цветов *frm* отражает структуру передаваемого кадра Ethernet и анализируется на очередном этапе обработки кадра в коммутаторе;
- множество цветов *frame* представляет однонаправленный канал, который может быть либо занят передачей кадра (*frm*), либо свободен (*avail*), что позволяет выявлять и обрабатывать события, связанные с наличием или отсутствием кадра.

Маркеры с цветом *avail* позволяют обеспечить безопасность сети Петри (один маркер в любой позиции) и решить проблему отсутствия в CPN Tools ингибиторных (запрещающих) дуг.

2. «Управляющие» маркеры, цвет которых отображает состояние обработки потока кадров в коммутаторе (размер очереди, количество переданных данных, признак очистки очереди и др.).

Среди механизмов обеспечения качества (маркировка и классификация трафика, обслуживание очередей, управление перегрузкой, управление полосой пропускания) важнейшим является механизм обслуживания очередей, т.е. правила выбора очереди для считывания из нее данных. Этот механизм непосредственно определяет задержку пакетов и вероятность их потери.

В коммутаторах уровней L2, применяются строго приоритетный (Strict Priority – SP) алгоритм, при котором очереди обслуживаются в соответствии с жестко установленным приоритетом, и циклические (RoundRobin-RR) алгоритмы, которые поддерживают последовательное обслуживание очередей, гарантируя доступ каждой к выходному порту. Обычно алгоритм RR реализуется в вариантах, позволяющих регулировать полосы пропускания для каждого канала:

- взвешенный (WeightedRoundRobin-WRR) – вес задает количество кадров, пересылаемых из очереди за один сеанс обслуживания, однако выделяемая полоса зависит от длины передаваемых пакетов;

- дефицитный (DeficitRoundRobin – DRR) – позволяет точно выделять полосу независимо от длины передаваемых пакетов.

Часто используются комбинированные варианты обслуживания очередей SP+WRR или SP+DRR, позволяющие обеспечивать критическому трафику минимальную задержку. Подсеть «Buf», реализующая подобные алгоритмы, показана на рис. 2 и включает следующие компоненты:

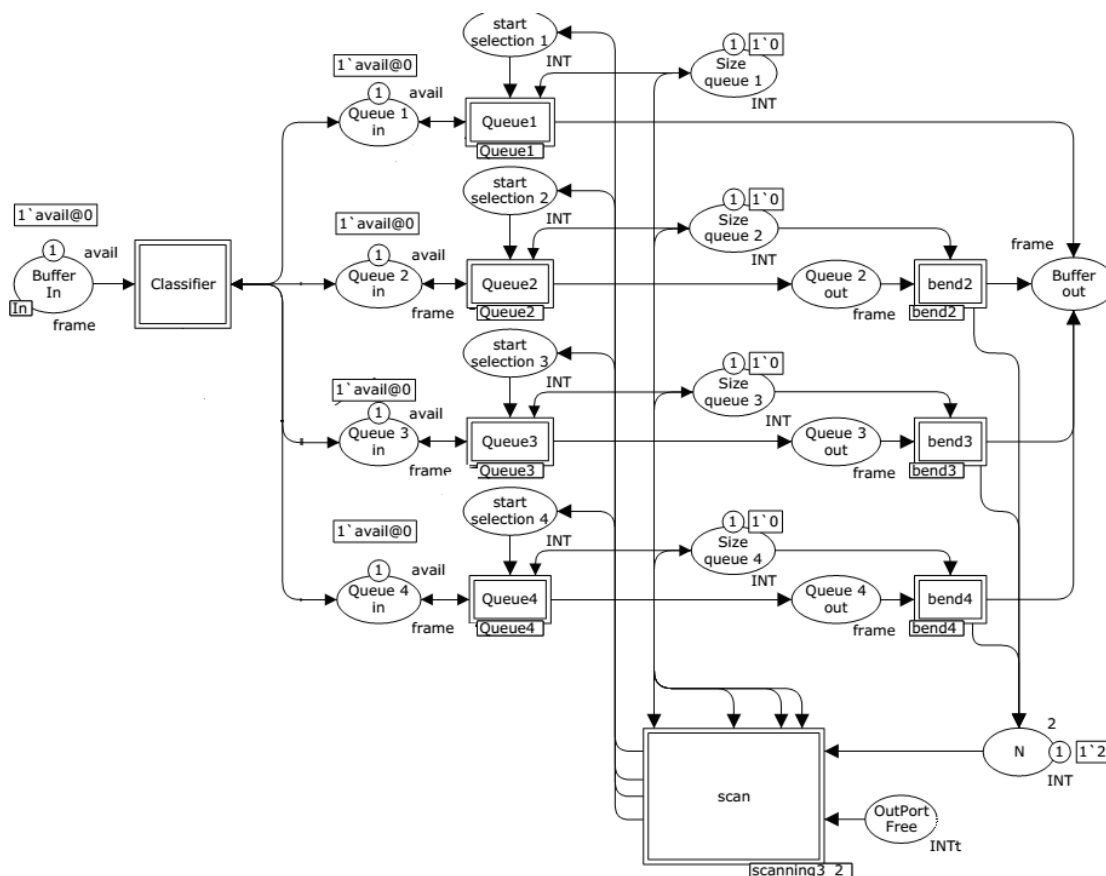


Рис. 2. Модель буфера выходного порта для циклических алгоритмов

- «Bufferin» и «Bufferout» – входной и выходной порты буфера выходного порта;
- «Classifier» - классификатор входящего трафика и направляющего его (трафик) в одну из четырех очередей на обработку.
- «Queue» очередь FIFO, содержащая пакеты ожидающие обработку;
- «Scan» подсеть, реализующая схему обзора (выбора) очередей;
- «Bend» подсети, реализующие учет весов для задания полосы каждой очереди;
- «N» позиция, содержащая номер очереди, которая должна обрабатываться следующей;

Модели алгоритмов WRR и DRR отличаются в реализации управления полосы: в первом случае – полоса выделяется в пакетах, а во втором – в битах. Подробная реализация алгоритмов WRR и DRR представлена в [8].

Для исследования модели построена схема генерации трафика сочетающего случайную и регулярную компоненты, принципы построения которой изложены в [9]. Предложенные модели позволяют исследовать вероятностно-временные ха-

рактеристики (средняя задержка, джиттер, доля потерянных пакетов)разнородного трафика, коммутируемого в сети промышленного Ethernet.

Работа выполнена в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития разработки научно-технологического комплекса России на 2014–2020» (соглашение № 14.574.21.0045 от 06.19.2014)

Список литературы

1. Кисилёв, В. Промышленный Ethernet в стиле Hirschmann / В. Кисилёв // СТА. – 2005. – № 2. – С. 6–12.
2. Кругляк, К. Локальные сети Ethernet в АСУ ТП: быстрее, дальше, надежнее / К. Кругляк // СТА. – 2003. – № 1. – С. 6–13.
3. Grübmerand, R. Ethernet-based fieldbuses for industrial networks: the basics / R. Grübmerand, S. Rupp // Industrial Ethernet Book. – URL: <http://www.iebmedia.com/index.php?id=8249&parented=63&themeid=255&hft=66&showdetail=true&bb=1>
4. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы : учеб. для вузов / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб. : Питер, 2010.
5. Jensen, K. Coloured Petri Nets: modelling and validation of concurrent systems / K. Jensen, L. Kristensen. – Springer Verlag, 2009.
6. Zaitsev, D. A. Switched LAN simulation by colored Petri nets / D. A. Zaitsev // Mathematics and Computers in Simulation. – 2004. – Vol. 65, № 3. – P. 245–249.
7. Механов, В. Б. Применение сетей Петри для моделирования механизмов обеспечения QoS в компьютерных сетях / В. Б. Механов // Новые информационные технологии и менеджмент качества (NIT&MQ'2010) ФГУ ГНИИ ИТТ «Информика» : материалы междунар. симп. – М. : ЭГРИ, 2010. – С. 209–214.
8. Механов, В. Б. Моделирование цветными сетями Петри обслуживания очередей алгоритмом WRR / В. Б. Механов, Е. А. Кизилов // Новые информационные технологии и системы : тр. IX Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2010. – Ч. 1. – С. 67–73.
9. Никишин, К. И. Моделирование трафика сети Ethernet цветными сетями Петри / К. И. Никишин, Н. Н. Коннов, А. Л. Домнин // Современные проблемы компьютерных наук (СПКН-2013) : сб. материалов I Междунар. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2013. – С. 120–123.

С. Н. Бессонов

Россия, Пенза, Пензенский государственный университет

НАЗНАЧЕНИЕ БЕСПРОВОДНЫХ МНОГОРАНГОВЫХ САМООРГАНИЗУЮЩИХСЯ СЕТЕЙ, ТЕХНОЛОГИИ И ПРИНЦИПЫ ПОСТРОЕНИЯ

Дается описание формальной интерпретации топологических изменений в самоорганизующихся сетях. Описан подход для разработки математической модели для процесса оценки качества работы систем с динамической топологией сети.

Область применения беспроводных широкополосных сетей значительно расширилась в настоящее время благодаря актуальности двух задач, известных всем как задача последней мили и задача построения децентрализованных сетей.

Задача последней мили заключается в организации доступа к сервисам традиционной проводной инфраструктурной сети конечных пользователей. Для решения такой задачи в проводную сеть необходимо включить специальное устройство – точку доступа (базовая станция или сервер), к которой по беспроводному каналу связи будут подключаться клиентские устройства пользователей. В рамках данной клиент-серверной архитектуры доступ к среде передачи данных может осуществляться централизованным или распределенным методом. В первом случае базовая станция на правах монополии управляет доступом к среде, предотвращая коллизии (одновременную передачу пакетов данных разными устройствами). Во втором случае доступ к сети осуществляется на основе конкурентной борьбы между всеми клиентскими станциями и самой точкой доступа за право передать свои пакеты данных.

Децентрализованные сети, или *ad-hoc* сети, – это самоорганизующиеся сети, создающиеся из равнозначных пользовательских устройств тогда, когда это необходимо, без использования проводной инфраструктуры. Задачу построения таких сетей также можно решить с помощью выделения в сети некоторого координирующего устройства и наделения его полномочиями сервера по отношению к другим устройствам данной сети, играющими роль клиентских устройств, но это просто нецелесообразно. В отличие от задачи последней мили, принудительное назначение устройствам ролей клиентов и сервера при решении задачи построения сети с децентрализованной организацией снижает надежность этой сети. Если устройство – координатор выйдет из строя, то будет нарушена работа всей сети, даже если это устройство не выполняет никаких функций, исполняемых клиентскими устройствами сети. Вот почему при решении задачи построения децентрализованных сетей предпочтительно использование исключительно распределенного управления доступом к каналу передачи данных.

Отказ от клиент-серверной архитектуры при построении *ad-hoc* сетей делает решение задачи последней мили и решение задачи построения сетей с децентрализованной организацией совершенно различными, что наиболее ярко отражается в разработанном международным комитетом IEEE 802 стандарте IEEE 802.11 [1] беспроводных сетей известных под торговой маркой Wi-Fi. В стандарте описываются два типа сетей: инфраструктурные сети и сети *ad-hoc*.

Технология инфраструктурных сетей (Wi-FiHotspot) очень известна благодаря своему широкому распространению и простоте развертывания сетей, она уже хорошо изучена, и в данной работе рассматриваться не будет.

Сети *ad-hoc*, требующие инфраструктуры, в рамках базового стандарта IEEE 802.11 являются одноранговыми сетями, где каждое устройство находится в зоне непосредственного радиоприема всех остальных устройств. Однако с момента начала работы над стандартом появилось множество новых задач, связанных с развертыванием таких беспроводных сетей, как:

- 1) сенсорные сети различного назначения, передача данных с различных датчиков, передача данных на большие расстояния, за счет ретрансляции;
- 2) сети связи военного назначения тактического уровня;
- 3) аварийные сети, развертываемые в условиях чрезвычайных ситуаций;
- 4) домашние сети и сети предприятия, развлекательные сети;
- 5) сети контроля и диспетчеризации транспортных средств;
- 6) сети, в удаленных районах, где отсутствует стационарная инфраструктура;

Такие разнообразные задачи требуют расширения зоны покрытия сети и обеспечения бесперебойной работы движущихся в пространстве устройств.

Расширение зоны покрытия сети будет означать, что, если беспроводная сеть и остается связной, некоторые устройства могут в течение своей работы могут

находиться то в зоне непосредственного радиоприема друг друга, поэтому для доставки пакетов данных между ними требуется ретрансляция этих пакетов через промежуточные станции. Таким образом, расширение зоны покрытия рано или поздно должно привести к переходу от одноранговой сети к многогранговой.

Движение же станций в пространстве означает, что топология сети может изменяться с течением времени и устройства могут находиться то в зоне непосредственного радиоприема друг друга, то за пределами этой зоны. Технологиями, призванными расширить зону покрытия сети и обеспечить безотказную работу движущихся в пространстве устройств, стали технология самоорганизующихся мобильных adhoc сетей MANET[2], оформленная в виде спецификаций организаций IETF (англ. InternetEngineeringTaskForce), и технология сетей Wi-FiMesh, разработанная в комитете IEEE 802 LAN/MAN StandardsCommittee [3].

Исследования, проводимые зарубежными и отечественными учеными, показывают, что на современном уровне развития в MANET сетях практически невозможна передача трафика в реальном времени при значениях скорости передачи не более 1 Мбит/с, что неприемлемо при организации передачи информации между различными транспортными средствами. Основной причиной, по которой возникает невозможность передачи трафика реального времени с большой скоростью, являются особенности работы протоколов канального сетевого уровней, вызванные сильным влиянием формального представления топологических изменений в MANET сети.

Поэтому, целью дальнейшего исследования будет являться, корректировка формальной интерпретации топологических изменений, разработка математической модели для процесса оценки качества работы систем с динамической топологией сети и создание алгоритма функционирования программного комплекса по анализу правильности выбора параметров для устройств в сети с динамической топологией.

Сети MANET

Сети MANET (англ. MobileAdhocNETwork – мобильная беспроводная децентрализованная сеть) представляют собой радиосети, координация взаимодействия случайных мобильных устройств в которых осуществляется на сетевом уровне, а доступ к каналу связи осуществляется с помощью одной из уже существующих технологий, допускающих построение децентрализованных беспроводных сетей. Наибольшее развитие и известность получили сети MANET, построенные на базе технологии Wi-Fiad hoc, и использующие для многошаговой доставки пакетов данных протоколы маршрутизации, работающие на сетевом (протокол IP) уровне.

Особенности распределенного метода доступа к среде передачи данных в сетях Wi-Fiad hoc

Стандарт IEEE 802.11 определяет набор правил передачи пакетов, позволяющих обеспечить распределенный конкурентный доступ к среде передачи данных. В сетях ad hoc базовым методом для доступа к среде передачи данных является режим распределенного управления DCF (англ.: DistributedCoordinationFunction), в основе которого лежит механизм множественного доступа с контролем несущей информации и избеганием коллизий CSMA/CA (англ.: CarrierSenseMultipleAccesswithCollisionAvoidance).

Механизм DCF работает следующим образом [4]:

В непустую очередь устройства сети поступает пакет данных. Этот пакет данных будет ожидать в очереди, пока сетью не будет обслужен предыдущий пакет данных. После этого станция инициализирует счетчик отсрочки с помощью случайного целого числа из конкурентного окна, которым называется интервал $[0; SW_{min}]$. Пока среда передачи данных свободна, значение счетчика отсрочки бу-

дет уменьшаться на 1 каждый интервал времени длительностью slot. Когда среда передачи данных становится занятой, счетчик отсрочки замораживается, и возобновление отсчета слотов происходит после того, как среда передачи данных освободится на определенный интервал времени. В качестве такого интервала времени выступает EIFS (англ. Extended interframe space – увеличенный межкадровый интервал), если передается сигнал, который устройство не смогло декодировать, и DIFS (англ. DCF interframe space – межкадровый интервал DCF) – если сигнал был декодирован устройством. Устройство может начать передачу пакета данных при соблюдении следующих условий:

- 1) Если среда оставалась свободной в течение интервала DIFS или EIFS (выбор интервала описан выше);
- 2) Если значение счетчика отсрочки равно 0.

Пакет может предназначаться одному или нескольким устройствам. Если пакет предназначен одному устройству (одноадресная передача), то оно должна сразу же подтвердить его получение. Для этого, по истечении короткого интервала времени SIFS (англ. Short Interframe Space – короткий межкадровый интервал) после приема пакета устройство-приемник отправляет короткий сигнал – кадр ACK (англ.: acknowledgement – подтверждение). Если устройство отправитель пакета получает сигнал ACK в течение интервала AckTimeOut (интервал ожидания сигнала Ack) после отправки пакета, то на этом передача (обслуживание) пакета прекращается, пакет считается удачно переданным. Если попытка передачи была неудачной и лимит RL (англ. Retry Limit – ограничение числа повторов) количества повторных попыток передачи одного пакета не достигнут, станция начинает новую попытку передачи пакета данных. Новая попытка начинается с выбора нового интервала задержки из увеличенного интервала $[0; SW_{new}]$:

новое значение интервала задержки SW_{new} зависит от предыдущего (старого) SW_{old} согласно следующей формуле [4]:

$$SW_{new} = \min\{2(SW_{old} + 1) - 1, SW_{max}\}; \quad (1)$$

где SW_{max} – максимальный размер межкадрового интервала.

Если попытка передачи пакета данных была неудачной и достигнут лимит RL, то устройство прекращает обслуживание пакета. Завершение обслуживания пакета всегда сопровождается инициализацией счетчика задержки с помощью случайного числа из минимального конкурентного окна $[0; SW_{min}]$. Если пакет поступил в очередь устройства, когда оно не занято обслуживанием какого-либо пакета, то устройство может начать его передачу сразу же, как только счетчик задержки, введенный после предыдущей передачи пакета данных, обнулится и среда станет свободной в течение интервала DIFS или EIFS. Очевидно, что методы DCF оказываются эффективны, только когда все устройства находятся в зоне уверенного радиоприема друг друга. В этом случае устройства, имеющие пакеты данных для передачи, будут отсчитывать счетчики задержки синхронно. Таким образом, коллизия может возникнуть только тогда, когда счетчики нескольких устройств будут одновременно обнуляться. Вероятность такой ситуации тем меньше, чем больше будет размер конкурентного окна, увеличивающегося сразу же, если произойдет коллизия.

В многоканальной беспроводной сети возможны такие явления, как например, «эффект скрытых устройств». Скрытыми обычно называют устройства, находящиеся вне зоны уверенного радиоприема друг друга, но имеют при этом общее устройство. На рис. 1 это устройства А и С.

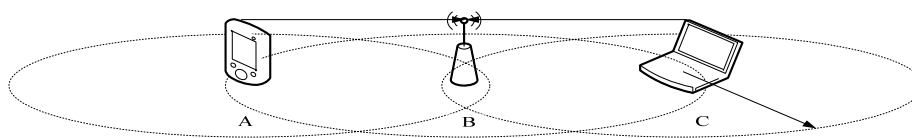


Рис. 1. Пример эффекта скрытых устройств

Поскольку скрытые устройства не «слышат» радиосигналы друг друга, они считают слоты отсрочки независимо друг от друга, и одно из них может начать передачу в то время, когда другое уже передает свой пакет данных. Произойдет коллизия, и устройство В не сможет корректно принять ни пакет от устройства А, ни пакет устройства С. Вероятность такого события существенно выше вероятности одновременного начала передачи пакетов данных. Как указано в работах [6] наличие скрытых устройств может приводить к существенному снижению скорости передачи в беспроводной сети, а в некоторых случаях – к блокировке соединений между устройствами, что может привести к нарушению связности сети.

Итак, распределенный метод доступа, используемый в сетях Wi-Fi, оказывается неэффективным в многошаговых сетях и может привести к низкой вероятности успешной доставки пакета, что необходимо принять во внимание при дальнейшей работе с этими сетями.

Каждый из используемых протоколов маршрутизации (AODV и OLSR) потенциально имеет свои преимущества и недостатки при использовании в условиях мобильных Adhoc-сетей [6, 7]. Например, проактивные (OLSR) протоколы обладают явным преимуществом перед реактивными (AODV) во времени построения маршрута. У проактивных протоколов этот процесс, происходит заранее, и требуется лишь считать маршрут из таблицы, тогда как реактивным протоколам необходимо разослать широковещательный запрос и дождаться подтверждения от адресата. Однако проактивным протоколам необходимо постоянно осуществлять широковещательные рассылки, на что расходуется значительная доля пропускной способности сети, особенно в условиях крупных сетей с высокой мобильностью узлов.

Список литературы

1. IEEE Standard for Information technology – Telecommunications and information exchange between systems – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2007.
2. Mobile Ad-hoc Networks (manet): Rep.: IETF, 2008. – URL: <http://www.ietf.org/html.charters/manet-charter.html>.
3. IEEE 802.11s STANDARD for Information Technology – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment: Mesh Networking [Electronic resource], 2011.
4. Хоров, Е. М. Метрика маршрутизации для трафика, чувствительного к задержкам / Е. М. Хоров. – 2010. – С. 11–19.
5. Bianchi, G. Performance Analysis of the IEEE 802.11 Distributed Coordination Function / G. Bianchi // IEEE Journal on Selected Areas in Communications – 2000. – № 18(3). – P. 535–547.
6. Lyakhov, A. IEEE 802.11 Direct links: Selected Lectures on Multiple Access and Queueing Systems / A. Lyakhov, I. Pustogarov, A. Gudilov // Revised Selected Papers from International Workshop on Multiple Access Communications (MACOM-2008), Saint-Petersburg, Russia, 16–17th June, 2008. – Saint-Petersburg, 2008. – P. 15–24.
7. Ляхов, А. И. Проблема неравномерного распределения пропускной способности в сетях IEEE 802.11 / А. И. Ляхов, И. А. Пустогаров, А. С. Гудиллов // Информационные процессы. – 2008. – Т. 8, № 3. – С. 149–167.

ON MODIFICATION ACK-BASED SCHEME USING FSA FOR AD-HOC NETWORKS

The collection of wireless nodes is nothing but the mobile ad hoc network which is also called as MANET. In MANET, nodes are moving arbitrarily means totally in dynamic way. Positions of the nodes are not fixed as compared to the wired nodes which are located to particular position. Thus the connection in between the all mobile nodes depends on their current positions and hence that connection is continuously and frequently changing. In such network to make the connections between such mobile nodes some mechanism is needed in order to make the communication and route discovery in between them, such mechanism is called as MANET routing protocols. For the overall MANETs performance all routing protocols plays crucial role for the same. There are number of routing protocols which are designed and proposed by researchers and network designers for the MANET with prime aim of improving the routing mechanism and efficient network communication. Ad-hoc on demand routing protocol (AODV), optimized linked state routing protocol (OLSR), dynamic source routing protocol (DSR) etc. are the well-known routing protocols for the same. In order to understand the working of such routing protocols, we have to do the exact communication scenarios and network configuration for the MANET.

INTRODUCTION

In this project I'm focusing on the use of the ad-hoc on-demand distance vector routing protocol (AODV), this protocol I've developed through reviewing some studies and research papers which they are related to this topic of the MANET routing protocols and I've developed the (AODV) protocol based on some facts which I've collected from previous studies and research papers which I've mentioned earlier above .So, my development was exactly is how to ensure a higher level of security through several methods and then by the using of the fuzzy logic concept so finally, I've developed the protocol to be (FL-SAODV). So as a basic understanding of my project here, it is all about acquiring a deep understanding of the (AODV) routing protocol in MANET and the concept of the optimized network evaluation (OPNET) modeler 14.0 and then I decided to use the NS-2 modeler to carry on with the work of mine and it will be clear why I've done this through the whole work of mine and the results. And through my proposed work I've developed the way of an efficient routing for the MANET by detecting the malicious nodes and how to avoid the route through them to send a packet to its destination and a proposed algorithm to deal with such nodes and enhancing the routing concept.

AD HOC ON-DEMAND DISTANCE VECTOR (AODV) PROTOCOL

Ad hoc On-Demand Distance Vector, AODV, is a distance vector routing protocol that is reactive. The reactive property of the routing protocol implies that it only requests a route when it needs one and does not require that the mobile nodes maintain routes to destinations that are not communicating. AODV guarantees loop-free routes by using sequence numbers that indicate how new, or fresh, a route is. The AODV protocol is one of the on-demand routing protocols for ad-hoc networks which are currently developed by

the IETF Mobile Ad-hoc Networks (MANET) working group. it follows the distance vector approach instead of source routing. In AODV, every node keeps a local routing table that contains the information to which of its neighbors it has to forward a data packet so that it reaches eventually the desired destination. In general, it is desirable to use routes which have minimal length according to hop-count as a distance metric. However, AODV provides the functionality like DSR, namely to transport data packets from one node to another by finding routes and taking advantage of multiple hop communication. AODV is based on UDP as an unordered transport protocol to deliver packets within the ad-hoc network. Moreover, it requires that every node can be addressed by a network wide unique IP address and sends packets correctly by placing its IP address into the sender field of the IP packets. This means also that AODV is expected to run in a friendly network, where security is a minor concern. It should be mentioned that there are some attempts to extend AODV to prevent malicious nodes from attacking the integrity of the network by using digital signatures to secure routing control packets. AODV requires each node to maintain a routing table containing one route entry for each destination that the node is communicating with. Each route entry keeps track of certain fields. Some of these fields are:

- a. **Destination IP Address:** The IP address of the destination for which a route is supplied.
- b. **Destination Sequence Number:** The destination sequence number associated to the route.
- c. **Next Hop:** Either the destination itself or an intermediate node designated to forward packets to the destination.
- d. **Hop Count:** The number of hops from the Originator IP Address to the Destination IP Address.
- e. **Lifetime:** The time in milliseconds for which nodes receiving the RREP consider the route to be valid.

NETWORK SIMULATORS

The NS-2 simulator is a discrete event simulator widely used in the networking research community. It was developed at the University of California at Berkeley and extended at Carnegie Mellon University to simulate wireless networks. These extensions provide a detailed model of the physical and link layer behavior of a wire/wireless network and allow arbitrary movement of nodes within the network.

NS-2 Simulator is usually a software package that simulates a real system scenario. Through the simulation we can test how a device or a system will perform in terms of timing and result. In addition to that it can be used to explore new policies, operating procedures without interrupting the system in real time.

Network simulation is very important because the network designer can test a complex network and make the right decisions about the designing in order the network will not face any problems in the future. New network devices can be added and testing without disturbing the existing network. NS-2 has high performance and it is very easy to use because of the combination of two languages. NS architecture follows the OSI model.

Node in a network is a point that connects other points, either a distribution point or an end point for data transmissions. A node can send or receive data. All kind of nodes in NS-2 are separated in two types of nodes. A unicast node that sends packets to only one node and a multicast node that sends packets to more than one node. Attack traffic source node is a node that sends malicious data (spoofed data) to other nodes. Traffic agents such as TCP or UDP are assigned on those nodes.

The receiving node is called sink node. A sink node can be an end node of the network. It can receive from different type of traffic source nodes. In case of a sink receives

data from a TCP traffic node is defined as Agent/TCP sink and as Agent/Null if it is received from a UDP traffic node.

NS-3 is a new simulator (not backwards-compatible with NS-2). It is a free, open-source software project organized around research community development and maintenance; the target user community is networking researchers and educators. NS-3 is not an extension of NS-2. It is a new simulator, written from scratch [1].

PROPOSED ACK-BASED SCHEME

Proposed Modified Ack-Based scheme for node authentication with AODV existing protocol in MANET. Ack-Based scheme also provides facility for the detection of worm-hole attack and node misbehaviors in ad hoc network. Different researchers proposed different schemes for Ack-Based for providing security in mobile ad hoc network. But all techniques suffered common problems that are generation of huge amount of packet overhead and node ambiguity. Due to this problem the given scheme is not used in general form. So we proposed the Ack-Based scheme with the help of finite state machine for controlling a generation of packet and also improve performance of AODV protocol. Here first we discuss basic Ack-Based scheme and then we discuss finite state machine and finally we discuss combined approach for Ack-Based scheme. ACK based scheme overcomes the limitation of passive-feedback technique when power control transmission is used. To implement this scheme, an authentication mechanism is used to prevent the next hop from sending a forged Ack packet on behalf of the intended two hop neighbor. The main drawback of this scheme is the huge overhead. In order to reduce the overhead, the authors have proposed in that each node asks its two hop neighbor to send back an ACK randomly rather than continuously. Likewise, this extension also fails when the two hop neighbor refuses to send back an Ack. In such situation, the requester node is unable to distinguish who is the malicious node, its next hop or the requested node. To overcome the previous ambiguity in determining the true malicious node, focuses on detecting malicious links instead of malicious nodes. The authors propose the 2Ack scheme to detect malicious links and to mitigate their effects. This scheme is based on 2Ack packet that is assigned a fixed route of two hops in the opposite direction of the received data traffic's route. In this scheme, each packet's sender maintains the following parameters; (i) list of identifiers of data packets that have been sent out but have not been acknowledged yet, (ii) a counter of the forwarded data packets, (iii) and a counter of the missed packets. According to the value of the acknowledgement ratio (Rack), only a fraction of data packets will be acknowledged in order to reduce the incurred overhead. This technique overcomes some weaknesses of the Watchdog/path rater such as: ambiguous collisions, receiver collision and power control transmission. Both of the previous works remain vulnerable to the attacks launched by group of nodes. To counter these attacks, provides a framework to mitigate the damage caused by the colluding black hole attack in AODV. The proposed technique has a moderate overhead induced by the ACK sent back by the destination during selected intervals of data transfer period. Throughout the data packets transmission, a flow of special packets is transmitted at random intervals along with the data. The reception of these special packets invokes the destination to send out an ACK through multiple paths. The ACK packets take multiple routes to reduce the probability that all ACKs being dropped by the malicious nodes, and also to account for possible loss due to broken routes or congestion in certain nodes. If the source node does not receive any ACK packet, then it becomes aware of the presence of attackers in the forwarding path. As a reaction, it broadcasts a list of suspected malicious nodes to isolate them from the network.

All the nodes running a solution based on acknowledgment need to maintain a timeout (To) value. This timeout represents an upper bound of the time that the sender node has to wait for the ACK to arrive. The determination of this timeout value is critical

since a small value induces a large number of false accusations and a large value increases the memory required to store the outgoing packets for further comparisons. Figure depicts an example of the lower bound of the timeout value maintained by node A for the reception of two hops ACK from node C. The timeout value should be greater than the estimated threshold (T_h) value which can be calculated as follows $T_h = T_1 - T_2$, where T_1 and T_2 are the sending (reception) time of the packet (Ack), respectively this threshold is estimated for a successful transmission at MAC layer without any retransmission, which is not a realistic assumption in MANETs, thus the timeout value should satisfy the following condition $T_o > T_h + (avg_{rt} \times \text{hop delay})$ where AVG_{RT} is the average number of retransmissions of a packet at MAC layer, and hop delay is the one hop transmission delay which includes packet transmission delay, random Backoff delay at the MAC layer and the processing delay. So the proposed work will be the following steps:

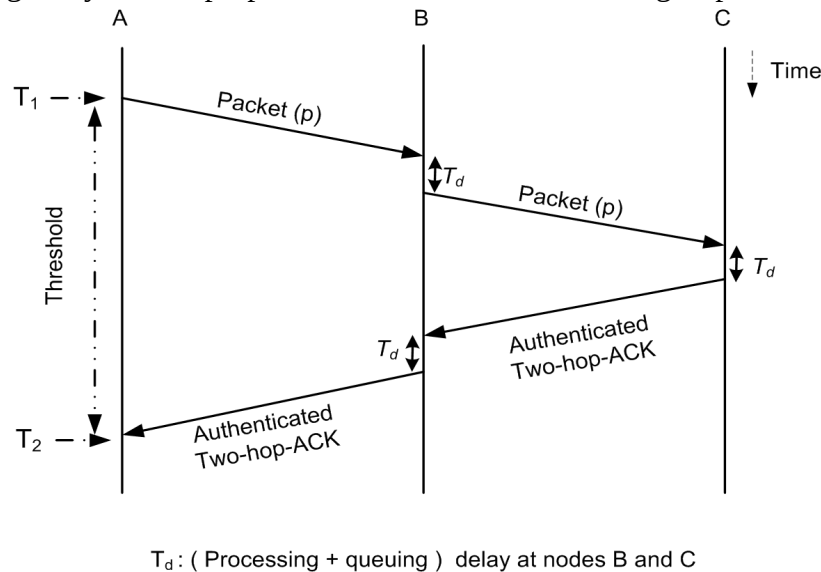


Figure. State diagram of ack scheme

1. Sender node forwards application packet to one of its immediate neighbor delegating the responsibility of further forwarding it to that neighbor. Sender also sets a timer (which is twice the network diameter) to receive acknowledgement from destination node.

2. If acknowledgement is received before timer expires then the route is considered to be trusted and no further action is needed.

3. If timer expires and acknowledgement is not received then the route is not considered to be trusted. Now sender sends an application packet to next to next hop address node and again sets a timer to receive an acknowledgement.

- If acknowledgement is received before timer expires then the neighbor node is considered to be trusted and no further action is needed, otherwise the neighbor node is considered as half-trusted. In this situation, this half-trusted node will be under observation until it shows same malicious behavior again when an application packet is forwarded to it next time.

4. If the half-trusted node does not show malicious behavior when the application packet is forwarded to it next time then it will be considered as trusted-node, otherwise this half-trusted node will be considered as malicious and following actions will be performed:

- No further RREP messages from this node will be entertained.
- The application packets will not be forwarded to this node.

- New routes will be discovered to forward application packets to those destinations that have this un-trusted (malicious) node as the next hop address in the routing table.

MODIFIED ACK-BASED SCHEME

Node	path	RREQ PACK	RREPPACK	FSA
A	source	broadcast	none	q0
B	A-B	B receive	A wait	q0
C	A-C	C receive	B reply	q1
D	B-D	D receive	B wait	q0
E	destination	E receive	D reply	q1

State maintains FSA algorithm with mobile node

Proposed Steps of FSA Algorithm

The Algorithm broadcasts with FSM (RREP, RRQP, SO, S1)

Here RREP → Reply packet

RRQP → Request packet

S0 → initial state

S1 → final state

- **if** source node RREP $\in$ S0 **then**
- **if** Id = 1 **then**
- **if** int_node $\in$ S0 set **then**
- prepare HELLO rep pkt
- send HELLO rep pkt
- **end if**
- process HELLO msg
- **else**
- process HELLO msg
- **end if**
- **end if**

Algorithm Receive with FSM (RREP, RRQP, SO, S1)

Here RREP → Reply packet

RRQP → Request packet

S0 → initial state

S1 → final state

- **if** (dist node) $\in$ S1 set **then**
- **if** int_node $\in$ RREP **then**
- prepare hop ackpkt
- send hop ackpkt
- **end if**
- UPDATE(S1)
- **else**
- **if** sender addr $\in$ S0 **then**
- add its own Request S0 STATE
- forward RREP Packet
- **end if**
- **end if**

IMPLEMENTATION

The main code is implemented in aodv.cc and the functions are declared in aodv.h. In aodv_packet.h, the AODV message formats (RREQ, RREP, RERR and HELLO) are defined. Moreover the new message format FSAPAC (Finite state automata) has been added and defined in this file. The main modifications have been done in aodv.cc. a) When a mobile node is to send a data packet to a destination, it tries to find a route to the destination (FSARt_resolve). b) If the mobile node does not have any valid route to the destination it broadcasts a RREQ message (send Request). c) The RREQ message is eventually received by the destination or another node which knows a route to the destination (recvRequest). d) The node sends a RREP/RREP_I message back to the originator of the RREQ (send Reply). e) The originator of the RREQ receives the RREP/RREP_I message (recvReply) and starts sending data packets to the destination (find_send_entry if the destination is a fixed node). Simulation is run for 100 seconds of simulated time. 10 of the 25 mobile nodes are constant bit rate traffic sources.

CONCLUSION AND FUTUREWORK

Without infrastructure and node mobility in adhoc network is a great challenge as concern to the security. For security concern various method are used for node authentication in mobile adhoc network. The authentication scheme of leader agent and member surveillance greatly reduces the relative calculating overheads and communication costs. Generally speaking, when leader agent node and surveillance nodes are not destroyed, the united nodes can ensure the reliability, the authentication result is reliable. Suggestion of future work is to apply modified Ack-Based FSA add with other protocol to requirement of memory. During path discovery and path establishment it take much time in compression of normal Ack-Based schemes in future minimized route calculation with finite state machine.

References

1. A study of comparison of Network Simulator – 3 and Network Simulator – 2 / R. Chaudhary et al. // (IJCSIT) International Journal of Computer Science and Information Technologies. – 2012. –Vol. 3 (1). – P. 3085–3092.
2. Djahel, S. Mitigating Packet Dropping Problem in Mobile Ad Hoc Networks: Proposals and Challenges / S. Djahel, F. Naït-abdessalam, and Z. Zhang // IEEE Communication and survey. – 2010.
3. Mitigating routing misbehavior in mobile ad hoc networks / S. Marti, T. J. Gyu-li, K. Lai and M. Baker // Proc. 6th annual international conference on Mobile computing and networking (MOBICOM '00), Boston, Massachusetts, USA, August 2000. – Boston, 2000.
4. Hu, Y. C. Ariadne: A secure On-Demand Routing Protocol for Ad Hoc Networks / Y. C. Hu, A. Perrig and D. B. Johnson // Proc. 8th ACM International Conference on Mobile Computing and Networking, Westin Peachtree Plaza, Atlanta, Georgia, USA, September 2002. – Atlanta, 2002.
5. Djenouri, D. New Approach for Selfish Nodes Detection in Mobile Ad hoc Networks / D. Djenouri and N. Badache // Proc. Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecurComm'05), Athens, Greece, September 2005. – Athens, 2005.
6. Detecting Black Hole Attacks in Tactical MANETs using Topology Graphs / E. Gerhard's-Padilla, N. Aschenbruck, P. Martini, M. Jahnke and J. Tolle // Proc. of the 33rd IEEE Conference on Local Computer Networks (LCN), Dublin, Ireland, and October 2007. – Dublin, 2007.

7. A secure incentive protocol for mobile ad hoc networks / Y. Zhang, W. Lou, W. Liu and Y. Fang // *Wireless Networks journal*. – 2007. – Oct. – № 13(5). – P. 569–582.
8. Kurosawa, S. Detecting Blackhole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic Learning Method / S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, Y. Nemoto // *International Journal of Network Security*. – 2007. – Nov. – № 5(3). – P. 338–346.
9. Agrawal, P. Cooperative black and gray hole attacks in mobile ad hoc networks / P. Agrawal, R. K. Ghosh and S. K. Das // *Proc. of the 2nd International Conference on Ubiquitous Information Management and Communication (ICUIMC 2008)*, SKKU, Suwon, Korea, Jan/Feb 2008. – Suwon, 2008.
10. Amitabh, M. Security and quality of service in ad hoc wireless networks / M. Amitabh. – 1st ed. – Cambridge University Press, 2008.
11. RADAR: a ReputAtion-based scheme for Detecting Anomalous nodes in wireless mesh networks / Z. H. Zhang, F. Naït-abdesselam, P. H. Ho and X. Lin // *Proc. IEEE Wireless Communications and Networking Conference (WCNC2008)*. – Las Vegas, USA, 2008.
12. SA-OLSR: Security Aware Optimized Link State Routing for Mobile Ad Hoc Networks / B. Kannhavong, H. Nakamaya, Y. Nemoto, N. Kato and A. Jamalipour // *Proc. International Conference of Communication (ICC 2008)*. – Beijing, China, 2008.
13. S. Djahel, F. An Acknowledgment- Based Scheme to Defend against Cooperative Black Hole Attacks in Optimized Link State Routing Protocol / S. Djahel, F. Naït-Abdesselam and A. Khokhar // *Proc. of the International Conference on Communication (ICC 2008)*. – Beijing, China, 2008.
14. Li, Z. AWF-NA: A Complete Solution for Tampered Packet Detection in VANETs / Z. Li, C. Chigan and D. Wong // *Proc. Global Communications Conference (IEEE GLOBECOM 08)*. – New Orleans, LA, USA, 2008.
15. URL: www.cs.montana.edu/webworks.

Ю. Д. Ильина, А. П. Иванов

Россия, Пенза, Пензенский государственный университет

ОБЗОР МЕТОДОВ АНАЛИЗА СИГНАЛОВ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ

Приводится обзор методов анализа вида и параметров модуляции сигналов телекоммуникационных систем. По результатам анализа осуществлен выбор метода для практической реализации на цифровом процессоре обработки сигналов.

В течение одного сеанса связи в каналах могут изменяться вид модуляции и параметры используемых сигналов. Ряд задач обработки поступающего сигнала требует наиболее точных знаний о его модуляционной структуре. Этим обуславливается необходимость включения в тракты обработки средств автоматического определения параметров и видов модуляции принимаемых сигналов. Путем применения математических методов и средств вычислительной техники возможно реализовать определения априорно неизвестных модуляционных параметров в реальном масштабе времени.

Общая схема работы подсистемы анализа принимаемой смеси сигнала и шума приведена на рис. 1.

С точки зрения общей теории оптимального приема, описанной в [1], проблема определения модуляционной структуры сигнала сводится к следующему.

На временном интервале наблюдается смесь сигнала с оцениваемыми параметрами и шума канала приема. На основе расчета коэффициента правдоподобия необходимо принять решение о приеме сигнала с одним из возможных вариантов модуляционной структуры. На практике для этого потребуется оценить корреляционный интеграл входной смеси с i -м параметром, $i = 1 \dots L$. Величину L нельзя ограничить из-за непрерывности оцениваемых параметров сигнала, следовательно использование классических методов оптимальной оценки параметров становится невозможным.



Рис. 1. Общая схема работы подсистемы анализа

Помимо методов теории оптимального приема, используются методы статистической теории распознавания объектов [2]. Задача сводится к выбору решающего правила отнесения контрольной выборки к сформированному классу.

В настоящее время хорошо изучена задача распознавания для полностью описанных классов. Проверка гипотез основана на сравнении отношения правдоподобия с некоторым заранее известным порогом.

Методы, основанные на восстановлении функций распределения классов путем статистического оценивания параметров их законов распределения используются, если функции распределения известны с точностью до вектора параметров. Наиболее часто используются такие параметры, как математическое ожидание и дисперсия.

Методы распознавания, включающие вычисление статистических оценок функции распределения или функций плотности вероятности по обучающим выборкам используются, если вид функции распределения не известен с точностью до вектора параметров.

В настоящее время хорошо изучены методы вычисления оценок по выборкам конечного объема. К ним относятся метод построения диаграммы [3] и метод k -го соседа [4].

Использование метода построения диаграммы заключается в следующем. Пространство признаков разбивается на непересекающиеся подмножества. Подсчитывается относительное количество попадания значений из обучающей выборки в каждое подмножество. Проблемой является способ разбиения множества признаков на подмножества, поскольку мелкое разбиение приводит к возникновению «пустых» интервалов, а крупное – к огрублению оценки плотности вероятности. Кроме того, оценка при стремящемся к бесконечности объеме выборки сводится к функции, отличающейся от действительной плотности вероятности.

Основным принципом метода k -го соседа является то, что объект присваивается тому классу, который наиболее распространен среди его соседей. Этот метод применим к многомерным выборкам. В качестве эталонных описаний распознаваемых классов могут быть использованы и непосредственные значения обучающих выборок. В этом случае применяются ранговые критерии распознавания: обучающие выборки для каждого из классов и контрольная выборка «перемешиваются», после чего упорядочиваются по возрастанию или убыванию. Для каждого из классов вычисляется функция, характеризующая степень перемешивания. По максимуму полученной функции и принимается решение о принадлежности контрольной выборки к какому-либо из классов.

Для распознавания сигналов с частотной (ЧМ) и фазовой (ФМ) модуляцией, рассмотренных в [5], в [3] предлагается использовать разности отсчетов мгновенной фазы $\varphi(nT)$

$$\Delta\varphi(nT) = \varphi(nT) - \varphi((n-l)T),$$

где $\varphi(nT)$ – мгновенная фаза сигнала в момент времени nT ; l – индекс значения, которого устанавливается с учетом максимальной скорости анализируемого сигнала.

Если исследовать особенности поведения случайного процесса $\Delta\varphi(nT)$ и вычислить его параметр

$$p_1 = \frac{1}{N} \sum_{i=1}^{N^*} \Psi_i^1(nT),$$

где $\Psi_i^1(nT)=1$, если $\Delta\varphi(nT) \in [3\pi/8, 5\pi/8] \cup [7\pi/8, 9\pi/8] \cup [11\pi/8, 13\pi/8]$ и $\Psi_i^1(nT) = 0$ в противном случае; N – объем выборки.

Тогда решающее правило распознавания ЧМ и ФМ сигналов выглядит следующим образом

$p_1 > 0.82$ – принимается гипотеза ФМ,

$p_1 \leq 0.7$ – принимается гипотеза ЧМ,

$0.7 < p_1 < 0.82$ – происходит отказ от принятия решения.

Преимуществом метода является отсутствие необходимости синхронизации приема сигналов по несущей частоте, что позволяет достаточно просто реализовать данный метод на цифровом сигнальном процессоре обработки сигналов, например, TMS320C5510.

Список литературы

1. Тихонов, В. И. Оптимальный прием сигналов / В. И. Тихонов. – М. : Радио и связь, 1983. – 320 с.
2. Фомин, Я. А. Распознавание образов: теория и применение / Я. А. Фомин. – М. : ФАЗИС, 2012. – 429 с.
3. Степанов, А. В. Методы компьютерной обработки сигналов систем радиосвязи / А. В. Степанов, С. А. Матвеев. – М. : СОЛОН-Пресс, 2003. – 208 с.
4. Журавлев, Ю. И. Распознавание. Математические методы. Программная система. Практическое применение / Ю. И. Журавлев, В. В. Рязанов, О. В. Сенько. – М. : ФАЗИС, 2005. – 159 с.
5. Теория электрической связи : учеб. пособие / К. К. Васильев, В. А. Глушков, А. В. Дормидонтов, А. Г. Нестеренко ; под общ. ред. К. К. Васильева. – Ульяновск : УлГТУ, 2008. – 452 с.

Е. Д. Кашаев, А. П. Иванов, М. С. Тикин

Россия, Пенза, Пензенский государственный университет

ВЛИЯНИЕ ШУМА В РАДИОКАНАЛЕ НА ВЫПОЛНЕНИЕ АЛГОРИТМА ОБЕСПЕЧЕНИЯ ЦЕЛОСТНОСТИ СИГНАЛОВ УСТРОЙСТВА ПРЕОБРАЗОВАНИЯ СИГНАЛОВ

В данной работе описаны результаты исследования алгоритма автоматической коррекции многолучевости в коротковолновых каналах связи различных телекоммуникационных систем.

Для обеспечения уверенной передачи информации на больших территориях применяются различные системы и технические средства связи, обеспечивающие в совокупности достаточно высокую надежность информационных связей страны. Радиосвязь в коротковолновом (КВ) диапазоне играет важную роль как средство внутренней и международной, зонавой, подвижной и производственно-диспетчерской связи общего и ведомственного пользования. Она обеспечивает следующие службы: магистральную, зонавую и местную радиосвязь, сеть радиовещания, службу стандартных частот, служебные линии для земных станций спутниковой связи, авиационную связь, морскую связь и другие.

Одной из проблем, стоящих перед разработчиками аппаратуры передачи данных, является необходимость понижения процента ошибок демодулирования принимаемых сигналов, которое позволит уменьшить время установления синхронизации устройств преобразования сигналов (УПС) приемника и передатчика. Один из путей решения данной проблемы – применение алгоритмов обеспечения целостности сигналов УПС на физическом уровне.

Один из основных параметров КВ канала связи – многолучевость. Многолучевость в последовательных системах приводит к затянувшимся переходным процессам при передаче элементарных посылок. При этом отклики от предыдущих и последующих посылок влияют на решение относительно принимаемой в данный момент посылки, т.е. появляется межсимвольная интерференция.

Данная статья описывает исследование алгоритма обеспечения целостности сигналов УПС [1], позволяющего автоматически корректировать частотно-временные характеристики КВ каналов связи с количеством лучей от 1 до 3.

На рис. 1 показано уменьшение амплитуды отраженных лучей с задержкой 1 мс и амплитудой 0,5 относительно амплитуды прямого луча при применении итеративного алгоритма коррекции многолучевости.

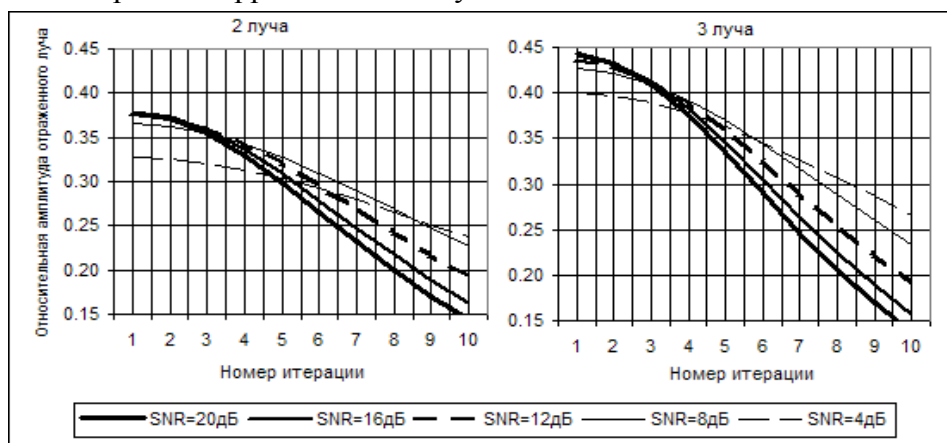


Рис. 1. Уменьшение амплитуды отраженных лучей относительно прямого луча при применении корректора многолучевости

Из рис. 1 видно, что выполнение десяти итераций алгоритма коррекции уменьшает амплитуду отраженного луча до значений от 0,15 до 0,27. Это означает, что исследуемый алгоритм за приемлемое время уменьшает амплитуду отраженных лучей в 2–3 раза.

На рис. 2 показаны результаты исследования влияния аддитивного шума на вероятность успешного выполнения алгоритма обеспечения целостности сигналов УПС, реализованного в корректоре многолучевости.

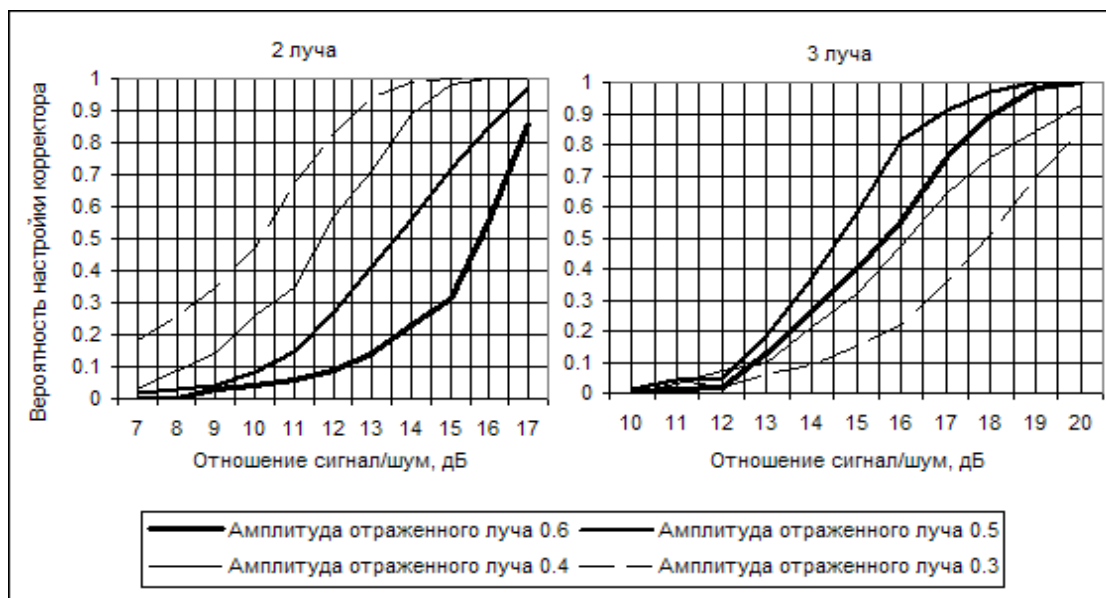


Рис. 2. Зависимость вероятности настройки корректора многолучевости от отношения сигнал/шум

Из рис. 2 видно, что рассматриваемый алгоритм коррекции с вероятностью 0,8 настраивается в двухлучевом канале при отношении сигнал/шум от 17 дБ до 22 дБ, а в трехлучевом – при отношении сигнал/шум от 16 до 20 дБ. Данные показатели являются приемлемыми для применения рассмотренного алгоритма в реальных телекоммуникационных системах.

Список литературы

1. Иванов, А. П. Алгоритм обеспечения целостности сигналов УПС на физическом уровне при воздействии линейных искажений / А. П. Иванов, М. С. Тикин // Вестник Пензенского государственного университета. – 2013. – № 1. – С. 122–128.

А. С. Крупский

Россия, Томск, Томский государственный университет систем управления и радиоэлектроники

МЕТОД РАСПЕРЕДЕЛЕНИЯ НАГРУЗКИ ПО ВРЕМЕНИ ПРИ РЕШЕНИИ ЗАДАЧ В ОДНОРОДНОЙ ВОСЬМИСВЯЗНОЙ СЕТИ

В работе рассмотрено превышение максимальной нагрузки на передающие мощности однородной компьютерной сети при распределенном решении вычислительной задачи, а также предложен метод разнесения нагрузки по времени для предотвращения коллизий.

Модель сети

Среди всего класса вычислительных задач возможно выделить такие, которые могут быть разбиты на независимые подзадачи. В этом случае, для координации процесса решения должен использоваться некоторый диспетчер, распределяющий подзадачи по вычислительным узлам. В качестве модели вычислительной сети рассмотрим однородную восьмисвязную сеть (рис. 1). На практике такой структурой может быть представлена совокупность одноранговых машин, будь то вычислительные узлы сенсорной сети или компьютеры отдела предприятия.

В общем случае, задача может возникнуть на любом из узлов сети (s). При распределенном решении вокруг узла-инициатора s возникают фронты узлов, участвующих в решении. Вычислительная мощность фронта определяется количеством узлов, входящих во фронт – длиной. Номером фронта будем называть расстояние r до его узлов от узла s . С увеличением фронта увеличивается задержка передачи и длина фронта.

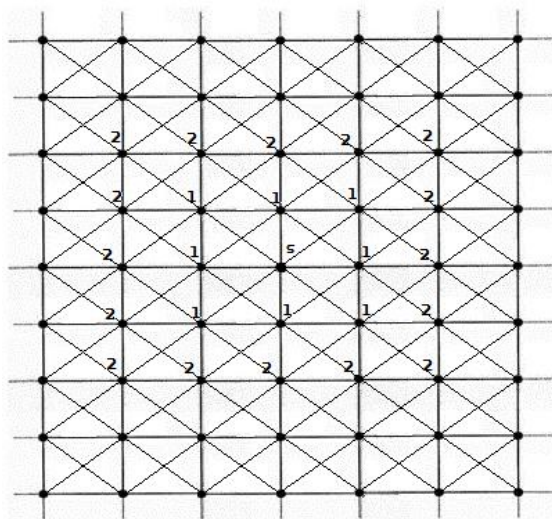


Рис. 1. Решение задачи в однородной сети
с помощью топологии с одним узлом-диспетчером

При таком подходе, факторами, ограничивающими масштабирование задачи, являются: задержка и пропускная способность связи.

Прохождение потоков данных

При наличии узла s узким местом системы в отношении объема передаваемых данных [1] является область, непосредственно прилегающая к узлу s . В момент

начала решения задачи узел s начинает передачу подзадач всем участвующим в решении фронтам. Графически этот процесс изображен на рис. 2.

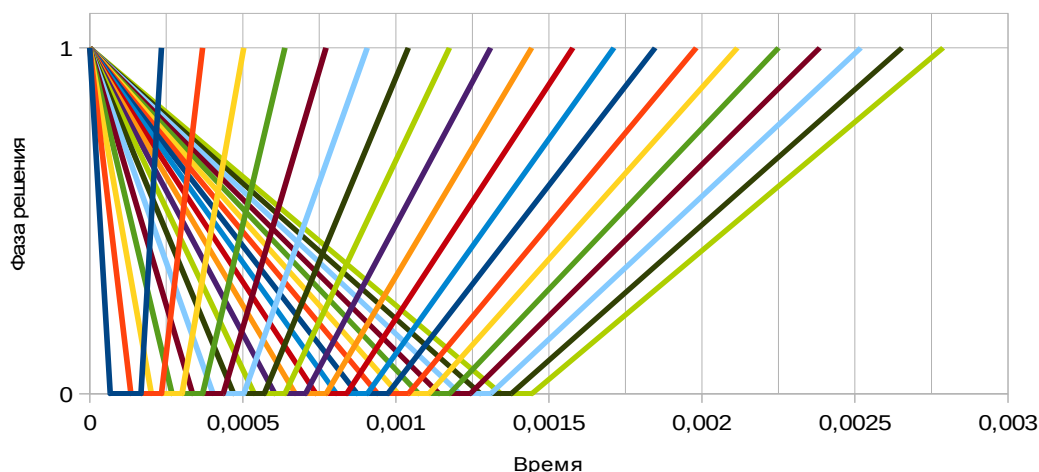


Рис. 2. Фазовая диаграмма загрузки сети передаваемыми данными

Единица на оси фазы решения соответствует наличию данных у узла s . Фаза 0 – решению задачи узлом-решателем. Таким образом, количество одновременно передающих узлов в сети представляет собой диаграмму (рис. 3).

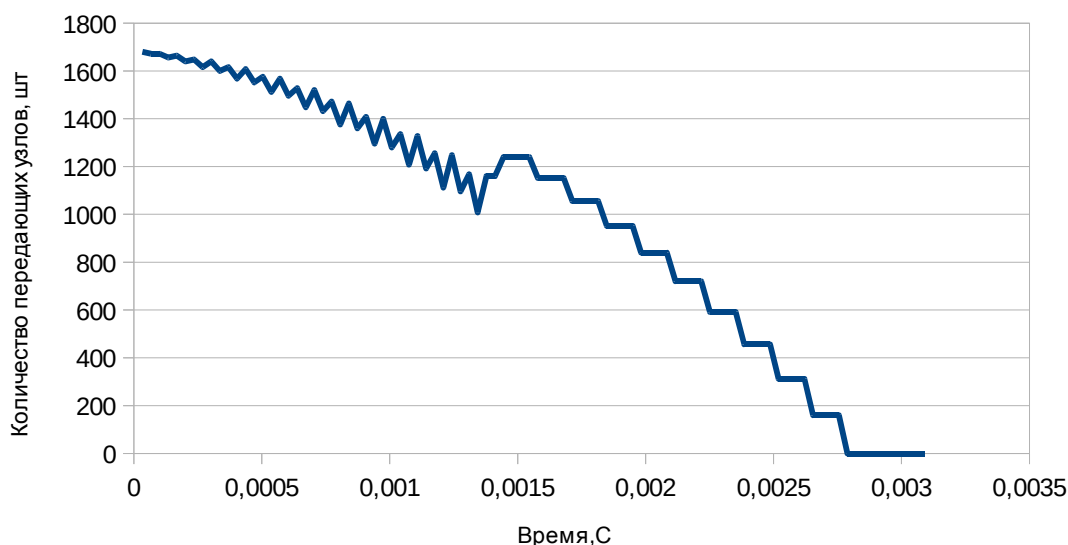


Рис. 3. Количество передающих узлов в сети

Как видно, нагрузка на сеть является неравномерной. При объеме передаваемых данных, превышающих пропускную способность сети. При нагрузке, превышающей максимально допустимую, в сети начинают возникать коллизии, для того чтобы разнести передачу по времени. Недостатком является то, что сеть в этом случае несет издержки, связанные с разрешением коллизий. Особенно это ощутимо для беспроводных сетей [2]. Для устранения этих издержек следует обеспечить нагрузку на сеть, не превышающую заданного уровня, с заданным числом коллизий. То есть требуется решить задачу минимизации по времени при заданном уровне нагрузки.

Процесс улучшения качества решения (уменьшения необходимого для этого времени) возможно провести графически: Среди всех фронтов выбирается фронт-донор – тот которому соответствует максимальное время окончания циклов решения, являющийся источником для переноса с него подзадач. Среди фронтов всех представленных радиусов r , начиная с самого малого, следует провести следующие действия: если фронт r не насыщен, то необходимо снять подзадачи с фронта-донора и дополнить фронт r до насыщенности. Если фронт r насыщен, то необходимо оценить: возможно ли добавить еще один цикл решения так, чтобы время окончания циклов решения данного фронта не превышало времени окончания циклов для фронта-донора.

Для фронта, не имеющего ни одного цикла, временем окончания цикла следует считать момент начала отсчета времени. В случае, несколько фронтов имеют одно и то же минимальное время окончания циклов, то следует выбрать в качестве фронта с минимальным временем окончания циклов тот фронт, у которого минимально r . В случае если несколько фронтов имеют одно и то же максимальное время окончания циклов, в качестве донора следует выбрать фронт с наибольшим r .

Данный алгоритм однозначно не увеличит время решения задачи, так как запрещается добавлять график, имеющий более позднее окончание, чем существующий. Средняя задержка ухудшится, в случае, если наиболее позднее окончание решения будет наблюдаться на фронте с меньшим номером, чем тот, на который происходит перемещение узлов. Однако, в таком случае, гарантированно уменьшится время решения задачи в целом, если фронт на который перемещаются узлы, не был занят. Это связано с тем, что количество узлов на каждом последующем фронте в восемь раз больше чем на предыдущем. Более того, такое отношение количества узлов делает вероятным (87,5 %) уменьшение времени решения даже в случае, если во фронте-реципиенте уже есть узлы.

С другой стороны, полное сокращение фронтов с большими порядками затруднительно в силу более низкой емкости фронтов с меньшим радиусом, однако даже в случае просто переноса части подзадач в фронты с меньшим радиусом, средняя задержка уменьшится. Данный алгоритм обеспечивает распределение подзадач по узлам, минимизируя общее время решения задачи при известном количестве и вычислительном объеме подзадач. Здесь речь идет о приведении уже существующего распределения подзадач по узлам к распределению с минимальным временем выполнения.

Список литературы

1. Крупский, А. С. Распределение задержки трафика в однородной восьмисвязной сети / А. С. Крупский, М. Ю. Катаев // Электронные средства и системы управления : материалы докл. X Междунар. науч.-практ. конф. – Томск : В-Спектр, 2014.
2. WiMAXQoSClasses // 2010 TranzeoWirelessTechnologies. – URL: http://www.tranzeo.com/allowed/Tranzeo_WiMAX_QoS_Classes_Whitepaper.pdf

МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

В. И. Волчихин, В. Н. Дубинин

Россия, Пенза, Пензенский государственный университет

МЕТОДИКА МОДЕЛИРОВАНИЯ НЕДЕТЕРМИНИРОВАННЫХ АВТОМАТОВ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА LD

Предлагается методика моделирования недетерминированных автоматов (НДА) с использованием языка LD, входящего в набор языков программирования ПЛК, определяемых стандартом IEC 61131-3. Для исключения неопределенности при параллельном изменении состояний НДА используется двухфазная схема. Для моделирования иерархических модульных НДА предлагается использовать дополнительно функционально-блоковые диаграммы.

Проблемы реализации НДА и их решение

Недетерминированные автоматы (НДА), определенные в работе [1], нашли практическое применение в проектировании аппаратных средств вычислительной техники и систем управления. Абстрактная модель НДА понятна и не вызывает вопросов. Все разрешенные переходы в НДА совершаются параллельно и без конфликтов. Проблемы возникают при реализации фрагментов НДА, связанных с вхождением-выходом состояний. Далее будем предполагать аппаратно-подобную реализацию НДА – чисто схемотехническую – на языке *VHDL* [2] или с помощью релейно-контактных схем, называемых в зарубежной литературе *Ladder diagram* (лестничные диаграммы или просто *LD*) [3], являющихся одним из языков программирования ПЛК в соответствии с международным стандартом IEC 61131-3. Первый опыт описания НДА на языке *VHDL* и их реализации на ПЛИС представлен в [4], а идея моделировать НДА с использованием языков ПЛК – в работе [5].

С каждым состоянием НДА будем связывать битовый элемент памяти (например, триггер или реле с самофиксацией), хранящим его статус. Для простоты считаем, что состояние и реализующий его элемент памяти имеют одинаковые имена. Если $S_i = 1$, то НДА находится в состоянии S_i , если $S_i = 0$, то нет. Назовем *входо-выходным* состоянием НДА состояние, в которое входит хотя бы одна дуга и одновременно выходит хотя бы одна дуга. Основная проблема при реализации НДА заключается в том, что возможны конфликты в элементах памяти при выполнении над ними нескольких параллельных операций. Для решения проблемы одновременного выполнения операций над одним элементом памяти предлагается двухтактное выполнение НДА. По аналогии с двухтактным триггером вводится понятие «двухтактного» или «двухкомпонентного» состояния. На первой фазе выполнения НДА синхронно производятся разрешенные (обычные) переходы состояний НДА, при этом устанавливаются первые компоненты состояний на основе вторых компонентов состояний. Таким образом, вторые компоненты состояний являются действующими, а вторые – буферными. Первая фаза является основной. Суть второй фазы связана с актуализацией вторых (основных) компонентов состояний на основе первых (буферных). Вторая фаза является вспомогательной.

На рис. 1 представлен пример расщепления состояния S на первую и вторую компоненты – S' и S'' . На данном рисунке сплошными дугами показаны переходы,

выполняемые на первой фазе. Пунктирной линией между подсостояниями отмечен переход, выполняемый на второй фазе. Выражения c_i ($i = 1, n$) – это условия перехода из состояний S_i ($i = 1, n$) в состояние S' .

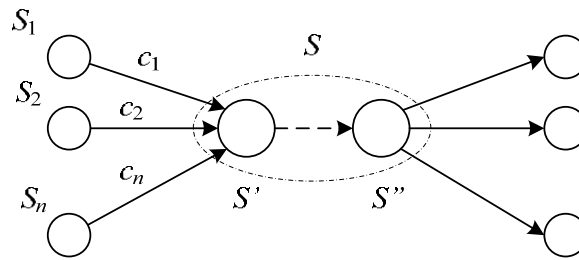


Рис. 1. Расщепление состояния на два компонента

Опишем более формально первую и вторую фазу.

На фазе 1 формула для определения активности первых компонентов состояний следующая:

$$S'(t+1) = \bigvee_{i=1}^n c_i(t) \& S_i(t). \quad (1)$$

В соответствии с данной формулой состояние автомата S' будет активным, если в автомате имеется хотя бы одно активное состояние, из которого в состояние S ведет истинный переход. Иначе состояние будет неактивным.

Вторая фаза описывается равенством:

$$S''(t+2) = S'(t+1). \quad (2)$$

В соответствии с данной формулой активность второго компонента состояния полагается равной активности соответствующей первой компоненты.

Для получения «реализационной» формы НДА предлагается правило расщепления состояния, приведенное на рис. 2.

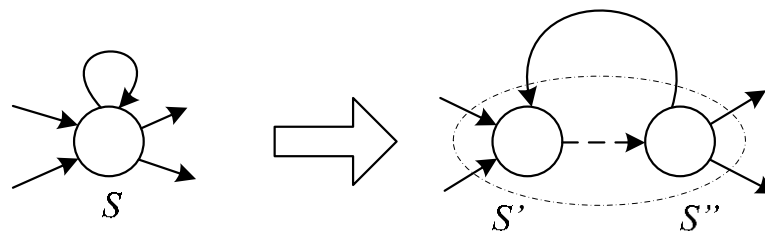


Рис. 2. Правило расщепления состояния

Пример преобразования НДА в реализационную форму

На рис. 3 изображен пример НДА из работы [1]. Этот автомат является довольно сложным и в нем можно найти все конструкции, интересные с точки зрения реализации.

На рис. 4 приведен автомат из рис. 3, подвергнутый преобразованию в соответствии с правилом из рис. 2 (т.е. показана реализационная форма). На первой фазе выполняются действия, отмеченные сплошными дугами, а на второй фазе – пунктирными. Начальное состояние НДА отмечено в виде маркера (метки) в вершине S_0'' .

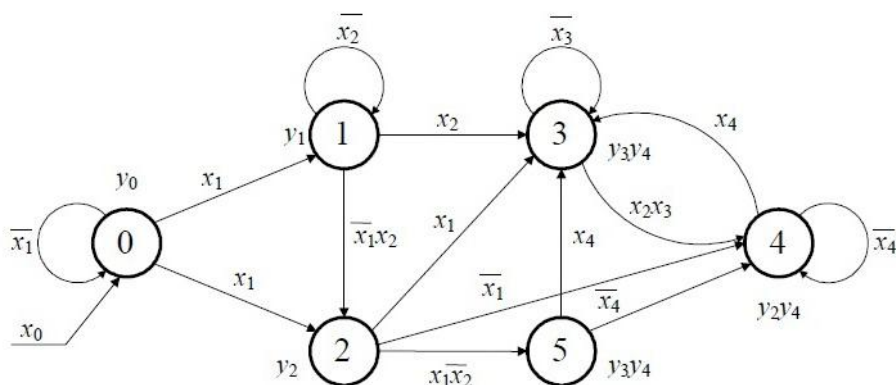


Рис. 3. Пример НДА

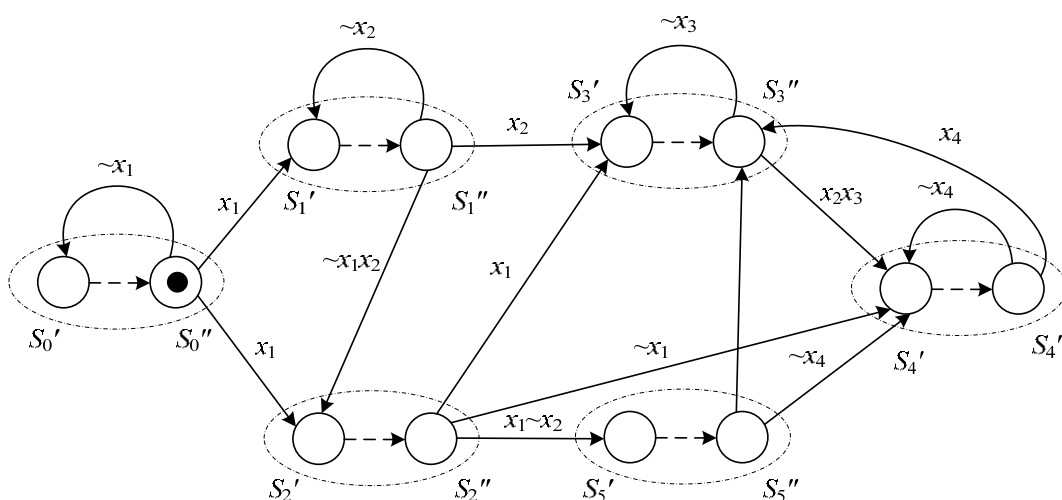


Рис. 4. Реализационная форма НДА, приведенного на рис. 3

Особенности аппаратной реализации НДА

Формулы (1) и (2) позволяют произвести аппаратную реализацию НДА, однако они все же являются довольно абстрактными для прямой реализации. Часто для работы с элементом памяти, представляющим состояние, требуется явное формирование двух сигналов – установки (SET) и сброса (RESET). Это относится как к триггерам в схемотехнике, так и к реле с самофиксацией в языке *LD*. Будем считать, что активным значением сигнала будет «1».

По формуле (1) могут быть сформированы следующие выражения для сигналов установки и сброса:

$$SET_{S_i} = \bigvee_{i=1}^n c_i(t) \& S_i(t) \quad (3)$$

$$RESET_{S_i} = \bigwedge_{i=1}^n (\overline{c_i(t)} \vee \overline{S_i(t)}) \quad (4)$$

Преобразование НДА в лестничную диаграмму

При реализации НДА на языке *LD* используется концепция реле с самофиксацией для моделирования элементов памяти, представляющих состояния. В соответствии с двухкомпонентным строением состояния реализационной формы НДА

будем выделять реле первой ступени – для реализации первых компонентов состояний, и реле второй ступени – соответственно для реализации вторых компонентов.

Чтобы сократить структурную сложность диаграммы за счет избежания формирования сигнала сброса RESET (4) предлагается расширить вторую фазу и дополнить формулу (2) следующей формулой:

$$S'(t+2) = 0 \quad (5)$$

В соответствии с данной формулой на второй фазе всегда сбрасываются все первые компоненты состояний. Опять же все просто на абстрактном уровне. На уровне реализации возникает необходимость во введении третьей фазы, поскольку на второй фазе над элементом памяти S' будут одновременно совершаться две операции – операция чтения (2) и операция сброса (5).

Однако язык лестничных диаграмм позволяет корректно выполнить эти действия за одну фазу (то есть как неделимую операцию) с сохранением порядка выполнения операций: сначала выполняется операция переписи информации из S' в S'' , а потом – операция сброса. Цепь, реализующая данные действия, изображена на рис. 5.

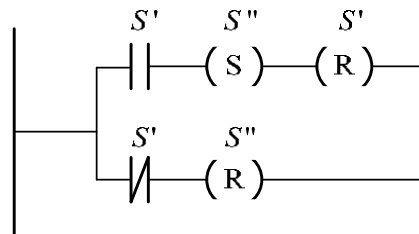


Рис. 5. Цепь для синхронной реализации операций переписи S' в S'' и сброса S'

Цепь работает следующим образом. Если S' установлено в единицу, то в единицу устанавливается и S'' и, одновременно с этим, сбрасывается S' . Сразу после совершения этих действий ток в цепи прерывается из-за размыкания контакта S' . Если же S' сброшено в ноль, то в ноль сбрасывается и S'' .

На рис. 6 приведена общая схема реализации НДА на языке LD при использовании двухфазного выполнения. Приведенная на рис. 6 лестничная диаграмма содержит три цепи. *Первая цепь* служит для инициализации системы, а именно – установки начального состояния. При поступлении сигнала $init$ в единицу устанавливаются те реле второй ступени ($S_k'', \dots, S_m''$), которые соответствуют маркированным состояниям в глобальном начальном состоянии. *Вторая цепь* реализует первую фазу выполнения НДА и используется для установки в единицу реле первой ступени, которые соответствуют первым компонентам состояний в реализационной форме НДА. Устанавливаются все те реле, для которых условие возбуждения C_i ($i = 1, n$) истинно. Условия возбуждения обозначены на рис. 6 овалами. *Третья цепь* предназначена для реализации второй фазы выполнения НДА. Третья цепь состоит из параллельно соединенных цепей, приведенных на рис. 5. Их число равно числу состояний НДА.

На рисунке 7 более подробно представлена структура цепи для установки реле первой ступени. Как видно из этого рисунка, условие возбуждения C_i представляет собой дизъюнкцию условий C_{ij} ($j=1, k$), каждое из которых представляет условие разрешенности перехода, ведущего в состояние S_i .

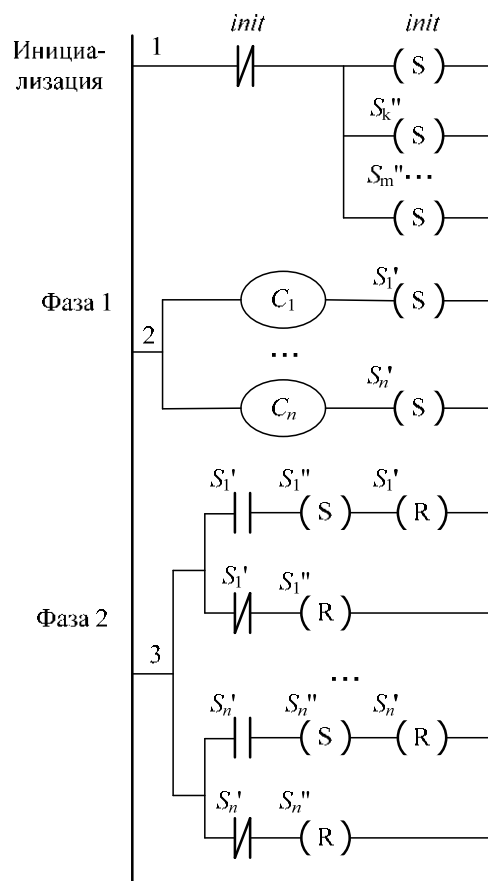


Рис. 6. Общая схема реализации НДА в виде лестничной диаграммы

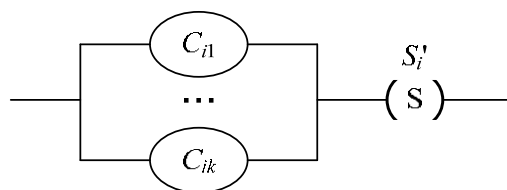


Рис. 7. Структура цепи для установки реле первой ступени

Условие разрешенности перехода из одного состояния НДА в другое определяется в виде конъюнкции входных сигналов и состояний, взятых с отрицанием и без [1]. Пример кодирования условия $x_1 \sim x_2 x_3 S_1 \sim S_2$ на языке *LD* приведен на рис. 8. Следует отметить, что в диаграмме *LD* используются контакты реле второй ступени.

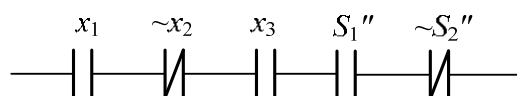


Рис. 8. Пример условия разрешенности перехода на языке LD

Представляет интерес моделирование расширения НДА – иерархических модульных недетерминированных автоматов (ИМНДА) [6] с использованием языков ПЛК. Базисный модуль ИМНДА может быть представлен с использованием лестничных диаграмм, как это было описано выше. Для моделирования составного модуля ИМНДА наиболее удобны функционально-блочные диаграммы (язык

FBD). Для автоматического преобразования ИМНДА, представленных в формате *XML*, в программы ПЛК, представленные в соответствии со стандартом *PLCOpen XML* [7], разработан соответствующий транслятор [8,9]. Область его применения - проектирование портативного программного обеспечения систем промышленной автоматики, основанных на ПЛК. Программа транслятора позволяет производить преобразования: 1) базисных модулей ИМНДА типа автоматов Мура в лестничные диаграммы (иначе, в программы на языке *LD*); 2) составных модулей ИМНДА в функционально-блоковые диаграммы (иначе, в программы на языке *FBD*). Для исключения неопределенности при параллельном изменении состояний ИМНДА используется двухфазная схема. Базисные модули ИМНДА представляются с помощью *XML DTD* системы *ACAD* [10], а составные модули – с помощью *XML DTD* системы *ViVe* [11]. Проверка целевого кода проводилась в системе *CODESYS V3.5* [12].

Представленная работа выполнена в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития разработки научно-технологического комплекса России на 2014–2020» (соглашение № 14.574.21.0045 от 19.06. 2014).

Список литературы

1. Вашкевич, Н. П. Недетерминированные автоматы в проектировании систем параллельной обработки / Н. П. Вашкевич. – Пенза : Изд-во Пенз. гос. ун-та, 2004. – 280 с.
2. Сергиенко, А. М. VHDL для проектирования вычислительных устройств / А. М. Сергиенко. – Киев : ЧП «Корнейчук», ООО «ТИД «ДС», 2003. – 208 с.
3. Петров, И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного программирования / И. В. Петров. – М. : СОЛОН-Пресс, 2004. – 256 с.
4. Вашкевич, Н. П. Аппаратная реализация функций синхронизации параллельных процессов при обращении к разделяемому ресурсу на основе ПЛИС / Н. П. Вашкевич, Р. А. Бикташев, Е. И. Гурин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 2. – С. 3–12.
5. Вашкевич, Н. П. Реализация недетерминированных автоматов на языке диаграмм Ладдера / Н. П. Вашкевич, В. Н. Дубинин // Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации (Расознавание-2013) : тез. докл. XI Междунар. науч.-техн. конф. – Курск, 2013. – С. 243–245.
6. Вашкевич, Н. П. Иерархические модульные недетерминированные автоматы / Н. П. Вашкевич, В. Н. Дубинин // Новые информационные технологии и системы (НИТС'2012) : тр. X Междунар. науч.-техн. конф. – Пенза, 2012. – С. 3–7.
7. PLCOpen for efficiency in automation. – URL: <http://www.plcopen.org/>
8. Вашкевич, Н. П. Транслятор XML-описаний иерархических модульных недетерминированных автоматов в программы ПЛК в формате PLCOpen XML / Н. П. Вашкевич, В. Н. Дубинин, Д. А. Будаговский // Проблемы техники и технологий телекоммуникаций (ПТиТТ-2013) : сб. тр. XIV Междунар. науч.-техн. конф. – Самара, 2013. – С. 245–247.
9. Свидетельство об официальной регистрации программы для ЭВМ № 2013619797. Транслятор XML-описаний иерархических модульных недетерминированных автоматов в программы ПЛК в формате PLCOpen XML / Будаговский Д. А., Дубинин В. Н., Вашкевич Н. П. – Заявка № 2013617483; Зарегистр. в реестре программ для ЭВМ 16.10.2013.
10. Лепилин, И. Б. Графический редактор сетевого представления недетерминированных автоматов / И. Б. Лепилин, Н. П. Вашкевич, В. Н. Дубинин // Хроники объединенного фонда электронных ресурсов «Наука и образование». – 2010. – № 2. – С. 20.
11. ViVe – VisualVerifier Tool Framework. – URL: <http://www.fb61499.com/license.html>
12. CoDeSys. – URL: <http://www.codesys.com/>

ПОЛУМАРКОВСКАЯ МОДЕЛЬ ПЕРЕХОДОВ СОСТОЯНИЙ ПАЦИЕНТА (БОЛЬНОГО) В ПРОЦЕССЕ ЛЕЧЕНИЯ ЗАБОЛЕВАНИЯ

Описана модель состояний пациента которая позволяет определить любое его целевое состояние, а именно исход лечения, зная только степень тяжести заболевания. Результат моделирования это последовательность переходов состояния пациента при лечении заболевания в соответствии с применяемой медицинской технологией.

Одним из инструментов, который становится все более доступным для решения вопросов эффективного управления здравоохранением, в общем, и медицинской организацией, в частности, является моделирование, в результате которого создаются возможности учесть влияние ряда основных факторов и их связей на результаты деятельности и своевременно принимать обоснованные управленческие решения. При наличии модели возникает целостное понимание работы, какой либо системы, влияния на нее различных факторов, что ведет к возможности управлять рядом из них, а соответственно добиваться максимального выполнения поставленных задач при оптимальном использовании ресурсов. Модель обладает большими возможностями прогнозирования в отличие от традиционных методов анализа деятельности, проводимого в медицинских организациях.

В проводимом исследовании основной акцент делается на аспекты управления лечебно-диагностическим процессом в медицинской организации с позиций его обеспеченности всеми видами ресурсов (персонал, медицинское оборудование, препараты и пр.), их использования и достижения заданного результата применяемой для лечения заболевания медицинской технологии.

Однако получить полную информационную картину как о состояниях пациента, так и о факторах, определяющие эти состояния, по данной модели не возможно. Поэтому предлагается следующее развитие модели пациента.

Приведенная на рис. 1 модель переходов состояний пациента и возможных исходов болезней при заболеваниях кардиологического профиля предполагает, что пациент поступает в медицинскую организацию как с заболеванием в легкой форме – состояние w_1 , – так и в одной из более тяжелых форм (табл. 1).

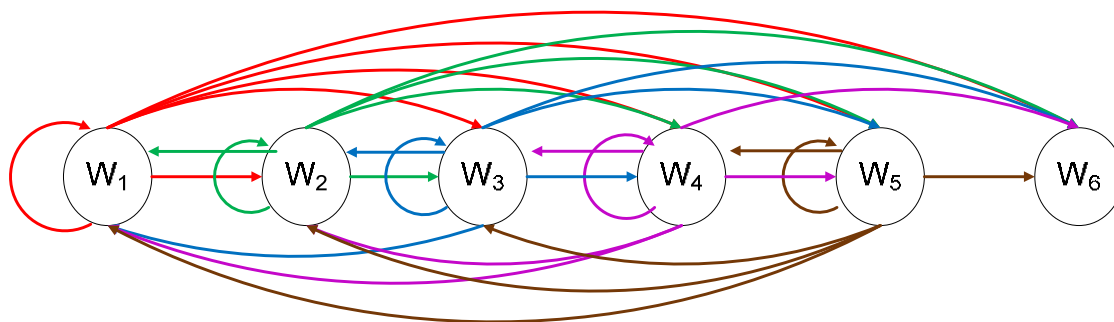


Рис. 1. Граф переходов состояний пациентов при применении медицинской технологии Q

В зависимости от степени тяжести заболевания возможными становятся различные варианты исходов лечения болезней. Состояние пациента также может последовательно ухудшаться. Применение той или иной медицинской технологии лечения заболевания влечет облегчение степени тяжести заболевания и определенный исход.

Таблица 1

Меры степени тяжести заболевания

№ п/п	Наименование степени тяжести заболевания	Мера тяжести заболевания	Примечания
1	Болезнь отсутствует	1	Значения 2, 4, 6, 8 соответствуют промежуточным состояниям тяжести заболевания
2	Легкое заболевание	2	
3	Заболевание средней тяжести	3	
4	Тяжелое заболевание	4	
5	Крайне тяжелое заболевание	5	
6	Летальный исход	6	

В предлагаемой модели считается, что заболевание может развиваться как постепенно – последовательное повышение степени тяжести болезни (развитие осложнений, обострение), так и «скачкообразно» – резкое ухудшение состояния здоровья пациента и, следовательно, резкое увеличение степени тяжести заболевания, вплоть до летального исхода.

Применение той или иной медицинской технологии лечения заболевания влечет облегчение степени тяжести заболевания и определенный исход. Описанная выше модель естественным образом формализуется с помощью математического полумарковского процесса.

Полумарковским (ПМ процессом) называется двумерный дискретно-непрерывный скачкообразный процесс

$$z(t) = \{v(t), x(t)\} = \{v(t), x(t); : \in [\xi_n, \xi_{n+1}], \xi_n = \sum_{k=1}^n x_k, n = 0, 1, \dots\},$$

с фазовым пространством $\Pi = I \times [0, \infty]$, $I = \{1, 2, \dots, m\}$, первая компонента которого $v_n = v(\xi_n) \in I$ характеризует номер состояния процесса после n-го перехода (скачка), а вторая компонента $x_n = x(\xi_n) \in [0, \infty]$ – время его пребывания в состоянии v_n до перехода в следующее состояние v_{n+1} .

Существует три способа задания ПМ-процесса.

ПМ – процесс задается матричной функцией $Q(t) = (Q_{i,j}(t))_{m \times m}$, элементами которой служат вероятности перехода процесса из состояния i в состояние j за время t : $Q_{i,j}(t) = P(v_{n+1} = j, x_n \leq t | v_n = i)$; и вектором $P(0)$ начального распределения вероятностей, $P(0) = (p_i(0))_{m \times 1}$, $p_i(0) = P(v_0 = i)$, $\sum_{i=1}^m p_i(0) = 1$.

Переходные вероятности $Q_{i,j}(t)$ должны удовлетворять следующим требованиям:

- $Q_{i,j}(0) = 0$; $Q_{i,j}(\infty) = \lim_{t \rightarrow \infty} Q_{i,j}(t) \leq 1$;
- $\forall t Q_{i,j}(t)$ – функция распределения момента перехода процесса из состояния i в состояние j ;
- $\forall i \sum_{j=1}^m Q_{i,j}(t) = 1$.

ПМ-процесс задается матрицей $F(t) = (F_{ij}(t))_{m \times m}$ условных функций распределения и матрицей переходных вероятностей $\Pi = (\pi_{ij})_{m \times m}$, где $F_{ij}(t) = P(T_{ij} < t) = P(x_n < 1 | v_n = i, v_{n+1} = j)$ – функция распределения времени T_{ij} пребывания процесса в состоянии i при условии, что он перейдет в состояние j ; $\pi_{ij} = P(v_{n+1} = j | v_n = i)$ – условная вероятность перехода процесса в состояние j из состояния i на n -м шаге; $\sum_{j=1}^m \pi_{ij} = 1$.

ПМ-процесс задается матрицей $T = (T_{ij}^{\sim})_{m \times m}$ независимых случайных величин, характеризующих время пребывания процесса в различных состояниях до перехода в другие состояния с функцией распределения $G_{ij}(t)$. Случайные величины $T_{ij}^{\sim}$ отличаются от T_{ij} тем, что они характеризуют время пребывания процесса в состоянии i до перехода в состояние j , как если бы этот переход был единственно возможным, т.е. осуществлялся независимо от других переходов.

Обозначим $T_i = \min_{j \neq i} T_{ij}^{\sim}$ – время пребывания в состоянии i .

Тогда $F_i(t) = P(T_i < t) = 1 - \prod_{j \neq i} [1 - G_{ij}(t)]$ – функция распределения времени T_i .

Все три способа задания ПМ-процесса эквивалентны между собой и связаны следующими соотношениями:

$$Q_{ij}(t) = \pi_{ij} F_{ij}(t); \quad \pi_{ij} = Q_{ij}(\infty); \quad (1)$$

$$F_{ij}(t) = \sum_{k \neq i} Q_{ik}(t);$$

$$Q_{ij} = \int_0^t \prod_{k \neq i, j} [1 - G_{ik}(t)] dG_{ij}(t). \quad (2)$$

Кроме того, используя равенства

$$F_i(t) = \sum_{j \neq i} Q_{ij}(t) = \sum_{j \neq i} \pi_{ij} F_{ij}(t) = 1 - \prod_{j \neq i} [1 - G_{ij}(t)],$$

получаем

$$\sum_{ij} \pi_{ij} [1 - F_{ij}(t)] = \prod_{j \neq i} [1 - G_{ij}(t)]. \quad (3)$$

При лечении пациента происходит частичное или полное восстановление.

Описанная модель состояний пациента позволяет определить любое его целевое состояние, а именно исход лечения, зная только степень тяжести заболевания. Построение марковской модели при имитации процессов лечения является индивидуальным, что означает построении такой модели для каждого пациента отдельно. Результатом моделирования будет последовательность переходов состояния пациента при лечении заболевания в соответствии с применяемой медицинской технологией.

В. С. Тимченко

Россия, Санкт-Петербург, Институт проблем транспорта им. Н. С. Соломенко
Российской академии наук

ПРИМЕНЕНИЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДЛЯ ОЦЕНКИ ПЕРЕРАБАТЫВАЮЩЕЙ СПОСОБНОСТИ ГРУЗОВОГО ФРОНТА

Представлено описание разработанной имитационной модели работы грузового фронта и результаты оценки его максимальной перерабатывающей способности в условиях обслуживания восьми категорий транспортных средств.

В Транспортной стратегии РФ на период до 2030 г. ставится задача интенсивного развития транспортной инфраструктуры. Одним из направлений ее научного обеспечения является создание имитационных систем, позволяющих моделировать системы различных видов транспорта.

Достоинством метода имитационного моделирования, является возможность автоматически определить значения параметров рассматриваемой системы, меняя при этом условия протекания процесса и случайные события, учет которых при традиционных подходах вызывает существенные затруднения. Это позволяет оперативно учитывать все изменения в проекте, а также получить более точные значения оптимальных параметров функционирования системы, чем при традиционно применяемом расчете [1].

Оценка пропускной способности транспортной инфраструктуры методом имитационного моделирования нашла свое применение на железнодорожном транспорте, в частности, для оценки пропускной способности железнодорожной линии Мга-Лужская, которая обслуживает морской торговый порт Усть-Луга, в условиях ее реконструкции с предоставлением большого количества «окон» в период до 2020 г. [2–4], а также для рассмотрения частных случаев функционирования железнодорожного транспорта [5–7].

Целью исследования является демонстрация возможностей метода имитационного моделирования при комплексном рассмотрении процесса функционирования объектов транспортной инфраструктуры.

Целью работы является оценка влияния процентного соотношения категорий транспортных средств, на максимальное количество транспортных средств, выгруженных на грузовом фронте в течение суток, на котором происходит обслуживание восьми различных категорий транспортных средств, время выгрузки которых находится в диапазоне от 20 до 34 мин с шагом в 2 мин.

Имитационная модель была построена в среде *AnyLogic*, структура которой представлена на рис. 1.

В имитационной модели заявки (транспортные средства) генерируются с заданной вероятностью, после чего они поступают в очередь на обслуживание блока (Ожидание_в_очереди), которая имеет максимальную вместимость, равную пяти заявкам.

Затем идет блок, который отвечает за одновременное обслуживание в системе только одной заявки. После него расположены три развилки, параметры которых позволяют задать вероятности появления различных категорий транспортных средств.

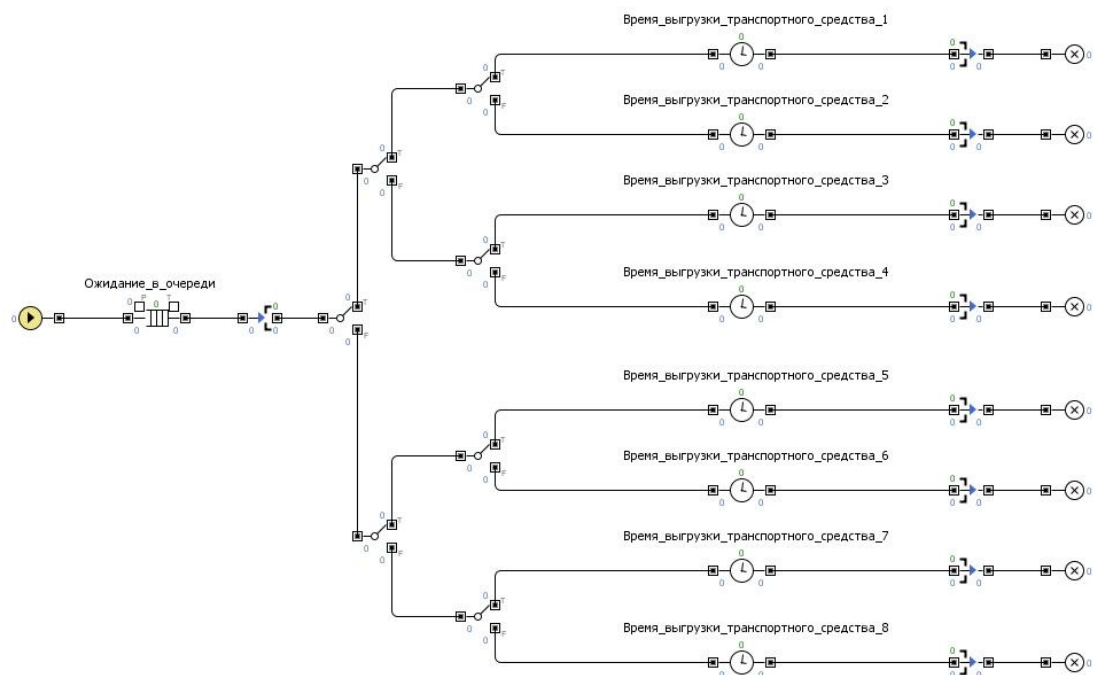


Рис. 1. Структура имитационной модели работы грузового фронта, при выгрузке четырех видов транспортных средств

Несмотря на то, что в рассматриваемой системе есть только один грузовой фронт, наличие восьми категорий транспортных средств с различным временем выгрузки, вызывает необходимость «размножить» количество грузовых фронтов в рамках имитационной модели, на каждом из которых будет обслуживаться своя категория транспортных средств, с длительностью выгрузки, заданной в блоке (Время_выгрузки_транспортного_средства).

Затем следует блок, который позволяет начать выгрузку следующего транспортного средства на грузовом фронте, после выгрузки предыдущего. Завершают имитационную модель блоки, удаляющие транспортные средства из имитационной модели и одновременно ведущие подсчет их количества по категориям.

Для наглядной иллюстрации преимуществ имитационной модели над аналитическим расчетом были проведены восемь расчетов, для случаев когда на грузовом фронте выполняется разгрузка только одной категории транспортных средств (отдельно для всех категорий). Данные расчеты можно также провести и на основании аналитически-детерминированных или аналитически-вероятностных зависимостей.

А девятый вариант расчета был проведен, в условиях разгрузки всех четырех категорий транспортных средств, вероятность появления которых в системе была задана одинаковой (0,125). Подобного рода результат без использования имитационного моделирования получить не представляется возможным.

Результаты расчетов приведены на рис. 2.

Кроме того, модель позволяет рассмотреть работу выгрузочного фронта, в том числе и при большем количестве категорий транспортных средств и различных их процентных соотношениях. При наличии законов распределения интенсивности поступления транспортных средств, имитационная модель позволяет не только оценить максимальную перерабатывающую способность, но и достаточность инфраструктурных и технологических мероприятий по ее увеличению, в

случае необходимости увеличения перерабатывающей способности грузового фронта.

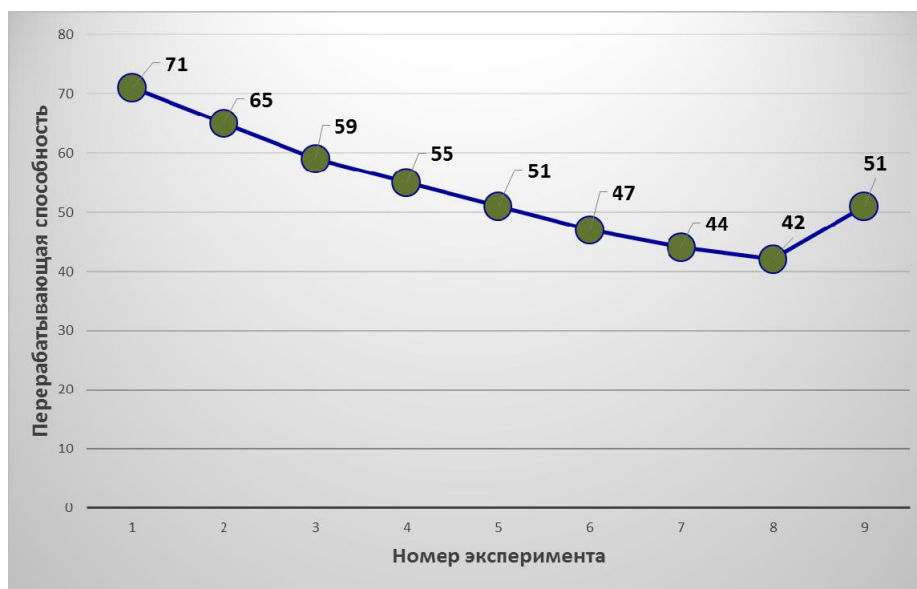


Рис. 2. Результаты расчета перерабатывающей способности грузового фронта

С помощью имитационной модели можно рассчитать следующие основные показатели эффективности работы грузового фронта, которые могут быть рассчитаны при наличии статистической информации функционирования реального грузового фронта, с целью поиска путей оптимизации его функционирования:

- Коэффициент загрузки каналов;
- Средняя длина очереди;
- Среднее время ожидания обслуживания;
- Вероятность ожидания обслуживания.

В статье представлена имитационной модели работы грузового фронта, которая позволяет оценить его перерабатывающую способность в условиях обслуживания восьми видов заявок, предназначенная для визуализации, анализа и поиска устойчивых параметров функционирования системы. Результаты имитационного моделирования могут быть использованы при выборе варианта реконструкции реального грузового фронта.

Список литературы

1. Имитационное моделирование как инструмент оценки инженерных решений при разработке проектов развития судостроительных и судоремонтных предприятий России / М. А. Долматов, Р. С. Нисенбаум, А. М. Плотников, Д. О. Федотов // Национальное общество имитационного моделирования. – URL: <http://simulation.su/uploads/files/default/ikm-mtmts-64-69.pdf> (дата обращения 19.09.2014 г.).
2. Кокурин, И. М. Методы определения «узких мест», ограничивающих пропускную способность железнодорожных направлений / И. М. Кокурин, В. С. Тимченко // Известия Петербургского университета путей сообщения. – 2013. – Вып. 1 (34). – С. 15–22.
3. Тимченко, В. С. Оценка перспективной пропускной способности участков железнодорожной сети с учетом предоставления «окон», на основе применения имитационного моделирования процессов перевозок / В. С. Тимченко // Молодой ученый. – 2014. – № 2. – С. 199–204.

4. Тимченко, В. С. Алгоритмизация процессов оценки пропускной способности железнодорожных участков в условиях предоставления окон / В. С. Тимченко // Транспорт Российской Федерации. – 2013. – № 5 (48). – С. 34–37.

5. Галкина, Ю. Е. Учет загруженности оперативного персонала при оценке пропускной способности железнодорожных направлений / Ю. Е. Галкина, К. Е. Ковалев, В. С. Тимченко // Наука и современность – 2014 : сб. материалов XXX Междунар. науч.-практ. конф. – Новосибирск : ЦРНС, 2014. – С. 131–136.

6. Ковалев, К. Е. Оценка мероприятий по развитию инфраструктуры в масштабе железнодорожных направлений с учетом загруженности оперативного персонала технических станций / К. Е. Ковалев, В. С. Тимченко // Молодой ученый. – 2014. – № 3. – С. 298–302.

7. Тимченко, В. С. Алгоритмы расчета графиков проведения ремонтных работ железнодорожного пути на перспективу / В. С. Тимченко // Наукoведение : интернет-журнал. – 2014. – № 3 (22). – URL: <http://naukovedenie.ru/PDF/06TVN314.pdf>

Д. В. Такташкин

Россия, Пенза, Пензенский государственный университет

КОМПЛЕКСНАЯ МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ПРОЦЕССА ОХЛАЖДЕНИЯ РЕЖУЩЕГО ИНСТРУМЕНТА

Рассматриваются проблемы моделирования тепло- и массообменных процессов в условиях охлаждения лезвийного инструмента распыленными смазочно-охлаждающими технологическими средствами. Разработана объектно-ориентированная модель, позволяющая рассчитывать процесс охлаждения режущего инструмента без использования сложных систем дифференциальных уравнений, и обоснована корректность предложенных моделей по результатам соответствующего экспериментального исследования.

Одним из наиболее перспективных способов повышения производительности процессов механической обработки лезвийным инструментом и снижения затрат на применение смазочно-охлаждающих технологических средств (СОТС) является их подача в зону обработки в распыленном состоянии [1, 2].

Сложность моделирования процесса охлаждения режущего инструмента распыленными технологическими средствами заключается главным образом в отсутствии математических моделей, комплексно описывающих этот процесс с учетом всех его наиболее значимых параметров [3, 4]. Возможным вариантом выхода из сложившейся ситуации может стать применение для решения данной проблемы объектно-ориентированного моделирования.

В результате декомпозиции системы были выделены пять основных объектов: «Капля СОТС», «Воздушная струя», «Зона обработки», «Воздушный поток» и «Тепловой поток». Все объекты системы взаимодействуют между собой на уровне методов [5].

В основу математической модели поведения объекта «Воздушная струя» положена теория Г. Н. Абрамовича [6]. С целью устранения вносимой в расчеты

ощутимой погрешности, обусловленной введенными в уравнения эмпирическими константами, предложено выразить текущий радиус воздушного потока R_i через тангенс угла раскрытия струи $\operatorname{tg} \alpha$, в результате чего получена новая уточненная формула:

$$R_i = \left(1 + \frac{a S_i}{0,29 R_0} \right) R_0,$$

где a – коэффициент структуры струи; S_i – расстояние между рассматриваемым сечением и срезом сопла; R_0 – радиус выходного сопла распыляющего устройства.

Для построения математической модели поведения объекта «Воздушный поток» рассмотрен случай вращения диска диаметром d_d с постоянной угловой скоростью ω_{ad} в покоящейся среде. В ходе анализа физики процесса установлено, что касательная составляющая скорости воздушного потока u_{ad} и его радиальная составляющая v_{ad} пропорциональны $\frac{d_d \omega_{ad}}{2}$ и выражаются следующими зависимостями [7]:

$$u_{ad} = \frac{d_d \omega_{ad}}{2} F \left(z_i \sqrt{\frac{\omega_{ad}}{\nu_v}} \right); \quad v_{ad} = \frac{d_d \omega_{ad}}{2} G \left(z_i \sqrt{\frac{\omega_{ad}}{\nu_v}} \right),$$

где z_i – расстояние от диска; δ_{ad} – толщина увлекаемого слоя среды; ν_v – кинематическая вязкость среды.

Разработанные объектно-ориентированные модели, описывающие аэродинамику газожидкостной струи аэрозоля СОТС, предоставили возможность оценить распределение жидкой фазы в объеме струи и непосредственно в зоне обработки.

Математическая модель поведения объекта «Капля СОТС» в контейнере «Струя аэрозоля СОТС» базируется на теории испарения Зака и законе испарения В. В. Шулейкина [8], которые дают достаточно хорошую зависимость скорости испарения от температуры окружающей среды:

$$W_v = \frac{W_0}{1 + \frac{\alpha r_k}{D} \left(\frac{RT_{sr}}{2\pi\mu} \right)^{1/2}},$$

где W_0 – скорость испарения в вакуум; α – коэффициент аккомодации; r_k – радиус капли; D – коэффициент диффузии паров; R – газовая постоянная, отнесенная к одному молю; T_{sr} – температура среды, в которую происходит испарение; μ – молекулярный вес.

На основе законов деформации и разрушения капель распыленного технологического средства в воздушном потоке разработана математическая модель поведения контейнера «Струя аэрозоля СОТС» при наличии в нем коалесценции и дробления. На основе формул Лапласа и Клапейрона–Клаузиуса [9] для случая перегрева капли технологического средства тепловым потоком выведено уравнение, характеризующее критическое время микровзрыва капли распыленной жидкости:

$$t_c = \sqrt{\frac{d_k^3 L \rho_n c_k \rho_k}{24 \varepsilon_k \sigma_k T_v^4 \lambda_k}},$$

где L – теплота парообразования; ρ_n – плотность пара при температуре кипения; c_k – удельная теплоемкость жидкости; ρ_k – плотность жидкости; ε_k – степень черноты капли; σ_k – коэффициент поверхностного натяжения жидкости; T_v – температура газовой среды; λ_k – коэффициент теплоотдачи излучением.

Разработанная математическая модель взаимодействия объектов «Капля СОТС» и «Тепловой поток» позволяет, зная температурные поля в зоне резания, определять дисперсность распыленного технологического средства и необходимую скорость струи аэрозоля.

Комплексный расчет двухфазного газожидкостного потока и процесса охлаждения режущего инструмента распыленными СОТС может быть представлен в виде диаграммы деятельности, приведенной на рис. 1. Как видно из рисунка, данные расчеты представляет собой совокупность ряда расчетов и связующих вычислений, полностью описывающих процесс охлаждения режущего инструмента с учетом всех значимых факторов.

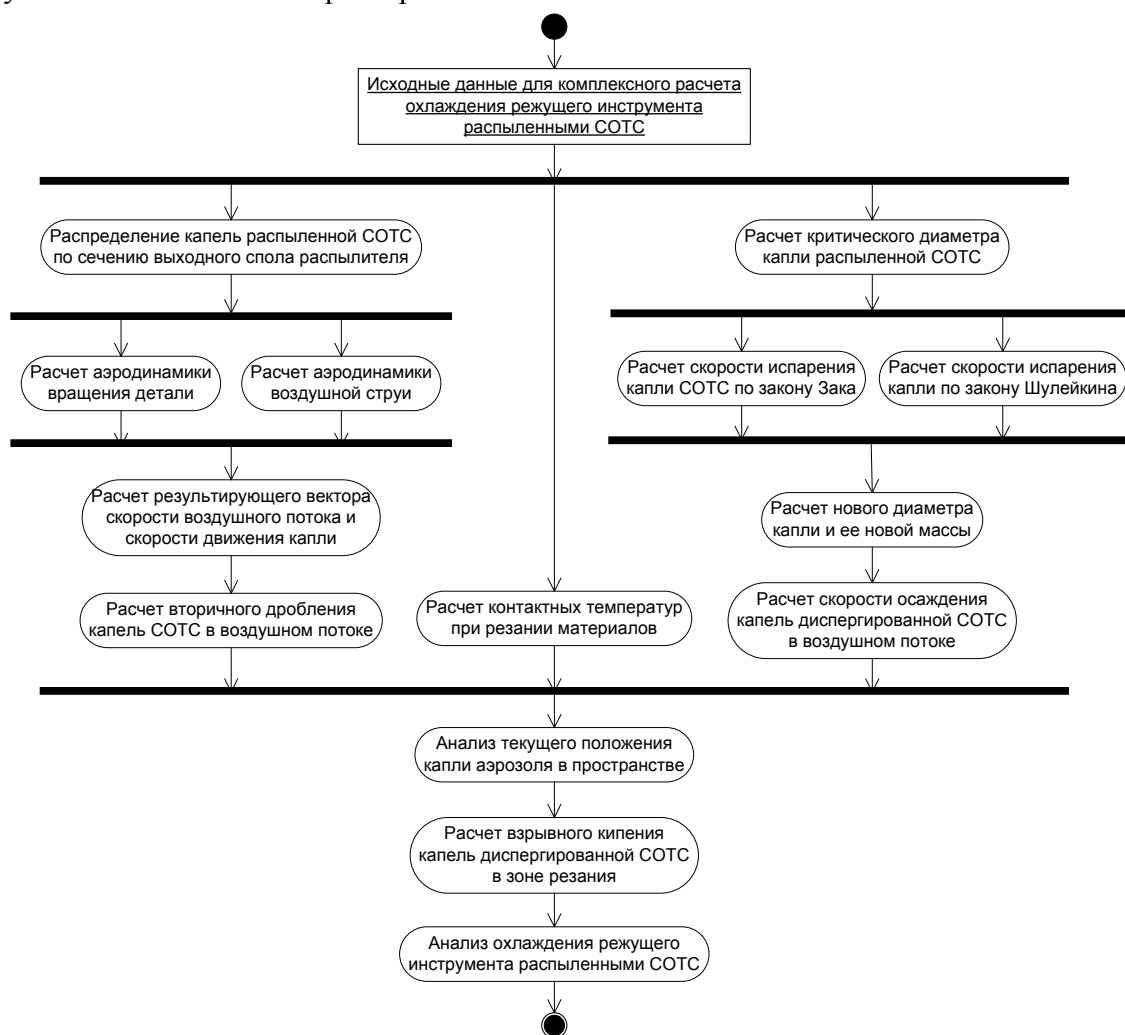


Рис. 1. Диаграмма деятельности компонентов расчета охлаждения режущего инструмента распыленными СОТС

Разработанные объектно-ориентированные модели, описывающие кинетику тепло-, массообменных процессов в зоне резания при охлаждении режущего инструмента распыленными технологическими средствами, дают возможность определить оптимальный расход СОТС, необходимый для обеспечения требуемого теплового баланса в зоне обработки.

Список литературы

1. Худобин, Л. В. Техника применения смазочно-охлаждающих средств в металлообработке / Л. В. Худобин, Е. Г. Бердичевский. – М. : Машиностроение, 1977. – 189 с.
2. Проскуряков, Ю. Г. Тонкораспыленное охлаждение режущих инструментов / Ю. Г. Проскуряков, В. Н. Петров. – М. : Машгиз, 1962. – 112 с.
3. Борискина, И. П. Моделирование процессов взаимодействия частиц в двухфазной среде : дис. ... канд. физ.-мат. наук : 05.13.18 / Борискина И. П. – Саранск, 2003. – 127 с.
4. Теряева, Н. Ю. Моделирование двухфазной среды и метод дискретных вихрей : дис. ... канд. физ.-мат. наук : 05.13.18 / Теряева Н. Ю. – Дубна, 2004. – 123 с.
5. Такташкин, Д. В. Объектно-ориентированное моделирование процесса охлаждения режущего инструмента / Д. В. Такташкин, В. И. Горбаченко // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2008. – № 1. – С. 3–15.
6. Абрамович, Г. Н. Прикладная газовая динамика / Г. Н. Абрамович. – М. : Изд-во технико-теоретической литературы, 1953. – 736 с.
7. Такташкин, Д. В. Математическая модель поля скоростей воздушных потоков у поверхности вращающейся заготовки / Д. В. Такташкин // Проблемы исследования и проектирования машин : материалы Междунар. науч.-техн. конф. – Пенза : ПДЗ, 2007. – С. 12–15.
8. Ирисов, А. С. Испаряемость топлив для поршневых двигателей и методы ее исследования / А. С. Ирисов. – М. : Изд-во Нефтяной и горно-топливной литературы, 1955. – 305 с.
9. Яворский, Б. М. Справочник по физике / Б. М. Яворский. – М. : Наука, 1968. – 940 с.

И. С. Васильев, С. А. Попов

**Россия, Великий Новгород, Новгородский государственный университет
им. Ярослава Мудрого**

ОЦЕНИВАНИЕ SPICE-ПАРАМЕТРОВ ТРАНЗИСТОРОВ ПРИ ЗАДАННОЙ ТОЧНОСТИ РАСЧЕТА ЭЛЕКТРИЧЕСКИХ РЕЖИМОВ ЭЛЕКТРОННЫХ УСТРОЙСТВ

Рассматривается оценивание Spice-параметров транзисторов и их ковариационной матрицы на основании многооткликовой модели эквивалентной схемы.

Для прогнозирования выходных электрических режимов электронных устройств необходимо знание SPICE-параметров транзисторов. Задача оценивания SPICE-параметров и их статистических характеристик формулируется как задача определения статистических оценок параметров многооткликовой модели по экспериментальным данным. Эта задача включает расчет вольтамперных характеристик (ВАХ), анализ ошибок наблюдений и построение плана эксперимента для расчета параметров математической модели. Многооткликовая модель ВАХ транзистора с абсолютной ошибкой представляется в виде:

$$\mathbf{I} = \mathbf{F}(\mathbf{B}, \mathbf{U}, \mathbf{I}) + \mathbf{R}, \quad (1)$$

где $\mathbf{F}(\mathbf{B}, \mathbf{U}, \mathbf{I})$ – система уравнений Кирхгофа, описывающая эквивалентную схему транзистора, $\mathbf{I} = \{i_1, \dots, i_k\}^T$ – вектор токов, $\mathbf{U} = \{u_1, \dots, u_m\}^T$ – вектор напряже-

ний, $\mathbf{B} = \{b_1, \dots, b_l\}^T$ – вектор коэффициентов модели, которые и являются определяемыми SPICE параметрами, $\mathbf{R} = \{r_1, \dots, r_m\}^T$ – абсолютная ошибка наблюдения.

При измерении тока в очень широких пределах абсолютная ошибка наблюдения также изменяется. На практике принимается, что абсолютная ошибка измерения пропорциональна величине измеряемой переменной. Таким образом, зависящая от величины тока абсолютная ошибка наблюдения тока $\mathbf{R}$ выражается через постоянную относительную ошибку наблюдения $\mathbf{E}$ в виде $\mathbf{R} = \mathbf{I}\mathbf{E}^T$. Тогда модель ВАХ с однородной ошибкой наблюдений выражается в виде

$$\ln \mathbf{I} = \ln[\mathbf{F}(\mathbf{B}, \mathbf{U}, \mathbf{I})] + \mathbf{E}, \quad (2)$$

где $\mathbf{E} = \{e_1, \dots, e_m\}^T$ – однородная ошибка наблюдений с ковариационной матрицей $\mathbf{V}_\mathbf{E}$.

Для расчета оценок параметров используется итерационная процедура, основанная на модифицированном методе Гаусса, который является достаточно быстродействующим и относительно устойчивым в вычислительном отношении. Оценки параметров, обеспечивающие максимум функции правдоподобия, в этом методе можно выразить в следующем виде:

$$\hat{\mathbf{B}}^{s+1} = \hat{\mathbf{B}}^s + \left[\sum_{j=1}^n \mathbf{P}(\hat{\mathbf{B}}^s, \mathbf{U}_j, \mathbf{I}_j) \mathbf{P}(\hat{\mathbf{B}}^s, \mathbf{U}_j, \mathbf{I}_j)^T \right]^{-1} \sum_{j=1}^n \mathbf{P}(\hat{\mathbf{B}}^s, \mathbf{U}_j, \mathbf{I}_j) [\ln \mathbf{I}_j - \ln \mathbf{I}_j^*], \quad (3)$$

где s – номер итерации, n – количество наблюдений ВАХ, $\mathbf{I}_j^*$ наблюденное значение тока, $\mathbf{I}_j$ – расчетное значение тока по модели при заданном напряжении в соответствии с выражением (2).

Векторная функция $\mathbf{P}(\mathbf{B}, \mathbf{U}, \mathbf{I})$ определяется как производная неявной функции (2) следующим образом:

$$\mathbf{P}(\mathbf{B}, \mathbf{U}, \mathbf{I}) = -\frac{1}{\mathbf{I}} \frac{\partial \mathbf{F}(\mathbf{B}, \mathbf{U}, \mathbf{I})}{\partial \mathbf{B}} [\mathbf{U}(\mathbf{B}, \mathbf{U}, \mathbf{I})]^{-1}, \quad (4)$$

где $\mathbf{L}$ – единичная матрица, а $\mathbf{U}(\mathbf{B}, \mathbf{U}, \mathbf{I}) = \left[\frac{\partial \mathbf{F}(\mathbf{B}, \mathbf{U}, \mathbf{I})}{\partial \mathbf{I}} - \mathbf{L} \right]$.

Ковариационная матрица $\mathbf{V}_\mathbf{B}$ оценок параметров рассчитывается по формуле

$$\mathbf{V}_\mathbf{B} \approx \left\{ \sum_{j=1}^n \mathbf{P}(\hat{\mathbf{B}}, \mathbf{U}_j, \mathbf{I}_j) \mathbf{V}_\mathbf{E}^{-1} \mathbf{P}(\hat{\mathbf{B}}, \mathbf{U}_j, \mathbf{I}_j)^T \right\}^{-1}. \quad (5)$$

Для реализации процедуры расчета (3) коэффициентов модели (2) на каждой итерации необходимо рассчитывать значение тока $\mathbf{I}$. Расчет выполняется путем решения системы уравнений ВАХ в виде

$$\mathbf{F}(\mathbf{B}, \mathbf{U}, \mathbf{I}) - \mathbf{I} = \Phi(\mathbf{B}, \mathbf{U}, \mathbf{I}) = 0. \quad (6)$$

Эта система нелинейных уравнений решается линейным разложением нелинейной функции $\Phi(\mathbf{B}, \mathbf{U}, \mathbf{I})$ в окрестности точки $\mathbf{I}^*$ в виде

$$\Phi(\mathbf{U}, \mathbf{B}, \mathbf{I}) \approx \Phi(\mathbf{B}, \mathbf{U}, \mathbf{I}^*) + \left[\frac{\partial \Phi(\mathbf{B}, \mathbf{U}, \mathbf{I})}{\partial \mathbf{I}} \Big|_{\mathbf{I}=\mathbf{I}^s} \right]^T (\mathbf{I} - \mathbf{I}^*).$$

Тогда итерационная процедура расчета вектора токов имеет вид

$$\mathbf{I}^{s+1} = \mathbf{I}^s - \Phi(\mathbf{B}, \mathbf{U}, \mathbf{I}^s) \left[\frac{\partial \Phi(\mathbf{B}, \mathbf{U}, \mathbf{I})}{\partial \mathbf{I}} \Big|_{\mathbf{I}=\mathbf{I}^s} - \mathbf{L} \right]^{-1}. \quad (7)$$

Исследования, проведенные методом статистического моделирования, показали, что итерационная процедура (7) имеет высокую скорость сходимости и обеспечивает устойчивое и точное решение системы (6) в диапазоне реальных значений параметров $\mathbf{B}$ внутри цикла (3).

Основной целью построения математической модели транзистора является предсказание значений параметров-критериев годности транзисторных устройств (например, транзисторных ключей). Оценка токов $\hat{\mathbf{I}}$ при заданных напряжениях определяется решением системы уравнений в виде

$$\mathbf{H}(\hat{\mathbf{B}}, \mathbf{Z}, \mathbf{U}, \hat{\mathbf{I}}) - \hat{\mathbf{I}} = 0, \quad (8)$$

где $\mathbf{H}(\hat{\mathbf{B}}, \mathbf{Z}, \mathbf{U}, \hat{\mathbf{I}})$ – система уравнений Кирхгофа, описывающая электронное устройство, $\hat{\mathbf{B}}$ – оценки SPICE-параметров транзисторов, получаемые по формуле (3), $\mathbf{Z}$ – известные компоненты электрической схемы, параметры которых можно измерить.

Система уравнений (8) решается с помощью итерационной процедуры, аналогичной (7). Если признать, что модель транзистора сама по себе правильна, то существуют два возможных источника ошибок предсказания выходных параметров устройства: ошибки в оценках коэффициентов и ошибки наблюдения откликов. Тогда ковариационная матрица ошибок предсказания выходных параметров рассчитывается следующим образом:

$$\mathbf{V}_Y = \mathbf{Q}(\hat{\mathbf{B}}, \mathbf{Z}, \mathbf{U}, \hat{\mathbf{I}}) \mathbf{V}_B (\mathbf{Q}(\hat{\mathbf{B}}, \mathbf{Z}, \mathbf{U}, \hat{\mathbf{I}}))^T + \mathbf{V}_E, \quad (9)$$

где $\mathbf{Q}(\hat{\mathbf{B}}, \mathbf{Z}, \mathbf{U}, \hat{\mathbf{I}})$ – матрица производных, получаемая по аналогии с (4).

Доверительные интервалы оцениваемых по модели (8) токов i_j выражаются в виде

$$\hat{i}_j - t_\alpha \sqrt{v_j} \leq i_j \leq \hat{i}_j + t_\alpha \sqrt{v_j}, \quad (10)$$

где $\hat{i}_j$ – оценка j -го тока по формуле (7); v_j – дисперсия оценки тока (j -й диагональный элемент ковариационной матрицы $\mathbf{V}_Y$); t_α – квантиль распределения Стьюдента на уровне значимости α и с числом степеней свободы $f = n - l/k$, $j = \overline{1, k}$.

В качестве критерия оптимальности плана в этом случае принимается определитель ковариационной матрицы $\det \mathbf{V}_Y$. Для расчета этого критерия оптимальности необходимо для заданного плана эксперимента рассчитать ковариационную матрицу оценок коэффициентов $\mathbf{V}_B$, а затем по формуле (9) рассчитать ковариационную матрицу оценок откликов $\mathbf{V}_Y$ и ее определитель. Оптимальные планы, полученные для этого критерия оптимальности, называются G -оптимальными. Из-

вестно, что при непрерывном планировании для построения G -оптимальных планов вместо критерия оптимальности в виде $\det \mathbf{V}_Y$ можно использовать критерий оптимальности в виде $\det \mathbf{V}_B$ (D -оптимальные планы). Это в значительной степени ускоряет процедуру планирования, поскольку упрощает вычисления критерия оптимальности.

В случае последовательного планирования задается некоторый начальный заранее заданный плана эксперимента с небольшим количеством точек наблюдений. Используя этот план, выполняются измерения и рассчитываются оценки коэффициентов модели. На основании полученных оценок отыскивается следующая точка наблюдения, обеспечивающая наилучшую точность оценивания. Количество измерений увеличивается до достижения необходимой точности оценивания параметров данного конкретного прибора. Алгоритм построения последовательного плана может быть представлен следующим выражением:

$$\det \mathbf{V}_B(\Pi + \mathbf{X}^0, \mathbf{B}) = \min_{\mathbf{X} \in \Sigma_X} \det \mathbf{V}_B(\Pi + \mathbf{X}, \mathbf{B}), \quad (11)$$

где Π – построенный план эксперимента; $\mathbf{X}^0$ – следующая добавляемая в план Π точка.

При разработке электронных схем, например транзисторных ключей, в качестве параметров-критериев годности задаются точки на ВАХ и соответствующие допустимые значения, например, для токов i'_j в виде

$$i_{mj} \geq i_{lj}, \quad (12)$$

где i_{mj} – измеренное значение тока; i_{lj} – нижнее допустимое значение тока годных изделий.

Чтобы сделать обоснованный (на уровне значимости α) вывод о годности прибора, необходимо, чтобы нижняя граница доверительного интервала (10) была больше допустимого значения i'_{lj} , т.е. требуется выполнение следующего неравенства:

$$\hat{i}_j - t_\alpha \sqrt{v_j} \geq i_{lj}. \quad (13)$$

Если выполняется неравенство

$$\hat{i}_j + t_\alpha \sqrt{v_j} \leq i_{lj}, \quad (14)$$

то прибор не соответствует ТУ по данному параметру. Если оба неравенства (13) и (14) не выполняются, то эксперимент нужно продолжать.

При выполнении неравенства (14) требуется коррекция технологического процесса с целью подгонки SPICE-параметров для получения требуемых значений параметров-критериев годности. Выражения (8) и (9) позволяют получать точечную и интервальную оценки параметров-критериев годности электронного устройства при заданных SPICE-параметрах транзистора, что дает возможность корректировать технологический процесс для обеспечения выполнения ТУ.

И. П. Болодурина, Д. И. Парфёнов

Россия, Оренбург, Оренбургский государственный университет

МОДЕЛИРОВАНИЕ ВИРТУАЛЬНЫХ ПОТОКОВ ОБЛАЧНОЙ СИСТЕМЫ И ОПТИМИЗАЦИЯ ДОСТУПА К ВЫЧИСЛИТЕЛЬНЫМ РЕСУРСАМ

В исследовании рассмотрена проблема формализации моделей облачных систем и оптимизации использования общих вычислительных ресурсов информационных систем.

В настоящее время проблема обработки и хранения больших объемов данных в современных информационных системах (ИС) решается посредством технологии облачных вычислений. Однако ИС также поддерживают ряд сервисов и приложений на единой платформе, имеющих индивидуальные особенности формирования потоков передачи данных. Это требует разработки соответствующих моделей, описывающих организацию самой архитектуры, потоков приложений, входящих в нее, а так же алгоритмов управления пулом вычислительных ресурсов (ВР) [2].

Концепция виртуализации вычислительных ресурсов строится на базе абстракций, представляющих собой кортежи отношений, связанных между собой элементами подмножеств. В настоящем исследовании нами разработана модель вычислительных ресурсов облачной системы. Облачную систему (ОбС) можно представить в виде связанных между собой множеств объектов, включающих в себя: вычислительные узлы $Snode$, систем хранения данных $Sstg$, сетевых хранилищ данных $Snas$ и серверов расписаний $Srasp$.

На каждом вычислительном узле одновременно запущено сразу несколько экземпляров виртуальных машин, описываемых кортежем $Snode_i = \{VM_{i,1}, VM_{i,2}, \dots, VM_{i,k}\}$, где k количество виртуальных машин (ВМ) на i -м вычислительном узле, $i=1...l$ (l - общее число узлов). При этом каждая ВМ содержит набор приложений $VM_j = \{App_{j,1}, App_{j,2}, \dots, App_{j,n}\}$, обеспечивающих, поддержку работы определенных сервисов, $j=1...m$ (m – число ВМ). В состав сетевого хранилища данных (СХД) входит набор готовых образов ВМ $Snas_y = \{VMimg_{y,1}, VMimg_{y,2}, \dots, VMimg_{y,p}\}$, $y=1...z$ (z число СХД), содержащих требуемую операционную систему с предустановленным программным обеспечением, и заданными параметрами виртуального аппаратного обеспечения $VMimg_{y,z} = \{OS_1, OS_2, \dots, OS_r\}$. Работа всей ОбС строится с использованием системы планирования выполнения каких-либо операций $Srasp = \{Rtask_1, Rtask_2, \dots, Rtask_f\}$, задаваемых посредством сервера расписаний. В распределенной СХД, состоящей, как правило, из отказоустойчивых RAID массивов $Sstg_f = \{RDisik_1, RDisik_2, \dots, RDisik_d\}$, осуществляется размещение необходимой для работы сервисов ИС информации $RDisik_d = \{Data_1, Data_2, \dots, Data_s\}$. Помимо этого, в облачной системе также присутствуют виртуальные и физические коммутаторы, соединяющие между собой все компоненты в единую сеть. Каждый из компонентов, входящих в множество узлов $Shcn = \{Snode, Snas, Srasp, Sstg, VM, \dots\}$ ОбС обладает следующими характеристиками:

$$Shcn = (State, Mem, Disk, Diskn, Core, Lan), \quad (1)$$

где, $State \in \{\text{"on"}, \text{"off"}\}$ – состояние объекта ОбС; $Mem \in N$ – объем оперативной памяти установленной для узла ОбС; $Disk \in N$ – объем дискового пространства устройств хранения данных, установленных для узла ОбС; $Diskn \in N$ – количество устройств хранения данных, установленных для узла облачной системы;

$Core \in N$ – количество ядер процессора узла ОбС; $Lan \in N$ – максимальная пропускная способность сетевого адаптера узла ОбС.

При этом для виртуальных машин из множества $Snode$, может быть создано подмножество $VMnode = \{Snode, Snas, Sstg, \dots\}$, включающее в себя все компоненты ОбС, что позволяет масштабировать архитектуру и изолировать вычислительные ресурсы для различных сервисов друг от друга. Декомпозиция ресурсов облачной системы и ее компонентов представлена на рис. 1.

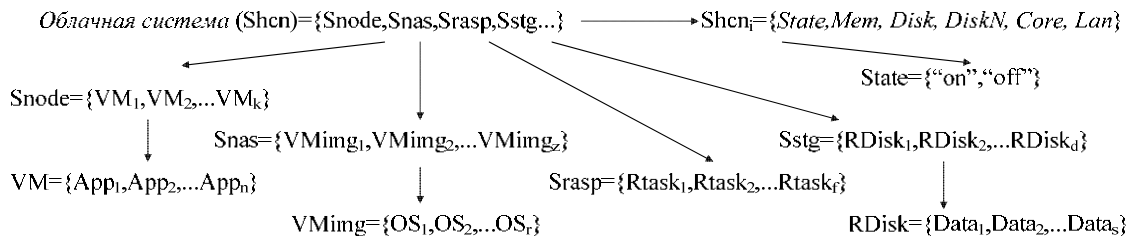


Рис. 1. Декомпозиция ключевых ресурсов облачной системы

Облачная система – динамический объект, изменяющийся в моменты времени t , ее состояние можно формализовать в виде ориентированного графа:

$$Shcn(t) = (Node(t), Connections(t), App(t)), \quad (2)$$

где, $Node(t) = \{Node_1, Node_2, \dots, Node_\lambda\}$ – вершины в виде активных элементы, входящие в одно из множеств $Snode_i$, $Sstg_j$, $Snas_k$, $Srasp_m$; $Connections(t) = \{Connections_1, Connections_2, \dots, Connections_v\}$ – активные подключения пользователей к приложениям; $App(t) = \{App_1, App_2, \dots, App_n\}$ – активные экземпляры приложений, запущенные на виртуальных ресурсах.

Таким образом, определен состав вычислительных ресурсов, описаны структура связей и механизмы взаимодействия основных компонентов облачной системы. Для оптимизации предоставления доступа к ресурсам ИС необходимо так же описать потоки передаваемых данных в ОбС.

Потоки передаваемых данных в информационных системах, как правило, обрабатываются за несколько фаз обслуживания. Интенсивность поступления и обслуживания рассматриваемых потоков запросов к мультимедийным сервисам ИС на каждой фазе, зависит от класса, определяемого уровнем конечного приложения [1]. При этом для каждого класса характерна собственная функция распределения соответствующая следующим законам:

1. для потоковых данных – распределение Парето;
2. для данных большего размера (файлы) – распределение Вейбулла;
3. для данных не большего размера (текст) – распределение Хи-квадрат.

Для описания динамики обслуживания заявок и траектории прохождения в многофазной ИС, использующей облачные ресурсы, формализовано описание потоков передаваемых данных в виде отображения, обеспечивающего наиболее эффективное обслуживание запросов пользователей $R^j: X^j \rightarrow Y^j$, где R^j – результирующий поток запросов, проходящий через все элементы j -й фазы; X^j – множество всех потоков, входящих в элементы j -й фазы; Y^j – множество всех потоков, выходящих из элементов j -й фазы; $j = \overline{1, f}$ – количество выделенных фаз

обслуживания. Потоки данных ИС, расположенной в облаке, можно представить в виде:

$$Y_i^j = R^j(X_i^j, t, F) = \begin{cases} R(X_i^j, t, F), & X_i^j \in X^j \\ P_Y^j(X^{j*}, t, F), & X^{j*} \in \bigcup_{S_i^j} \left[X_i^{j,0} \bigcup \left(\bigcup_{S_i^{j-1}} X_i^{j,j-1} \right) \right] \\ Q_X^j(Y^{j*}, t, F), & Y^{j*} \in \bigcup_{S_i^j} \left[Y_i^{j,0} \bigcup \left(\bigcup_{S_i^{j+1}} Y_i^{j,j+1} \right) \right] \end{cases}, \quad (3)$$

где $R(X_i^j, t, F)$ определяет потоки между элементами внутри фазы; $P_Y^j(X^{j*}, t, F)$ характеризует выходящие пересекающиеся потоки между фазами; $Q_X^j(Y^{j*}, t, F)$ характеризует входящие пересекающиеся потоки между фазами в момент времени t , в зависимости от внутренних и внешних процессов F .

В построенной модели количество источников и интенсивность формирования запросов каждым из них напрямую зависят от количества пользователей, обращающихся в данный момент к ИС. При этом сессии пользователей к приложениям на разных уровнях учитываются независимо друг от друга.

Представленная модель потоков позволяет определять наиболее подходящие вычислительные узлы ИС и ВМ, содержащие требуемые экземпляры приложений. Однако для предоставления доступа к ИС необходимо осуществить запуск всей инфраструктуры ОбС и обеспечить эффективное управление виртуальными ресурсами и приложениями.

Основной задачей управления ВР является выбор их количества в каждый последующий момент времени, то есть составление плана. При организации доступа к высоконагруженным ИС эффективное планирование особенно актуально, так как создаваемая нагрузка на сервисы изменяется в широких пределах за достаточно короткие интервалы времени. Как правило, в распределение ресурсов состоит в двух связанных друг с другом процессах: сбор, анализ и классификация данных; непосредственное обслуживание запросов.

В проведенном нами исследовании разработан алгоритм, позволяющий осуществлять динамическое планирование ресурсов облачной системы, используя статистику обращений пользователей к различным классам приложений ИС. В алгоритме применяются несколько шаблонов-образов виртуальных машин, запускаемых в зависимости от востребованности приложений. Это позволяет учитывать физические ограничения ВР, и организовывать работу ИС с учетом входящего потока запросов пользователей регулируя при этом число запущенных экземпляров приложений и ВМ.

Для оценки эффективности алгоритма управления ВР ОбС проведено исследование работы ИС с различными параметрами. При этом в качестве эталонных данных использовались результаты работы типовых алгоритмов ОбС Openstack. В ходе эксперимента для анализа данных создан поток запросов, аналогичный реальному трафику ИС вуза. При этом количество одновременных запросов, поступивших в систему, составило 10000. На рис. 2 представлен сопоставительный график, показывающий эффективность работы системы управления вычислительными ресурсами ОбС с модифицированным алгоритмом.

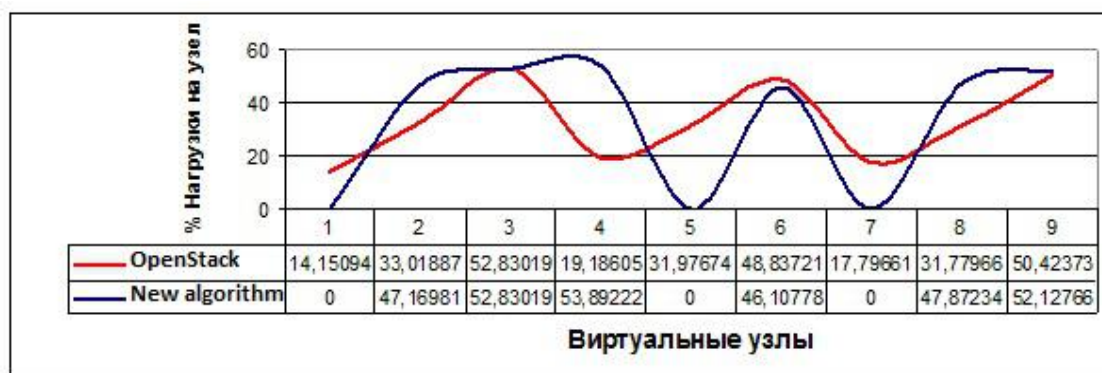


Рис. 2. График эффективности использования вычислительных ресурсов облачной системы

За счет оптимального выделения ресурсов на каждом вычислительном узле гарантировано обеспечение совместной работы всех запущенных экземпляров приложений, что удовлетворяет требованиям потенциальных пользователей. При этом благодаря работе алгоритмов оптимизации возможно высвобождение от 20 до 30% выделенных ресурсов. Кроме того, оценивая общий результат работы алгоритма управления ВР ОбС можно получить прирост производительности от 12 до 15% по сравнению со стандартными средствами. Кроме того сокращение числа выделенных ресурсов позволяет более эффективно масштабировать ОбС.

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проекты № 13-07-00198 и 13-01-97050).

Список литературы

1. Парфёнов, Д. И. Управление потоками запросов при доступе к широкополосным мультимедийным образовательным ресурсам системы дистанционного обучения / Д. И. Парфёнов, И. П. Болодурина // Проблемы управления. – 2013. – № 6. – С. 24–30.
2. Проблема адекватной оценки производительности веб-серверов в корпоративных сетях на предприятиях ЦБП / О. В. Гусев, А. В. Жуков, В. В. Поляков, С. В. Поляков // НИТ в ЦБП и энергетике : материалы 6-й науч.-техн. конф. – Петрозаводск, 2004. – С. 84–87.

С. А. Зинкин, О. Е. Шеланков

Россия, Пенза, Пензенский государственный университет,
Научно-производственное предприятие «Рубин»

ОБОСНОВАНИЕ СОЗДАНИЯ И ПРИМЕНЕНИЯ АЛГЕБРЫ СОБЫТИЙНО-ИЕРАРХИЧЕСКИХ ФРЕЙМОВ

Приводятся основные положения по формированию событийно-иерархической фреймовой сети на основе алгебры событийно-иерархических фреймов, фреймов-действий и событийно-иерархических фреймов.

Особенностью представления знаний о событиях информационных динамических процессов является формирование событийно-иерархической фреймовой сети посредством фреймов-действий, событийно-иерархических фреймов и алгеб-

раических операций. В связи с этим исследование свойств событийно-иерархических фреймовых сетей информационных динамических процессов описываемой предметной области наиболее целесообразно проводить методами теории алгебраических систем, основоположником которой является А.И. Мальцев.

В соответствии с положениями теории алгебраических систем событийно-иерархическая фреймовая сеть представляется в виде двухосновной алгебры событийно-иерархических фреймов.

$$A = (Fr, \alpha),$$

где A – алгебра событийно-иерархических фреймов; Fr – алгебра действий; α – алгебра условий.

Алгебра действий состоит из фреймов-действий и событийно-иерархических фреймов, которые определяют состояния информационных динамических процессов, а алгебра условий – из предикатов, принимающих значение $\{0, 1\}$.

Операции алгебры событийно-иерархических фреймов делятся на операции алгебры действий и операции алгебры условий. Первые принимают значения в множестве фреймов-действий и событийно-иерархических фреймов, вторые в множестве условий.

Операции алгебры действий:

1. Фреймовая каузальность (α_n),

$$\alpha_n = \langle Fr_1 \rightarrow Fr_j \rangle,$$

определенная для любого условия и фреймов-действий (событийно-иерархических фреймов). Если сработал фрейм Fr_1 , то срабатывает фрейм Fr_j . То есть причиной возникновения события, описываемого фреймом Fr_j , является действие или событие, описываемых фреймом Fr_1 .

2. Асинхронное «И» ($\alpha_{ак}$),

$$\alpha_{ак} = \langle Fr_1 \wedge \dots \wedge Fr_j \rangle$$

Алгебраическая операция асинхронного «И» определяет, что событие Fr_z может произойти только тогда, когда произойдут все действия или события, описываемые фреймами Fr_i и Fr_j .

3. Синхронное «И» ($\alpha_{ск}$),

$$\alpha_{ск} = \langle Fr_1 \wedge \dots \wedge Fr_j \rangle$$

Алгебраическая операция синхронного «И» определяет, что событие Fr_z может произойти только тогда, когда одновременно произойдут все действия или события, описываемые фреймами Fr_i и Fr_j .

4. Асинхронное «ИЛИ» ($\alpha_{ад}$),

$$\alpha_{ад} = \langle Fr_1 \vee \dots \vee Fr_j \rangle$$

Алгебраическая операция асинхронного «ИЛИ» определяет, что событие Fr_z может произойти только тогда, когда произойдет или одно или несколько действий (событий), описываемых фреймами Fr_i и Fr_j .

5. Синхронное «ИЛИ» ($\alpha_{сд}$),

$$\alpha_{сд} = \langle Fr_1 \vee \dots \vee Fr_j \rangle$$

Алгебраическая операция синхронного «ИЛИ» определяет, что событие Fr_z , может произойти только тогда, когда одновременно произойдут несколько действий (событий), описываемых фреймами Fr_i и Fr_j .

6. Исключающее «ИЛИ» ($\alpha_{искл}$).

$$\alpha_{искл} = \{Fr_i \vee \dots \vee Fr_j\}$$

Алгебраическая операция исключающего «ИЛИ» определяет, что событие Fr_z , может произойти только тогда, когда произойдет только одно действие (событие), описываемое фреймами Fr_i и Fr_j .

Типовое представление фреймов-действий с учетом алгебраических операций изображено на рис. 1.

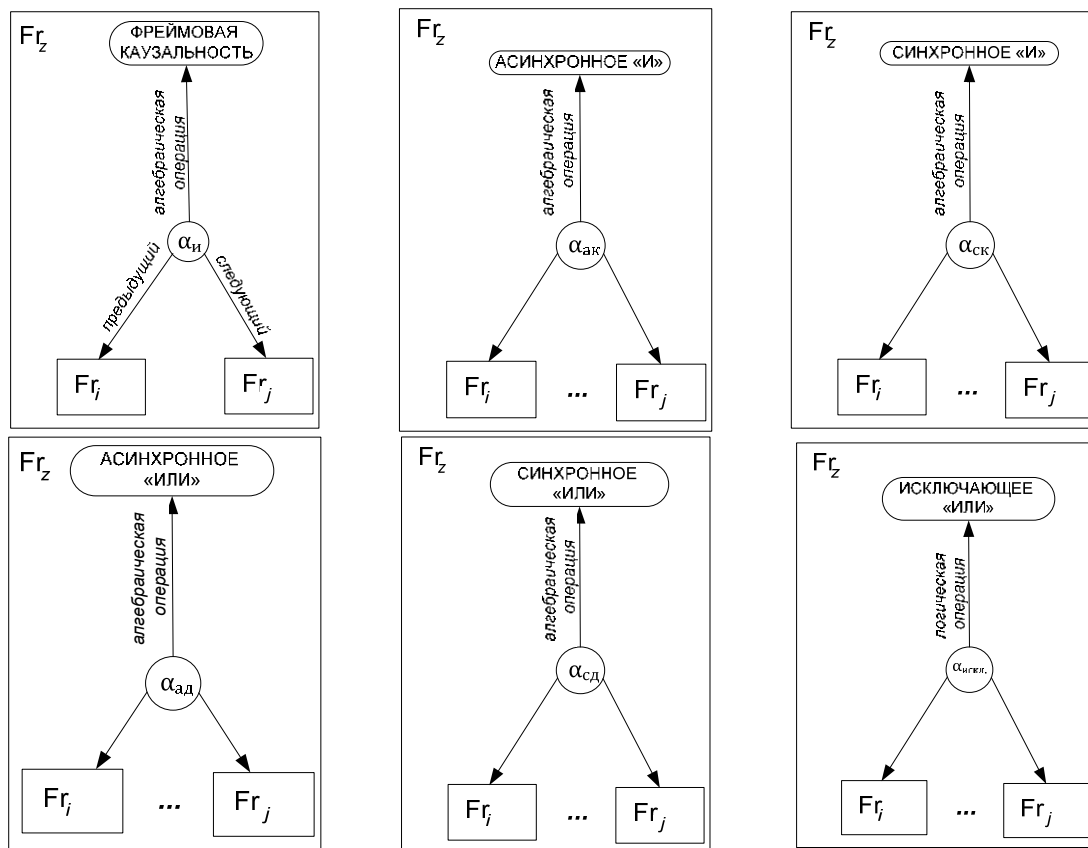


Рис. 1. Внешний вид типовых фреймов-действий

Операции алгебры условий представляют собой логические операции, соответствующие положениям булевой алгебры, кроме закона исключенного третьего и закона противоречия.

Таким образом, совершение действий и возникновение событий можно представить в следующем виде:

$$f(\alpha, Fr) = \begin{cases} Fr_{событ}, & \text{если } P(\alpha, Fr) = 1 \\ 0, & \text{если } P(\alpha, Fr) = 0 \end{cases},$$

где P – бинарный предикат, определяющий предусловия возникновения события; f – функция, определяющая факт срабатывания предусловия возникновения события и получения нового состояния объекта в виде нового событийно-иерархического фрейма; $Fr_{событ}$ – вновь сформированный событийно-иерархический фрейм.

Если при подстановке конкретных значений во фрейм-действие предикат принимает значение «истина», то возникает новое событие, описываемое событийно-иерархическим фреймом ($Fr_{\text{событ}}$). Если предикат принимает значение «ложь», то во фрейм-действие подставляются значения до тех пор, пока предикат не примет значение «истина».

Типовая структура событийно-иерархической фреймовой сети, формируемой посредством фреймов-действий и событийно-иерархических фреймов низшего порядка, представлена на рис. 2.

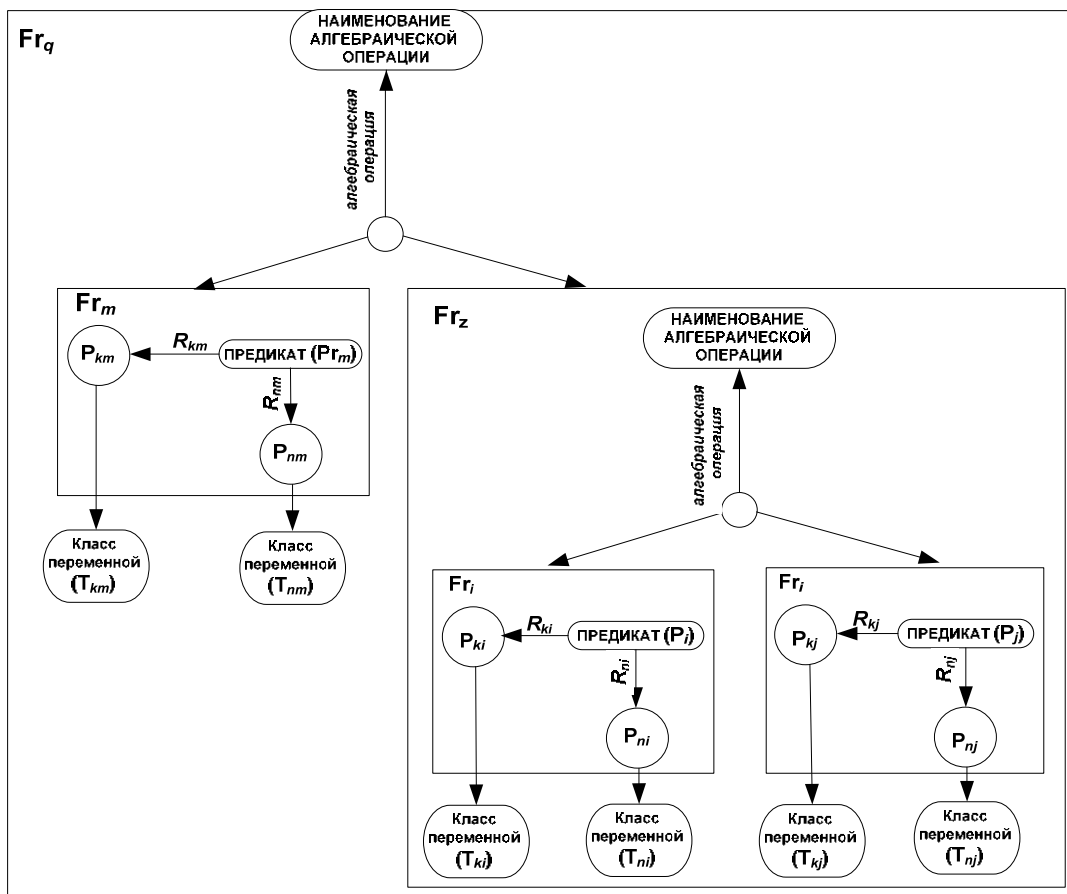


Рис. 2. Типовая событийно-иерархическая фреймовая сеть

Событийно-иерархический фрейм (Fr_q) образуется посредством фрейма-действия (Fr_m) и событийно-иерархического фрейма (Fr_z), которые связываются между собой алгебраическими операциями. В данном случае алгебраические операции определяют логику, условия возникновения события (Fr_q), а также порядок следования событий.

Таким образом, событийно-иерархическая фреймовая сеть представляет собой модель информационных динамических процессов, которая образуется объединением всех фреймов-действий и событийно-иерархических фреймов в единую модель информационно-динамических процессов описываемой предметной области.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ

В работе представлены результаты анализа сравнения базисных функциональных возможностей имеющихся на рынке программных продуктов систем для имитационного моделирования экономических процессов.

В настоящий момент известен целый программ для имитационного моделирования, и их полноценный сравнительный анализ является достаточно объемной задачей. В связи с этим ограничимся только такими системами как Arena, AnyLogic, GPSS/W, является наиболее популярными на отечественном рынке. Данные среды имитационного моделирования позволяют проводить вычисление статистических характеристик (средние, максимальные и минимальные значения, коэффициент использования), исследуемых процессов, оптимизируют параметры и структуру исследуемой системы; осуществлять выявление причин неисправностей в реальной системе и причин их возникновения; создавать компьютерные деловые игры как компоненты системы поддержки принятия решений.

В процессе оценки возможностей пакетов имитационного моделирования были учтены такие базисные функции и критерии, как дружелюбность интерфейса разработки имитационной модели; наличие документации; набор форматов для вывода; количество заранее подготовленных компонентов, позволяющих оценивать дополнительные возможности, дающие возможность квалифицированным специалистам разрабатывать собственные модели; поддержка пакетами, наличие графического отображения; агентное моделирование; поддержка анализа результатов.

Рассмотрение начнем с пакета AnyLogic – это отечественный профессиональный инструмент нового поколения, который предназначен для разработки и исследования имитационных моделей. Разработчик продукта – компания «Экс Джей Текнолоджис» (XJ Technologies), г. Санкт-Петербург. Пакет обладает возможностью использования компоновочных блоков, потоковых диаграмм. Развитая графическая среда моделирования поддерживает проектирование, разработку, документирование модели, выполнение компьютерных экспериментов, оптимизацию параметров относительно некоторого критерия. При разработке модели можно использовать элементы визуальной графики для диаграмм состояний (стейт-чарты), сигналов, событий (таймеры), портов в программе поддерживается синхронное и асинхронное планирование событий; библиотеки активных объектов. AnyLogic поддерживает известные виды моделирования: динамических систем, дискретно-событийного моделирования, системной динамики, агентного моделирования.

Другой популярной средой имитационного моделирования разработки является среда Arena, разработанная компанией Systems Modeling Corporation. Обладает возможностью авторского моделирования, поддерживает различные виды моделирования. Программа Arena снабжена удобным интерфейсом, обладает широкими функциональными возможностями по адаптации к различным предметным областям. Основой технологии моделирования Arena являются язык моделирования SIMAN и анимационная система Cinema Animation. Особенностью среды является то, что она поддерживает агентное моделирование, которое заклю-

чается в том, что при разработке агентной модели, инженер вводит параметры агентов (это могут быть люди, компании, активы, проекты, транспортные средства, города, животные и т.д.), определяет их поведение, помещает их в некую окружающую среду, устанавливает возможные связи, после чего запускает моделирование. Индивидуальное поведение каждого агента образует глобальное поведение моделируемой системы.

Последней рассматриваемой системой была GPSS, разработанная сотрудником фирмы IBM Джеффри Гордоном в 1961 году и с тех пор претерпевшая ряд модификаций и остающаяся популярной в сфере разработчиков имитационных моделей. GPSS относится к событийно-ориентированному языку программирования, также содержит встроенный язык программирования Plus, который позволяет существенно расширить функциональные возможности моделирования и реализовывает сложные алгоритмы обработки информации. GPSS World позволяет увидеть функционирование модели в виде анимационной картинки. Система имеет интерпретатор программного кода, то есть модель работает только в среде GPSS и не может компилироваться в исполняемые файлы.

При сравнительном анализе был также использован неотмеченный выше, такой важный критерий как, стоимость приобретения программного продукта. В соответствии с этим критерием данные программные продукты в порядке увеличения стоимости расположились так: Arena (20 тыс. руб.), AnyLogic (30 тыс. руб.), GPSS World (35 тыс. руб.). Тем не менее, видно, что их стоимости сопоставимы и разница в десять тысяч рублей, на наш взгляд, не может служить серьезным основанием для определения выбора среды при необходимости реализации сложных имитационных проектов.

Проведенный анализ позволяет сделать вывод, что для больших и сложных моделей целесообразно применять GPSS World, важным плюсом которого по сравнению с другими (на фоне примерно одинаковых других значений параметров сравнения) является наличие встроенного языка программирования. Следует понимать, что проведенный анализ содержит определенные элементы субъективизма, отражающие предпочтения автора, занимающегося разработкой систем имитационного моделирования в среде GPSS, поэтому Arena и AnyLogic могут быть использованы разработчиками в тех предметных областях, где их особенности могут иметь первостепенное значение.

С. А. Медовщикова, А. М. Паршин, Д. В. Пащенко

Россия, Пенза, Научно-производственное предприятие «Рубин»

МОДЕЛЬ РАСЧЕТА ПРОИЗВОДСТВЕННЫХ ВОЗМОЖНОСТЕЙ РЕМОНТНО-ВОССТАНОВИТЕЛЬНЫХ ПОДРАЗДЕЛЕНИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Приводятся основные положения и особенности создания модели задачи расчета производственных возможностей ремонтно-восстановительных органов.

Новые формы и способы вооруженной борьбы с внедрением новейших видов вооружения и военной техники (ВВТ) требует пересмотра организационно-

технических мероприятий, обеспечивающих своевременный ремонт ВВТ, быстрое восстановление и возвращение в строй поврежденного ВВТ. Следовательно, вопросы оптимизации планирования ремонта ВВТ в настоящее время имеют особую важность.

Особенностью организации ремонта ВВТ в РВО является разделение РВО на подразделения, специализирующиеся на ремонте определенной группы ВВТ. Таким образом, планирование ремонта ВВТ наиболее целесообразно осуществлять исходя из производственных возможностей каждого подразделения из состава РВО.

В данной работе предлагается методика оптимизации планирования ремонта ВВТ в оборонном комплексе РФ основанная на модели расчета производственных возможностей РВО для каждого подразделения из состава РВО.

Исходными данными для модели являются:

- перечень РВО;
- укомплектованность i -го РВО специалистами (по ВУС);
- обеспеченность i -го РВО средствами ремонта (по группам ВВТ);
- обеспеченность i -го РВО запасными частями, инструментом и принадлежностями (по группам ВВТ).

Структура модели расчета производственных возможностей РВО в виде UML-диаграммы активности представлена на рис. 1.

Модель по расчету производственных возможностей РВО позволяет:

1. Определить коэффициенты укомплектованности специалистами, средствами ремонта в РВО, по которому осуществляется анализ.

2. Рассчитать производственные возможности каждого подразделения из состава выбранного РВО, специализирующиеся на ремонте определенной группы ВВТ по R_q виду ремонта ($q = 1$ для текущего ремонта, $q = 2$ для среднего ремонта).

Для этого с учетом нормативных производственных возможностей каждого подразделения из состава выбранного РВО; минимального значения из значений коэффициентов укомплектованности специалистами и средствами ремонта для выбранного подразделения РВО рассчитываются реальные производственные возможности каждого подразделения (Π_{ij}) из состава выбранного РВО по формуле:

$$\Pi_{ijRq} = \min (K_{inc}; K_{imc}) * \Pi_{ijnRq} ,$$

где K_{inc} – коэффициент укомплектованности i -го РВО специалистами по ремонту j -й группы ВВТ (личным составом, в соответствии с военно-учетной специальностью); K_{imc} – коэффициент укомплектованности i -го РВО средствами по ремонту j -й группы ВВТ; Π_{ijnRq} – нормативные производственные возможности i -го РВО по ремонту j -й группы ВВТ с учетом вида ремонта.

3. Рассчитать производственные возможности всех РВО j -й группы ВВТ по R_q виду ремонта по формуле:

$$\Pi_{РВОjRq} = \sum \Pi_{ijRq}$$

4. Рассчитать производственные возможности каждого подразделения из состава выбранного РВО с учетом имеющегося наличия запасных частей, инструмента и принадлежностей.

5. Исходя из реальных производственных возможностей каждого подразделения из состава РВО, оптимизировать планирование ремонта ВВТ – распределять силы и средства ремонтных подразделений усиливая производственные возможности отдельного РВО на наиболее важных направлениях по ремонту ВВТ.

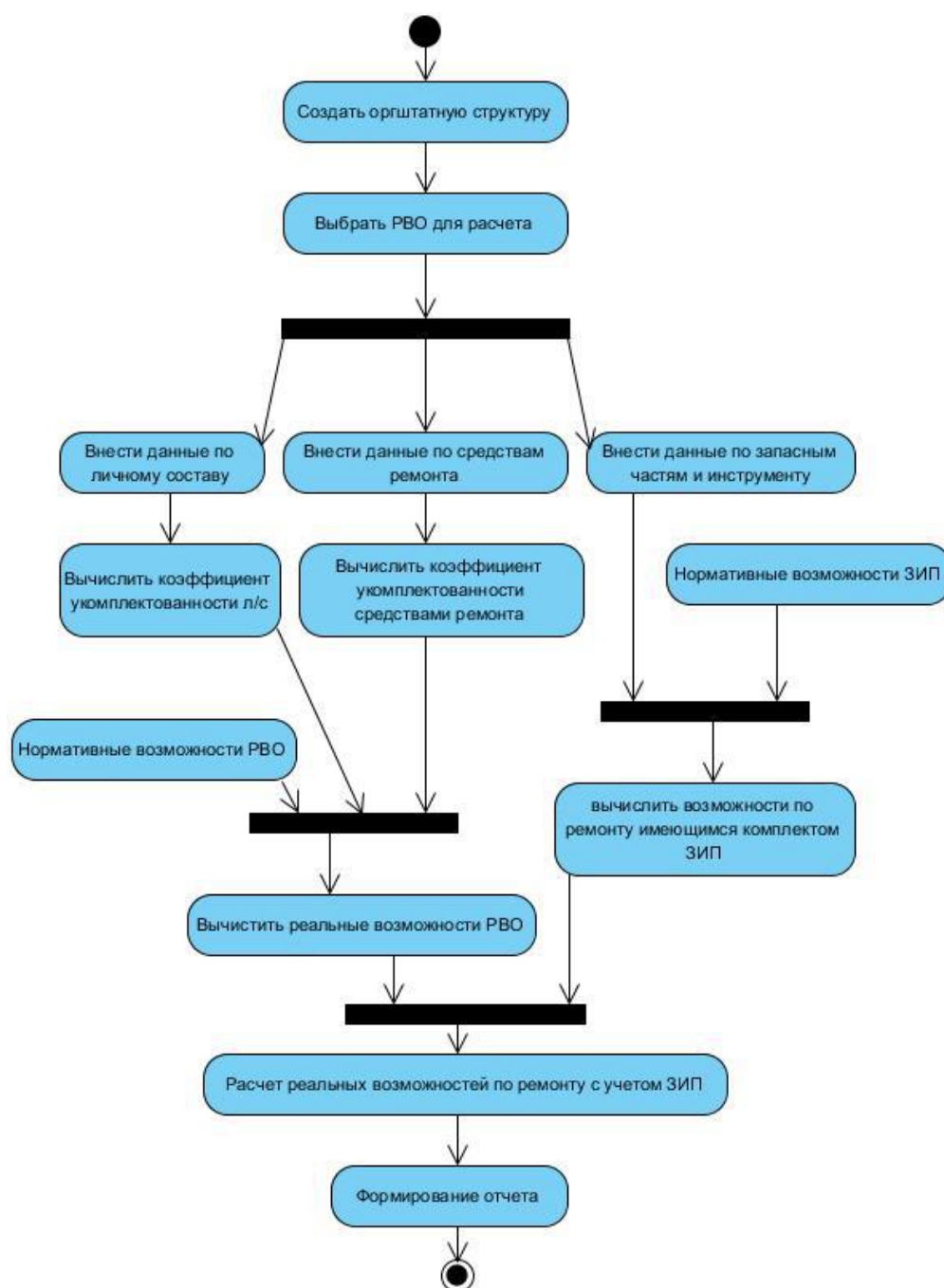


Рис. 1. Структура модели расчета производственных возможностей РВО в виде UML-диаграммы активности

Таким образом, настоящую методику по оптимизации планирования ремонта ВВТ в оборонном комплексе РФ на основе модели расчета производственных возможностей РВО каждого подразделения из состава РВО наиболее целесообразно применить при разработке задач автоматизации системы учета и планирования обеспечения запасными частями, инструментом и принадлежностями.

А. И. Мартышкин

Россия, Пенза, Пензенский государственный технологический университет,
Пензенский государственный университет

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СМО С ПОМОЩЬЮ MATLAB+SIMULINK

Рассматриваются возможности имитационного моделирования систем массового обслуживания средствами программы MATLAB+Simulink. Приводится подробный пример реализации модели и соответствующие графики для исследования полученных характеристик.

Целью работы является применение средств MATLAB + Simulink для выполнения учебно-исследовательских работ по моделированию систем массового обслуживания (СМО).

Согласно поставленной цели моделируемая СМО в общем виде должна представлять собой некий «черный ящик», имеющий вход для поступающих заявок и выход для обслуженных. В состав его должны входить минимум два подблока – канал обслуживания и очередь перед ним. Заявки должны поступать как с выхода очереди на вход канала, так и с выхода канала на вход очереди (по обратной связи). Кроме того, необходимо предусмотреть источник потока заявок с заданными характеристиками (количеством, интенсивностью и пр.), а также приемник обслуженных заявок, в котором будет осуществляться статистическая обработка.

Функциональная схема канала обслуживания приведена на рис. 1,а. Основным элементом данной подсистемы является канал обслуживания. Он принимает заявку и задерживает ее на заданный период времени. Длительность может задаваться как свойство блока, а может быть передана ему как входной сигнал (этот вариант применен в статье). Источником времени задержки является подсистема генератор времени обслуживания (рис. 1,б). Необходимость выделения отдельной подсистемы для генератора обоснована необходимостью сформировать случайный сигнал. Кроме того, в блоке учитывается вес пришедшей заявки. Этот учет осуществляется суммированием сгенерированного времени обслуживания и веса заявки. Входным параметром для генератора является вес заявки, передаваемый как ее атрибут и извлекаемый соответствующим блоком на входе всей подсистемы канала обслуживания.

В исследуемой модели для каждой обслуженной заявки нужно решить вопрос: выйдет ли заявка из системы или она должна возвратиться в очередь и вновь поступить на обслуживание? Для простоты положим, что «решающая» величина имеет равномерный закон распределения и лежит в диапазоне от 0 до 1 – это позволит легко задавать пороговую вероятность возврата заявки в очередь. Именно такой подход реализован в канале обслуживания.

Функциональная схема подсистемы очереди заявок представлена на рис. 1в. Главным элементом здесь является блок очереди из библиотеки SimEvents. Он реализует очередь заявок заданной емкости с дисциплиной обслуживания FIFO. Емкость очереди задается как свойство блока. Дисциплина обслуживания выбрана такой лишь из соображений упрощения модели (можно выбрать LIFO или приоритетную) [2]. Блок принимает на вход заявку, ставит ее в конец очереди, а стоящую в начале очереди заявку отправляет на выход, если он не заблокирован. В каждый момент опроса очередь сообщает свое содержимое, среднюю длину и среднее время ожидания каждой заявки.

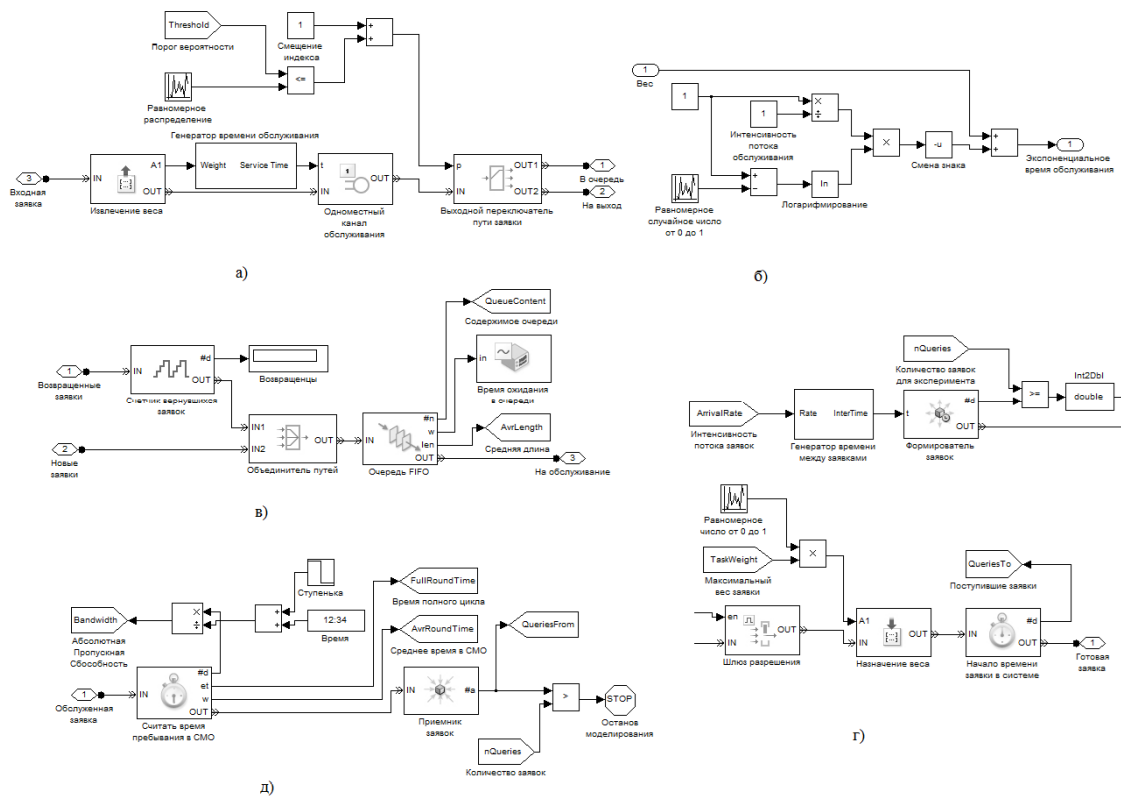


Рис. 1. Схема подсистемы канала обслуживания (а), генератора времени обслуживания (б), подсистемы очереди (в), подсистемы источника (г) и подсистемы приемника заявок (д)

Заявки в очередь могут поступать не только из источника заявок, но и по обратной связи из канала обслуживания. Однако очередь имеет лишь один вход. Решением данной проблемы является блок объединителя путей, который направляет заявки со своих входов на один выход. Настройки блока таковы, что если две заявки придут в блок одновременно, на выход первой поступит заявка, пришедшая на вход IN1.

Функциональная схема подсистемы источника заявок представлена на рис. 1,г. Основным элементом этой подсистемы является Формирователь заявок. Его настройки позволяют указывать интервал между заявками как свойство самого блока, так и через входной сигнал. Чтобы проводить эксперимент на конкретном, количестве поступающих в СМО заявок, здесь предусмотрен шлюз разрешения. Он управляется сигналом, преобразованным из сигнала результата сравнения заданного и фактического количеств заявок. Другими словами, в периоды времени, когда заданное количество заявок уже поступило в систему, но еще не прошло обслуживание, Генератор заявок продолжает работу, однако шлюз разрешения блокирует его выход.

У каждой заявки имеется некоторое свойство. В создаваемой модели таким свойством является вес заявки. Заявка с большим весом обслуживается дольше, чем с меньшим. Другим применением этого параметра может стать распределение приоритетов заявок в соответствии с весами. Чтобы веса у заявок были разными, но подчинялись пользователю модели, предусмотрен простой генератор весов заявок. Он генерирует равномерно распределенное случайное число от 0 до 1, которое умножается на заданный максимальный вес, обеспечивая тем самым случайность этого параметра.

Функциональная схема подсистемы приемника заявок представлена на рис. 1,д. Конечным пунктом назначения заявки является блок библиотеки

SimEvents, обозначенный как приемник заявок. Он поглощает заявку и сообщает в виде сигнала количество пришедших заявок. Это количество сравнивается с заданным перед экспериментом количеством заявок. Их совпадение говорит о том, что все заявки прошли обслуживание. При выполнении этого условия моделирование можно остановить.

Приведем аналитические формулы, используемые при расчете модели [1]. Коэффициент загрузки СМО:

$$\rho = \frac{\lambda}{\mu}, \quad (1)$$

где λ – интенсивность потока входных заявок; μ – интенсивность потока обслуженных заявок.

Средняя длина очереди:

$$L = \frac{\rho^2}{1 - \rho}, \quad (2)$$

Абсолютная пропускная способность системы:

$$A = \lambda(1 - \rho p_0), \quad (3)$$

где p_0 – вероятность отсутствия заявок в системе.

В модели эта величина рассчитывается по формуле:

$$A = \frac{N}{T}, \quad (4)$$

где N – количество обслуженных заявок за время T .

Среднее время нахождения заявки в системе:

$$W_{\text{сист}} = \frac{\rho}{\lambda(1 - \rho)}. \quad (5)$$

Чтобы убедиться в правильности построения модели, нужно начать ее проверку с эксперимента, результаты которого легко определить аналитически. Укажем интенсивность входного потока, равную 1. Максимальный вес заявки установим в 0, исключив его влияние на время обслуживания. Укажем вероятность возврата равной 0, для оценки чистого времени обслуживания и простоя в очереди. Число заявок примем равным 10. Очевидно, что при таких параметрах абсолютная пропускная способность системы должна составить около 1. Среднее время нахождения заявки в системе должно быть небольшим. Очередь не успеет переполниться, а время ожидания в ней будет сопоставимо с интенсивностью обслуживания. Графики результатов моделирования с перечисленными выше параметрами представлены на рис. 2,а.

В результате эксперимента получено, что $A = 1,1$ (заявки/с), $W_{\text{сист}} = 2,182$ с, $L = 1,604$ заявок. По двум верхним графикам рис. 2а можно отметить общую динамику процесса, происходившего в системе. В первые 0,8 с. источник заявок бездействовал; простаивал и канал обслуживания. Сразу после этого источник начал интенсивно генерировать заявки, и все из вновь сгенерированных заявок, кроме первой, заняли места в очереди. Это и обусловило рост их числа в очереди на нижнем правом графике. Интенсивность потока обслуживания в этот период отклонилась в меньшую сторону, но канал продолжал работать, о чем свидетельствует спад числа заявок в очереди в момент времени $\sim 1,8$ с. Другим подтверждением служит равномерность роста числа обслуженных заявок (левый верхний график) в период, когда на входе наблюдался явный перерыв (~ 3 – 5 с).

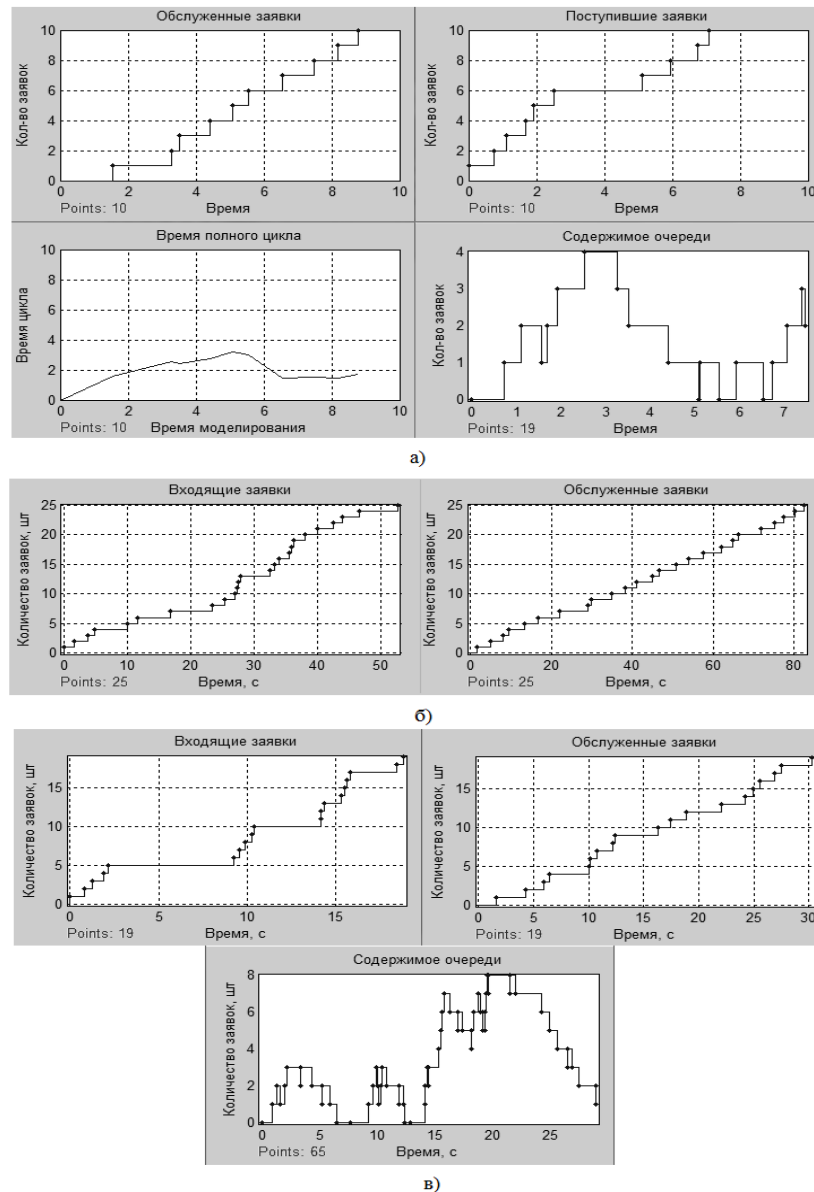


Рис. 2. Графики результатов первого эксперимента (а), моделирование с учетом весов заявок (б), моделирование с учетом обратной связи (в)

Для верификации других возможностей модели зададим вес заявок ненулевым. Создадим поток в 25 заявок с $\lambda = 0,5$ заявок/с. и максимальным весом 4. Введение веса заявок увеличит время обслуживания каждой из них на определенную случайную величину. Результаты моделирования с такими параметрами представлены на рис. 2б. По графикам видно, что последняя заявка вышла из системы примерно на 30 с. позже момента своего поступления (общее число заявок – 25).

В третьем эксперименте задействована вероятность возврата заявки на дообслуживание. Установим максимальный вес заявки равным нулю, а вероятность возврата примем равной 0,5. Модель не различает заявки, уже возвращенные по обратной связи ранее. Каждому экземпляру заявки необходимо присвоить соответствующий признак. В связи с этим, каждая заявка может быть возвращена более одного раза. Зададим входящий поток из 20 заявок с интенсивностью $\lambda = 0,9$ заявок/с. Как и в первом эксперименте, такие параметры должны привести к равномерной нагрузке на канал и общее время обслуживания почти сравнялось бы с периодом поступления заявок. Тем временем, введение обратной связи в

корне меняет представленную картину. Результаты моделирования с указанными параметрами отражены на рис. 2,в.

При анализе представленных на рис. 2,в графиков видно существенное различие в общих временах поступления и обслуживания. Более того, на графике обслуженных заявок видно несколько приостановок, каждая из которых говорит о том, что заявка, прошедшая обслуживание в канале, была вместо приемника направлена снова в очередь на обслуживание. Также об этом говорят скачки числа заявок в очереди в моменты выхода обслуженных заявок из канала обслуживания. Число заходов в очередь на графике содержимого очереди составило 35, в то время как число заявок в эксперименте всего 20. Это говорит о том, что всего по обратной связи в очередь вернулись 15 заявок. Подтверждением этой цифры служат показания подсистемы очереди. Полученные результаты можно считать адекватными.

Программный пакет MATLAB включает в свой состав мощные средства, позволяющие строить и отлаживать модели СМО. Эти средства представлены комплексом Simulink и входящей в его состав библиотекой SimEvents. Блоки библиотеки SimEvents являются гибким и достаточно легким в освоении инструментом для описания моделей СМО. Разработанные в статье модели могут служить средством решения реальных задач.

Список литературы

1. Алиев, Т. И. Основы моделирования дискретных систем / Т. И. Алиев. – СПб. : СПбГУ ИТМО, 2009. – 363 с.
2. Справочные материалы по пакету MATLAB R2011a. – URL: <http://mathworks.com>

Н. С. Белинская

Россия, Томск, Томский политехнический университет

ПРИМЕНЕНИЕ КОМПЬЮТЕРНОЙ МОДЕЛИРУЮЩЕЙ СИСТЕМЫ ДЛЯ ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОГО СООТНОШЕНИЯ ВОДОРОД/СЫРЬЕ В ПРОЦЕССЕ ГИДРОДЕПАРАФИНИЗАЦИИ ДИЗЕЛЬНЫХ ТОПЛИВ

Разработана математическая модель процесса гидродепарафинизации дизельных топлив, реализованная в виде компьютерной моделирующей системы. Проведено исследование влияния ключевого технологического параметра процесса на состав получаемого продукта, такого как мольное соотношение Водород:Сырье (в диапазоне 2,7:1,0–3,7:1,0).

Для эффективного управления процессами нефтепереработки в настоящее время на предприятия широко внедряются новейшие информационные технологии. Для управления процессом каталитической гидродепарафинизации дизельных топлив построена математическая модель с учетом физико-химических закономерностей процесса, а именно реакционной способности углеводородов дизельной фракции, механизма реакций на бифункциональном катализаторе, термодинамических и кинетических закономерностей протекания реакций [1].

Математическая модель процесса гидродепарафинизации реализована в среде программирования Delphi 7 в виде компьютерной моделирующей системы (КМС). Она содержит в себе интерфейс, базу данных, базу знаний и алгоритм решения дифференциальных уравнений математической модели.

Интерфейс, разработанной КМС, показан на рис. 1.

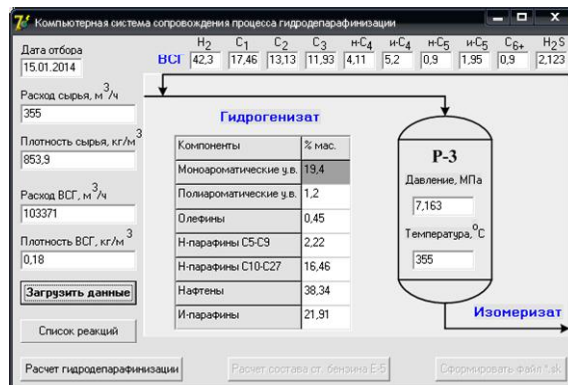


Рис. 1. Интерфейс КМС процесса гидродепарафинизации дизельных топлив

Наибольший вклад в значения низкотемпературных характеристик дизельных топлив вносят концентрации высокомолекулярных н-парафинов и и-парафинов [2].

С использованием разработанной КМС проведены расчеты влияния мольного соотношения Водород:Сырье на конверсию высокомолекулярных н-парафинов и выход и-парафинов в пределах 2,7:1,0–3,7:1,0 моль водорода/моль сырья (рис. 2). При увеличении мольного соотношения Водород:Сырье с 2,7:1,0 до 3,3:1 моль/моль конверсия н-парафинов $C_{10}-C_{27}$ увеличивается на 11 % с 32 % до 43 %. Выход и-парафинов возрастает на 11 % с 12 % до 23 %.

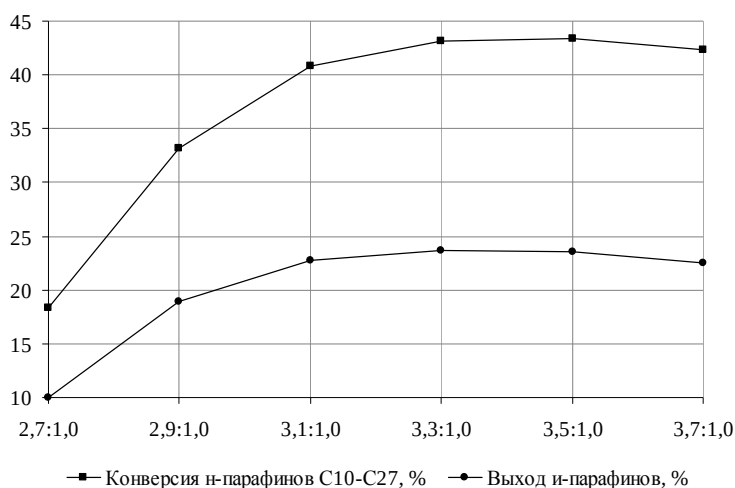


Рис. 2. Исследование влияния мольного соотношения Водород:Сырье на конверсию н-парафинов $C_{10}-C_{27}$ и выход и-парафинов

Увеличение мольного соотношения Водород:Сырье способствует снижению коксообразования на катализаторе, но тормозит целевую реакцию изомеризации, протекающую через стадию дегидрирования н-парафинов. Следовательно, важно

поддерживать расход ВСГ на оптимальном уровне, чтобы обеспечить селективность целевых реакций.

Выявлено, что для получения продукта с требуемыми низкотемпературными характеристиками процесс гидродепарафинизации следует проводить при мольном соотношении Водород:Сырье, не превышающем 3,3:1,0 моль/моль.

Таким образом, разработанная компьютерная моделирующая система позволяет проводить оптимизацию и прогнозирование работы установки каталитической гидродепарафинизации с высокой точностью без значительных затрат на проведение эксперимента и малого количества времени, что делает ее эффективным инструментом повышения эффективности работы промышленных установок и производства в целом.

Список литературы

1. Mathematical model of straight run diesel catalytic hydroisomerization / N. Belinskaya, E. Ivanchina, E. Ivashkina, G. Silko // XVIII International Scientific Symposium in Honour of Academician M. A. Usov: PGON2014 – IOP Conf. Series: Earth and Environmental Science. – 2014. – № 21.
2. Effect of n-paraffins on the low-temperature properties of aviation diesel fuels / V. Ovchinnikova, V. A. Boldinov, E. A. Esipko, and I. S. Prozorova // Chemistry and Technology of Fuels and Oils. – 2005. – Vol. 41 (6). – P. 462–467.

И. Ю. Балашова

Россия, Пенза, Пензенский государственный университет

КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ДЛЯ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ ЭЛЕКТРОННОГО ОБУЧЕНИЯ

Рассмотрена проблема интеграции и анализа знаний в сфере электронного обучения, обусловленная неопределенностью терминологии, отсутствием однозначной интерпретации и обоснованной классификации знаний о предметной области. Предложено решение задачи построения единой модели знаний на основе предметной области. Предложены пути практического применения построенной предметной онтологии. Сформулированы направления дальнейших исследований.

Одним из перспективных путей развития современной системы образования является электронное обучение. Изучение данной сферы активно проводится как в России, так и за рубежом. Вместе с тем исследования [1, 2] позволяют отметить наличие нерешенных проблем в области концептуального моделирования электронного обучения, в частности, неопределенности терминологии, отсутствия однозначной интерпретации и обоснованной классификации знаний о предметной области. Все это негативно влияет на возможность интеграции, совместного использования и анализа накопленных знаний. В связи с отсутствием единого представления знаний предметной области, информационные системы электронного обучения зачастую несовместимы, обладают низкой семантической интеропера-

бельностью, что снижает действенность их применения. Таким образом, актуальной является задача построения динамически развивающейся единой модели знаний о предметной области.

Одним из наиболее эффективных способов решения задачи модельного представления знаний является проведение онтологического анализа предметной области. К числу наиболее разработанных онтологий относятся модели содержания учебных дисциплин, образовательного контента, обучающихся. Однако данные онтологии характеризуются концептуальной локальностью и локальностью в использовании. Их фрагментарный характер не позволяет составить целостное представление об электронном обучении. Для интеграции знаний об электронном обучении в единую модель необходимо объединение существующих онтологий на основе некоторой базисной. Отсутствие базисной онтологии затрудняет процесс объединения существующих онтологий, зачастую несовместимых между собой.

С целью единой интерпретации знаний о предметной области проведен онтологический анализ нормативной базы, регламентирующей сферу электронного обучения. Выделены ключевые классы онтологии, составляющие общий словарь терминов предметной области. Установлена иерархия классов онтологии «Электронное обучение», фрагмент которой представлен на рис. 1.

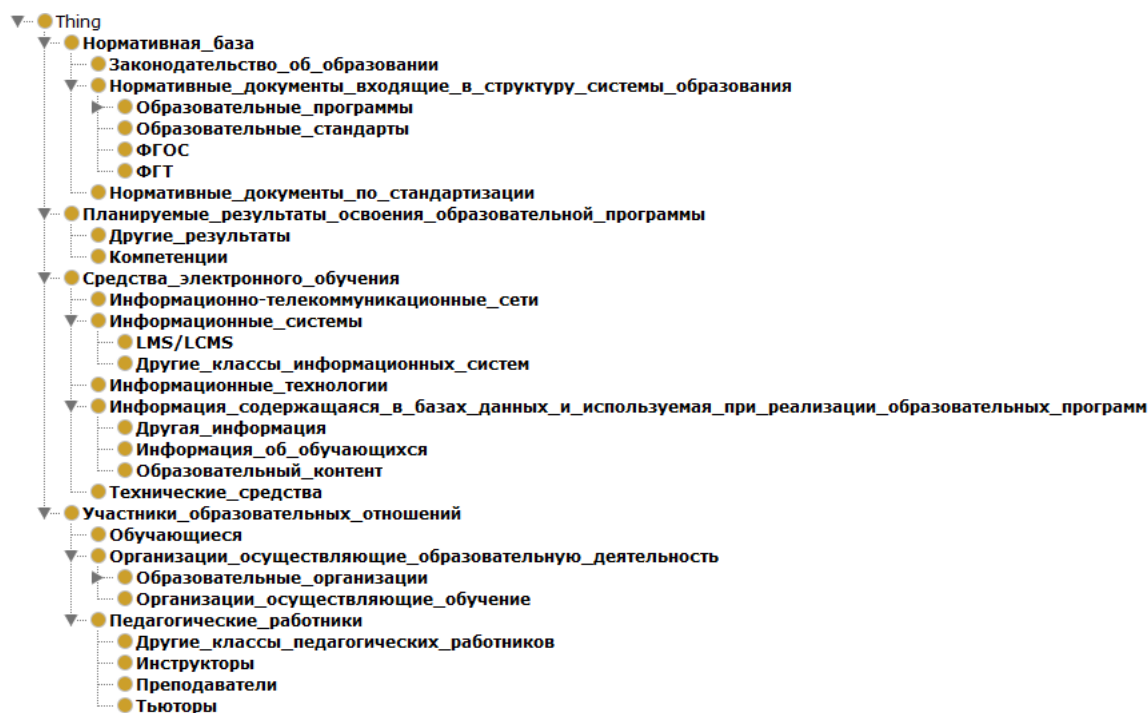


Рис. 1. Фрагмент иерархии классов онтологии «Электронное обучение»

В результате анализа нормативной базы определен набор значимых для настоящего исследования объектных свойств. Описание некоторых из этих свойств представлено в табл. 1.

Свойства $f_1, f_2, \dots, f_8$ обладают обратными свойствами, которые формулируются естественным образом. Ключевыми классами онтологии являются обучающиеся и информационные системы электронного обучения, в частности, системы управления обучением и системы управления образовательным контентом (LMS/LCMS). Системой управления обучением (learning management system, LMS) называется «информационно-управляющая система, предназначенная для выпол-

нения административных процессов и процессов технической поддержки, связанных с электронным обучением» [3].

Таблица 1

Объектные свойства онтологии «Электронное обучение»

Свойство	Домен свойства	Диапазон значений свойства	Описание свойства
f_1	Законодательство об образовании Нормативные документы системы образования	Участники образовательных отношений	Законодательство об образовании и нормативные документы, входящие в структуру системы образования, регулируют деятельность и отношения в сфере образования участников образовательных отношений
f_2	Нормативные документы по стандартизации	Средства электронного обучения	Документы по стандартизации устанавливают для всеобщего и многократного использования общие принципы и характеристики средств электронного обучения
f_3	Организации, осуществляющие образовательную деятельность	Образовательные программы	Организации, осуществляющие образовательную деятельность, реализуют образовательные программы
f_4	Участники образовательных отношений	Средства электронного обучения	Участники образовательных отношений используют средства электронного обучения для организации образовательной деятельности
f_5	LMS/LCMS	Обучающиеся	LMS/LCMS поддерживают образовательную деятельность обучающихся
f_6	LMS/LCMS	Образовательный контент	LMS/LCMS поддерживают управление образовательным контентом
f_7	Обучающиеся	Образовательная программа	Обучающиеся осваивают образовательную программу
f_8	Обучающиеся	Планируемые результаты освоения образовательной программы	Обучающиеся должны овладеть совокупностью планируемых результатов освоения образовательной программы

По сути, LMS представляет собой программную платформу для развертывания электронного обучения в организации и автоматизации управления образовательной деятельностью. Система управления образовательным контентом (learning content management system, LCMS), согласно [3], представляет собой информационно-управляющую систему для создания, хранения, компоновки и/или поставки продуктов и услуг электронного обучения. Следует отметить, что LCMS предоставляет инструментарий для коллективной, многопользовательской работы с контентом. Сейчас границу между LMS и LCMS провести сложно. Большинство производителей систем LCMS включают в них функциональность управления об-

разовательной деятельностью, а ведущие решения категории LMS теперь реализуют и возможности управления образовательным контентом. Широкое распространение в отечественной системе образования получили как коммерческие системы, такие, как WebTutor, eLearning Server, «Прометей», REDCLASS Pro, Blackboard Learn, так и LMS/LCMS с открытым исходным кодом (Moodle, eFront, Sakai, OLAT, ILIAS и др.). В настоящее время ведущие разработчики LMS/LCMS осуществляют сертификационные программы по определению совместимости и интероперабельности между своими продуктами. Тенденцией является сертификация LMS/LCMS на соответствие стандарту SCORM и спецификациям IMS, что позволяет экономически эффективно предоставить доступ к образовательному контенту, созданному в сторонних средствах разработки. Одной из важнейшей функций LMS/LCMS является поддержка разработки образовательного контента. При этом остаются нерешенными проблемы трудоемкости создания образовательного контента и ограниченности применяемых в LMS/LCMS методов оценки его качества. Решение выделенных проблем является перспективным направлением исследований в области электронного обучения.

В заключение необходимо отметить, что онтология «Электронное обучение» описывает иерархию специальных для данной области терминов и отражает значимые логические взаимосвязи между введенными терминами. Построенная онтология является прозрачной, связной. Структура онтологии предусматривает возможность ее расширения без необходимости пересмотра существующих определений. Онтология «Электронное обучение» может использоваться в качестве основы динамически развивающейся единой концептуальной модели знаний о предметной области при проектировании систем электронного обучения.

Список литературы

1. Sangra, A. Building an Inclusive Definition of E-Learning: An Approach to the Conceptual Framework / A. Sangra, D. Vlachopoulos, N. Cabrera // *International Review of Research in Open and Distance Learning*. – 2012. – Vol. 13, no 2. – P. 145–159.
2. Ступин, А. А. Электронное обучение (E-Learning) – проблемы и перспективы исследований / А. А. Ступин, Е. Е. Ступина // *Дистанционное и виртуальное обучение*. – 2012. – № 1. – С. 38–49.
3. ГОСТ Р ИСО/МЭК 2382-36–2011. Информационные технологии. Словарь. – М. : Стандартинформ, 2012. – Ч. 36. Обучение, образование и подготовка. – 12 с.

МОДЕЛИ И ПРЕДСТАВЛЕНИЯ ОНТОЛОГИИ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Онтологии, как наиболее удобный подход к представлению знаний, за последние несколько лет стали использоваться достаточно часто. В данной статье рассматривается применение онтологий в области разработки программного обеспечения, а так же модели и представления данных онтологий.

При проектировании и разработке программного обеспечения требуется решить задачу представления знаний о разрабатываемом проекте программного обеспечения (ПО). Одним из наиболее перспективных подходов к представлению знаний, по мнению многочисленных специалистов, является использование онтологий [1].

Термин «онтология» начал широко применяться в таких сферах Информатики, как Искусственный Интеллект, Агентные системы, Базы данных или Веб-Технологии. Отходя от его оригинального философского значения, в контексте Компьютерных Наук термин онтология применяется для обозначения точной спецификации концептуализации [2]. Такое определение термин получил от Томаса Грубера в 1993 г. В его работе рассматривались различные аспекты взаимодействия интеллектуальных систем между собой и с человеком. Под концептуализацией понимается обобщение понятий и информации, необходимых для описания объектов и решения задач в выбранной области знаний – свойства, отношения, аксиомы, утверждения, ограничения.

Разбиение на модули, распространение, повторное использование и интеграция компонентов программного обеспечения и систем – это проблемы, которые изначально являлись наиболее значимыми при разработке программного обеспечения. Чем больше эти задачи расширяются и автоматизируются, тем важнее определить и использовать онтологии как концептуальную основу таких компонентов. Новые методологии разработки ПО, такие как модельно-ориентированная разработка (MDD) [3], делают модель центральным знанием, базируясь на котором могут быть получены различные реализации, а так же таким образом стимулируют повторное использование моделей. Если системы или компоненты должны обмениваться знаниями, то обмен установить ещё на уровне модели, а не на уровне реализации. В связи с этим возникает осязаемая необходимость определения, внедрения, распространения и использования онтологий [4].

В 2013 г. в журнале International Journal of Computer, Information, Systems and Control Engineering была опубликована статья, в которой предложена онтология проектирования архитектуры ПО, и рассмотрен пример онтологии [5]. Данная онтология основывается на модели IEEE для описания архитектуры и модели Kruchten 4+1 для описания классификации представлений. Для категоризации стилей и представлений был использован стандарт ISO/IEC 42010. Задачи, решаемые с применением предложенной онтологии – классификация и размещение информации о проектировании архитектуры ПО.

В стандарте ISO/IEC 42010 архитектура рассматривается как разделение целой структуры на небольшие части, так, чтобы эти части находились в особых отношениях друг с другом. Такое разделение помогает при глобальной разработке ПО. И хотя отдельные группы пишут ПО, стараясь сделать его легко интегрируе-

мым в работу другими разработчиками, людям бывает сложно взаимодействовать и работать кооперативно из-за существующих различий в методологиях разработки и отличающейся терминологии. Роль онтологии здесь заключается в том, чтобы обеспечить общее понимание для разных терминологий и сделать знания доступными для будущего использования.

На рис. 1 показана модель IEEE для описания архитектуры.

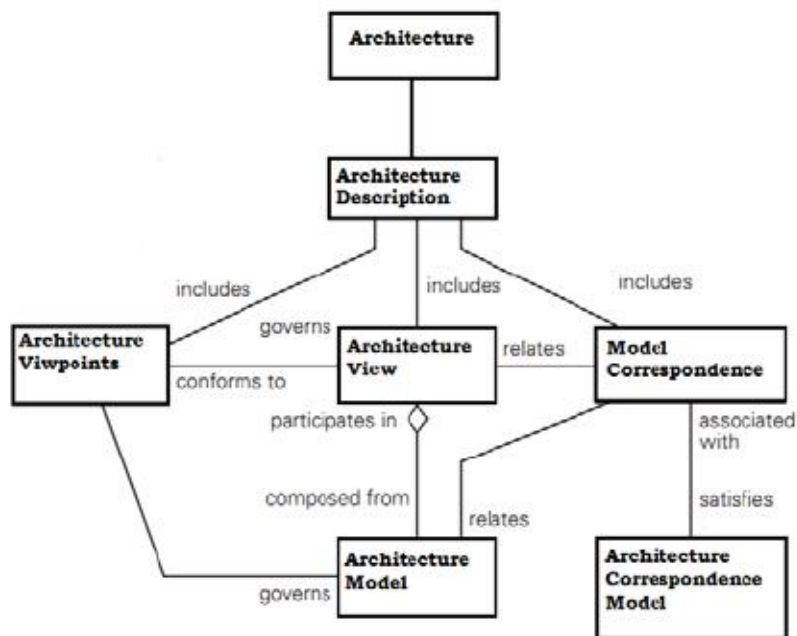


Рис. 1. Модель IEEE для описания архитектуры

Стандарт ISO/IEC 42010 предназначается для описания разрабатываемой структуры архитектуры систем и ПО. В подходе, который используется в стандарте 42010 представления играют центральную роль в документировании архитектуры ПО [6]. Согласно архитектуре стандарта ISO 42010, структура должна содержать: описание стейкхолдеров и проблем, связанных с архитектурой, архитектурные представления, модели и обоснования.

Представления могут быть разделены по категориям на стили: модульный стиль, стиль компонентов и соединений, стиль распределения и гибридный стиль. Модель представлений 4+1, спроектированная Филиппом Крюхтенем, используется, чтобы описать быстро развивающееся ПО. [5] Для этого вводятся пять конкурирующих представлений, каждое из которых предназначено для решения конкретного множества проблем. Структура данной модели приведена на рис. 2.

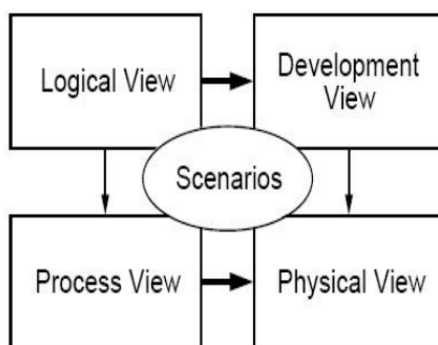


Рис. 2. Структура модели представлений 4+1

Четыре представления, логическое, реализации, процесса и развертывания используются для того, чтобы охватить все возможные решения, принимаемые при проектировании, и пятое, представление сценария, используется для того, чтобы утвердить и проиллюстрировать остальные представления.

Логическое представление связано с функциональным обеспечением системы для конечного пользователя. И для описания дизайна объектной модели, когда используется ООП метод (объектно-ориентированное проектирование). Рациональный подход был использован для описания логического представления с помощью диаграммы классов и шаблонов [7].

Представление реализации связано с описанием системы с точки зрения разработчиков и управляющих разработкой. Так же представление реализации используется для описания статической организации программного обеспечения в среде разработки. В этом представлении принимается во внимание внутреннее требование, связанное с простотой разработки [7].

Представление процесса связано с внедрением, производительностью, масштабируемостью, параллелизмом и т.д. системы, или, другими словами, его описывают такие аспекты проектирования как синхронизация и параллелизм. В представлении реализации учитываются некоторые нефункциональные свойства, такие как производительность, доступность и т.д. При создании представления реализации разработчики разделяют программное обеспечение на несколько частей.

Представление развертывания описывает систему с точки зрения системного инженера и описывает оборудование, на котором будет развертываться программное обеспечение. С этой точки зрения мы принимаем во внимание нефункциональные требования системы: пропускную способность, отказоустойчивость, доступность и надежность и т.д.

Представление сценария описывает последовательность взаимодействия между процессами и между объектами, и оно также называется представлением случаев. Это представление показывает, что элементы четырех других представлений слаженно работают вместе. Это представление является избыточным с учетом других представлений, но играет две жизненно важных роли: помогает разработчику на этапе проектирования, а также утверждает и иллюстрирует проект архитектуры и на бумаге, и на стартовой точке для тестирования архитектурного прототипа.

Список литературы

1. Гаврилова, Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб. : Питер, 2001. – 384 с.
2. Gruber, T. A translation approach to portable ontologies / T. Gruber // Knowledge Acquisition. – № 5(2). – С. 199–220.
3. Денисова, И. Ю. Онтологическое исследование процесса электронного обучения и проектирование средств поддержки / И. Ю. Денисова, П. П. Макарычев // Онтология проектирования. – 2012. – № 4 – С. 61–72.
4. Wolfgang Hesse: Ontologies in the Software Engineering process // EAI 2005 – Tagungsband Workshop on Enterprise Application Integration / ed. by R. Lenz et al. – GITO-Verlag Berlin, 2005.
5. Muhammad Irfan Marwat Sadaqat Jan, Syed Zafar Ali Shah. International Journal of Computer, Information, Systems and Control Engineering Vol:7 No:12, 2013. – Vol. 7. – № 12.
6. Денисова, И. Ю. Математические модели онтологии базы знаний информационной обучающей системы / И. Ю. Денисова, П. П. Макарычев // Онтология проектирования. – 2012 – № 3 (5). – С. 62–78.
7. Kruchten, P. B. The 4 + 1 view model of architecture / P. B. Kruchten // Software, IEEE, 1995. – С. 42–50.

Г. Ф. Убиенных, А. Г. Убиенных

Россия, Пенза, Пензенский государственный университет

ЧИСЛЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНЫХ КРАЕВЫХ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ НА ОСНОВЕ РБФАППРОКСИМАЦИИ

Рассмотрено использование функций радиально-го базиса для численного решения нелинейных краевых задач математической физики методом РБФ – коллокации с использованием ньютоновской итерации. На примере модельной нелинейной краевой задачи исследовано влияние выбора типа радиальной базисной функции и размещения точек коллокации на точность решения.

Дифференциальные уравнения в частных производных (ДУЧП) представляют собой широко применяемый математический аппарат при разработке моделей в самых разных областях науки и техники, таких как гидрогеология, теплофизика, аэродинамика, ядерная физика и многих других.

В последнее время для решения краевых задач теории поля, описываемых с помощью дифференциальных уравнений в частных производных, широкое применение нашли радиальные базисные функции (РБФ) [1]. Значительная часть таких задач носит нелинейный характер. Стандартным инструментом для численного решения нелинейных ДУЧП является ньютоновская итерация.

Целью работы является исследование метода решения нелинейных краевых задач математической физики на основе ньютоновской итерации с использованием различных радиальных базисных функций.

Радиальные базисные функции представляются в форме $\phi(r)$, где $r = \|\mathbf{x} - \mathbf{x}_j\|$ – радиус; $\mathbf{x}_j$ – центр РБФ; $\|\cdot\|$ – некоторая норма (обычно евклидова). РБФ могут иметь параметр формы c , в этом случае они представляются в виде $\phi(r, c)$. Наиболее широко используемыми РБФ являются: $\phi(r) = \sqrt{r^2 + c^2}$ (мультиквадрик), $\phi(r) = \frac{1}{\sqrt{r^2 + c^2}}$ (инверсный мультиквадрик), $\phi(r) = r^2 \log r$ (сплайн-поверхность), $\phi(r) = \frac{1}{1 + r^2}$ (функция Коши), $\phi(r) = e^{-c^2 r^2}$ (гауссиан). Они определены на всей области, где ищется решение, поэтому их относят к группе РБФ с глобальным носителем.

Рассмотрим вкратце суть метода несимметричной коллокации, предложенного Канзой (Kansa) [2]. Аппроксимация решения стационарной линейной краевой задачи, представленной в форме

$$\begin{aligned} L u(\mathbf{x}) &= f(\mathbf{x}), \quad \mathbf{x} \in \Omega \\ B u(\mathbf{x}) &= g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega, \end{aligned} \quad (1)$$

где L, B – дифференциальные операторы в пространстве $\mathbf{x} \in \mathbb{R}^d$; $u(\mathbf{x})$ – переменная поля; $\Omega \subset \mathbb{R}^d$ – ограниченная область решения и $\partial\Omega$ – ее граница, $f, g: \mathbb{R}^d \rightarrow \mathbb{R}$ – заданные функции, ищется в виде линейной комбинации радиальных базисных функций

$$u(\mathbf{x}) = \sum_{i=1}^N \lambda_i \phi_i(\|\mathbf{x} - \mathbf{x}_i\|), \quad (2)$$

где $\phi_i(\|\mathbf{x} - \mathbf{x}_i\|): \mathcal{R}^d \rightarrow \mathcal{R}$ – РБФ с центром в точке $\mathbf{x}_i$, $\|\cdot\|$ – евклидова норма, λ_i , $i = 1, 2, \dots, N$ – неизвестные коэффициенты, которые подлежат определению. На практике центры РБФ выбираются совпадающими с точками множества $\mathbf{X}$, и вектор неизвестных коэффициентов λ вычисляется путем коллокации разрешающих уравнений (3) на множестве $\mathbf{X}$. В результате такая пространственная дискретизация приводит к системе линейных алгебраических уравнений (СЛАУ)

$$\mathbf{A} \lambda = \mathbf{b}, \quad (3)$$

где $\mathbf{A} = \begin{bmatrix} \mathbf{A}_L \\ \mathbf{A}_B \end{bmatrix} \in R^{N \times N}$ – несимметричная коллокационная матрица, $\lambda = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}$. Решая СЛАУ (3), находим коэффициенты λ_i . Подставляя их в выражение (2), находим аппроксимацию решения уравнений (1).

Рассмотрим обобщенное нелинейное эллиптическое ДУЧП, заданное в операторной форме

$$P\mathbf{u} = \mathbf{f}, \quad (4)$$

где P – произвольный нелинейный дифференциальный оператор. Для решения (1) построим ньютоновский итерационный процесс [3], суть которого заключается в следующем. Задаем начальное приближение решения u_0 . На k -ой итерации новое приближение ищется как

$$u_k = u_{k-1} - T_{h_k}(u_{k-1})F(u_{k-1}), \quad k = 1, \dots, n, \quad (5)$$

где $F(\mathbf{u}) = P\mathbf{u} - \mathbf{f}$ – невязка решения; T_{h_k} – численный оператор инверсии, аппроксимирующий $(F')^{-1}$ на сетке с размером шага h_k и зависящий от решения u_{k-1} , полученного на предыдущей итерации. Реализацию T_{h_k} будем проводить путем несимметричной коллокации с использованием РБФ.

Полностью алгоритм аппроксимации ньютоновской итерации можно представить в следующем виде:

1. Внутри области решения Ω и на границе $\partial\Omega$ задаем множество точек коллокации $\mathbf{X} = \{x_i | i=1, \dots, m \in \Omega, x_i | i=m+1, \dots, N \in \partial\Omega\}$, совпадающих с центрами радиальных базисных функций («вычислительную сетку»), и начальное приближение решения u_0 .

Наиболее часто используемыми на практике вычислительными сетками являются:

- равномерная сетка (сетка с постоянным заданным шагом),
- сетка Холтона (сетка со случайным плотным заполнением заданной области),
- сетка Чебышева (сетка с уменьшением размера шага при движении к границе области – сгущающаяся сетка).

2. Для $k = 1, 2, \dots, n$

а) решаем на $\mathbf{X}_k$ линеаризованную задачу

$$L_{u_{k-1}} v = \mathbf{f} - P u_{k-1} ,$$

где $L_{u_{k-1}}$ – линеаризация нелинейного дифференциального оператора P на $(k-1)$ -й итерации; v – поправка к решению.

б) уточняем аппроксимацию решения

$$u_k = u_{k-1} + v .$$

В качестве модельной задачи использовалось нелинейное эллиптическое ДУЧП

$$\begin{aligned} -\mu^2 \nabla^2 \mathbf{u} - \mathbf{u} + \mathbf{u}^3 = \mathbf{f} , \quad \text{в } \Omega = (0,1) \times (0,1) \\ \mathbf{u} = 0 \quad \text{на } \partial\Omega . \end{aligned} \quad (6)$$

Правая часть $\mathbf{f}$ выбирается так, что (6) имеет аналитическое решение в форме

$$u(\xi, \eta) = \psi(\xi) \psi(\eta),$$

где $\psi(t) = 1 + e^{-1/\mu} - e^{t/\mu} - e^{(t-1)/\mu}$.

Здесь (ξ, η) означают декартовы координаты $\mathbf{x} \in \mathbb{R}^2$, параметр μ определяет размер граничных слоев около сторон области Ω . Во всех экспериментах использовалось значение $\mu = 0,1$.

Для модельной задачи линеаризация P задается как

$$L_u v = -\mu^2 \nabla^2 v + (3u^2 - 1)v ,$$

и поэтому уравнение, решаемое на шаге 2а алгоритма, можно записать в виде

$$-\mu^2 \nabla^2 v + (3u_{k-1}^2 - 1)v = \mathbf{f} + \mu^2 \nabla^2 u_{k-1} + u_{k-1} - u_{k-1}^3 .$$

В качестве начального приближения аппроксимации решения использовались значения функции

$$u_0(\xi, \eta) = 16 \xi (1 - \xi) \eta (1 - \eta) .$$

В качестве критерия окончания итерационного процесса использовалось соотношение

$$\ln \left(\frac{e_{k-1}^{\max}}{e_k^{\max}} \right) / \ln 2 < 0.001 ,$$

где $e_{k-1}^{\max}, e_k^{\max}$ – максимальная ошибка аппроксимации решения соответственно на $(k-1)$ -й и k -й итерациях.

Для проведения исследования было разработано приложение в среде Matlab R2013a, обладающее «дружественным» интерфейсом, позволяющим задавать число точек коллокации в виде квадрата какого-либо целого числа, выбирать вычислительную сетку, радиальную базисную функцию и отображаемые графики. Приложение позволяет проводить исследования с использованием трех упомянутых

ранее типов вычислительных сеток (равномерная, Холтона и Чебышева), восьми РБФ различного вида (гауссиан, инверсный мультиквадрик, квадратическая функция Матерна, кубическая функция Матерна, линейная функция, мультиквадрик, сплайн Вендланда С4, сплайн Вендланда С6) и выводить следующие графики: распределение точек коллокации, начальное приближение решения, зависимость средне-квадратической погрешности решения от числа итераций, зависимость нормы невязки решения от числа итераций, поверхность ошибки решения, точное решение.

Исследования показали, что из вычислительных сеток, в узлах которых выбираются точки коллокации, предпочтение по точности решения следует отдать сетке Чебышева. Использование функций Матерна в качестве РБФ позволяет получить более высокую точность решения, чем использование часто применяемых на практике мультиквадрик и гауссианов. Так, при 289 точках коллокации с использованием мультиквадрика была достигнута точность решения, соответствующая среднеквадратической погрешности $1.345482e-003$, в то время как кубическая функция Матерна позволила достичь точности, соответствующей среднеквадратической погрешности $1.803903e-004$.

Разработанная программа легко расширяется путем добавления новых РБФ и настройки под новое ДУЧП. Дальнейшая работа в этом направлении представляется в уточнении вычислительных сеток внутри ньютоновского алгоритма путем адаптивной коллокации.

Список литературы

1. Buhmann, M. D. Radial Basis Function: Theory and Implementations / M. D. Buhmann // Cambridge Monographs on Applied and Computational Mathematics, 2003. – 264 p.
2. Kansa, E. J. Multiquadrics – A scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations / E. J. Kansa // Comput. Math. Appl. – 1990. – No 19 (8/9). – P. 147–161.
3. Fasshauer, G. E. Newton iteration with multiquadrics for the solution of nonlinear PDEs / G. E. Fasshauer // Comput. Math. Appl. – 2002. – № 43. – P. 423–438.

Г. Ф. Убиенных, А. Г. Убиенных

Россия, Пенза, Пензенский государственный университет

МОДЕЛИРОВАНИЕ ЦЕНООБРАЗОВАНИЯ ОПЦИОНОВ НА ОСНОВЕ АППРОКСИМАЦИИ РЕШЕНИЯ УРАВНЕНИЯ БЛЭКА–ШОУЛЗА ФУНКЦИЯМИ РАДИАЛЬНОГО БАЗИСА

Проводится исследование точности результатов моделирования ценообразования европейских опционов на основе аппроксимации решения уравнения Блэка–Шоулза функциями радиального базиса.

Целью работы является исследование бессеточных коллокационных методов решения многомерных дифференциальных уравнений в частных производных на примере моделирования ценообразования опционов с использованием функций радиального базиса (РБФ), а также разработка программного приложения в среде

MATLAB, которое будет реализовывать эти методы для моделирования цены опционов.

Одной из наиболее распространенных моделей ценообразования опционов является модель Блэка–Шоулза [1], представляющая собой многомерное дифференциальное уравнение в частных производных и имеющая следующий вид:

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0, \quad 0 \leq t \leq T \quad (1)$$

В соответствии с этой моделью цена V европейского опциона на акции определяется пятью параметрами $V = V(T, S, K, r, \sigma)$, где T – время, оставшееся до исполнения опциона; S – цена базового актива в настоящий момент; K – цена исполнения опциона; r – безрисковая процентная ставка; σ – волатильность (изменчивость) цены базового актива. Особенностью данной модели является предположение о постоянной волатильности доходности за весь рассматриваемый период действия базового актива.

Поставим краевую задачу с граничными условиями вида

$$V(S, T) = V_g(S), \quad V_a(a, t) = \alpha(t), \quad V_b(b, t) = \beta(t)$$

в усеченной области $\Omega = [a, b] \times [0, T]$. Модель поведения цены акции, использованная Блэком, Шоулзом и Мертоном при выводе уравнения (1), основана на предположении, что изменения цены акции подчиняются логнормальному распределению. Путем экспоненциальной подстановки $S = e^x$ преобразуем уравнение (1) в дифференциальное уравнение с постоянными коэффициентами для области $\Omega = [\log(a), \log(b)] \times [0, T]$. В результате преобразования краевая задача будет описываться уравнением

$$\frac{\partial U}{\partial t} + \frac{1}{2} \sigma^2 \frac{\partial^2 U}{\partial x^2} + \left(r - \frac{1}{2} \sigma^2 \right) \frac{\partial U}{\partial x} - rU = 0 \quad (2)$$

с граничными условиями

$$U(x, T) = g(x), \quad U(\ln(a), t) = \alpha(t), \quad U(\ln(b), t) = \beta(t) \quad (3)$$

Решение краевой задачи (2), (3) будем осуществлять на основе бессеточного коллокационного метода [2], впервые предложенного Канзой (Kansa) [3]. Дискретизируем область Ω набором точек N по оси x и набором точек M по оси t . Таким образом, в области определения функции имеем замкнутое множество $\Xi = (\xi_1, \xi_2, \dots, \xi_N)$. На временном интервале получаем множество $\Theta = (\theta_0, \theta_1, \dots, \theta_M)$, где $\theta_0 = T$, $\theta_M = 0$.

Аппроксимацию цены опциона и ее производных будем искать в виде

$$U(x, t) = \sum_{j=1}^N \lambda_j(t) \phi(\|x - \xi_j\|_2),$$

$$\frac{\partial U}{\partial x} = \sum_{j=1}^N \lambda_j(t) \frac{\partial \phi(\|x - \xi_j\|_2)}{\partial x}, \quad \frac{\partial^2 U}{\partial x^2} = \sum_{j=1}^N \lambda_j(t) \frac{\partial^2 \phi(\|x - \xi_j\|_2)}{\partial x^2},$$

$$\frac{\partial U}{\partial t} = \sum_{j=1}^N \frac{\partial \lambda}{\partial t} \phi(\|\xi_i - \xi_j\|_2)$$

Подставляя эти выражения в уравнение (2) и коллоцируя в точках множества Ξ , получаем систему обыкновенных дифференциальных уравнений (ОДУ) N -го порядка, которую в векторно-матричной форме можно записать в виде

$$\dot{\lambda} = P\lambda \quad (4)$$

Здесь вектор $\dot{\lambda}$ представляет собой $\left\{ \frac{\partial \lambda_j}{\partial t} \right\}_{j=1}^N$, матрица P выражается как

$$P = rI - \frac{1}{2}\sigma^2[\phi]^{-1}[\phi''] - \left(r - \frac{1}{2}\sigma^2 \right) [\phi]^{-1}[\phi'],$$

где I – единичная матрица, $[\phi]$ – коллокационная матрица, $[\phi']$, $[\phi'']$ – первая и вторая производные коллокационной матрицы.

Систему ОДУ (4) можно решать различными численными методами, например, методами Эйлера, Рунге – Кутты. В работе использовался θ -метод, широко применяемый на практике для решения подобных задач. Аппроксимируя производную по времени в системе (4) на основе θ -метода, получаем

$$\lambda^k = \lambda^{k-1} - \Delta t P \left(\theta \lambda^{k-1} + (1-\theta) \lambda^k \right), \quad k = 1, \dots, M,$$

что приводит к системе линейных алгебраических уравнений (СЛАУ)

$$(I + \Delta t (1-\theta)P) \lambda^k = (I - \Delta t \theta P) \lambda^{k-1}, \quad k = 1, \dots, M, \quad (5)$$

где Δt – шаг дискретизации по времени. При $\theta = \frac{1}{2}$ получается известная схема Кранка – Николсона.

Определяя матрицы

$$H = I + \Delta t (1-\theta)P, \quad G = I - \Delta t \theta P,$$

представим СЛАУ (5) в виде

$$H \lambda^k = G \lambda^{k-1}, \quad k = 1, \dots, M$$

В нашем случае имеем нестационарные граничные условия Дирихле вида (3), которые должны выполняться на каждой итерации. На каждом временном шаге должно быть получено новое значение λ^k . Для нахождения вектора коэффициентов λ^0 решаем систему

$$g = [\phi] \lambda^0.$$

Для европейского опциона «колл» граничные условия будут иметь вид

$$V(S, T) = V_g(S) = \max(S - K, 0), \quad V_a(a, t) = \alpha(t) = 0, \\ V_b(b, t) = \beta(t) = S - Ke^{r(T-t)}$$

Для европейского опциона «пут» граничные условия примут вид

$$V(S, T) = V_g(S) = \max(K - S, 0), \quad V_a(a, t) = \alpha(t) = Ke^{r(T-t)}, \quad V_b(b, t) = \beta(t) = 0$$

Рассмотренный алгоритм решения уравнения Блэка–Шоулза может быть реализован на персональном компьютере стандартной архитектуры, а может быть использован как алгоритм обучения радиальной базисной нейронной сети, у которой на каждом временном шаге Δt меняется вектор весовых коэффициентов скрытого слоя λ .

Основной целью эксперимента, проводимого в среде MATLAB R2013a, являлось исследование точности результатов моделирования ценообразования европейских опционов в зависимости от изменения основных влияющих факторов: количества точек коллокации и их расположения, вида РБФ, значения параметра формы РБФ. Эксперимент проводился при следующих значениях параметров модели Блэка–Шоулза: цена исполнения опциона $K = 100$ (условных единиц); безрисковая процентная ставка $r = 5\%$; волатильность $\sigma = 0,3$; срок действия опциона $T = 0,5$ лет.

Оценка погрешности моделирования проводилась по значениям относительной максимальной и среднеквадратической погрешностей аппроксимации, в качестве точного решения использовалось аналитическое решение, полученное по формуле Блэка–Шоулза. Для определения среднеквадратической погрешности были выбраны 1000 равномерно распределенных тестовых точек, не совпадающих с центрами РБФ. Эксперимент проводился при выборе точек коллокации, расположенных в узлах равномерной сетки и сетки Чебышёва (сетка с уменьшением размера шага при движении к границе области – сгущающаяся сетка) с числом узлов 121 и 289 для разных видов РБФ («мультиквадрик», «инверсный мультиквадрик», «гауссиан») с параметром формы $c = 0,01$ и $c = 1,00$. Количество шагов по времени задавалось равным 200, что при сроке действия опциона – полгода примерно соответствует временному шагу – один день.

Эксперименты показали, что при относительно небольшом числе точек коллокации ($N = 121$) относительная максимальная погрешность для европейского опциона «колл» не превышает значения 0,95%, для европейского опциона «пут» – не превышает 0,901%. Относительная среднеквадратическая погрешность для европейского опциона «колл» не превышает значения 0,853%, для европейского опциона «пут» – не превышает 0,541%. При этом, наилучшие по точности результаты получаются в случае использования чебышевской сетки.

При увеличении числа точек коллокации до $N = 289$ относительная максимальная погрешность для европейского опциона «колл» не превышает значения 0,895%, для европейского опциона «пут» – 0,661 %. Относительная среднеквадратическая погрешность для европейского опциона «колл» не превышает значения 0,597%, для европейского опциона «пут» – 0,489 %. Последующее увеличение количества точек коллокации представляется нецелесообразным, так как оно существенно повлияет на точность решения.

Наилучшие результаты по точности для европейского опциона «колл» получены в случае использования радиальной базисной функции «мультиквадрик» со значением параметра формы $c = 0,01$ и количестве точек коллокации $N = 289$: значение максимальной погрешности равно 0,725 %, среднеквадратической – 0,538 %.

Наилучшие результаты по точности для европейского опциона «пут» получены в случае использования радиальной базисной функции «инверсный мульти-

квадрик» со значением параметра формы $c = 0,01$ и количестве точек коллокации $N = 289$: значение максимальной погрешности равно 0,541 %, среднеквадратической – 0,401 %.

Увеличение значения параметра формы c в большинстве случаев повлекло увеличение максимальной и среднеквадратической ошибок.

Полученные результаты говорят о том, что аппроксимация решения уравнения Блэка–Шоулза функциями радиального базиса позволяет получить сравнительно точные результаты при относительно небольших вычислительных затратах.

Литература

1. Black, F. The pricing of options and corporate liabilities / F. Black, M. S. Scholes // Journal Political Economy. – 1973. – № 81. – P. 637–654.
2. Убиенных, Г. Ф. Нейросетевая аппроксимация решений линейных эллиптических дифференциальных уравнений в частных производных на основе методов коллокации / Г. Ф. Убиенных // Нейрокомпьютеры: разработка, применение. – 2007. – № 9. – С. 160–165.
3. Kansa, E. J. Multiquadrics – A scattered data approximation scheme with applications to computational fluid-dynamics. II. Solutions to parabolic, hyperbolic and elliptic partial differential equations / E. J. Kansa // Comput. Math. Appl. – 1990. – No 19 (8/9). – P. 147–161.

Е. А. Малинова, А. Ю. Пучков

Россия, Смоленск, Филиал Московского энергетического института

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РАБОТЫ ЦЕХОВ ПО СБОРКЕ ОКОН НА ПРЕДПРИЯТИИ «МОСКОВСКИЕ ОКНА»

Представлены результаты оптимизации структуры производства предприятия с использованием системы для имитационного моделирования экономических процессов.

Эффективное планирование производства предполагает построение различных моделей, обеспечивающих успешную реализацию управленческих решений, и в конечном итоге, достижение высоких экономических показателей. Задача оптимизации структуры производства является одной из составляющих эффективного менеджмента организации, и ее решения позволяет повысить конкурентоспособность предприятия, расширить производство и клиентскую базу.

Такая задача возникла перед руководством ООО «Московские окна», которое является одним из лидеров на рынке светопрозрачных конструкций с 1992 года и за 22 года завоевали хорошую репутацию. Рассматриваемая компания занимается сборкой, установкой и регулировкой пластиковых окон. Продукция и услуги, предоставляемые этой фирмой, пользуются на рынке большим спросом.

Учитывая отсутствие аналитической модели процесса оптимизации структуры производства, а так же влияние внешней среды, часто имеющей случайные характеристики, было предложено использовать метод имитационного моделирования. Этот метод дает возможность провести неограниченное количество экспериментов с использованием различных параметров. Данный метод исследования

позволяет описывать структуры систем и процессов, не используя сложных математических формул и зависимостей. Кроме того, производственное моделирование помогает обнаружить и устранить проблемы, возникающие на этапе пуска и наладки оборудования, не требуя больших финансовых и временных затрат, тем самым снизив инвестирование в производство.

Моделирование работы цехов предприятия «Московские окна» производилось на языке GPSS, который является популярной средой разработки имитационных моделей. Функционал пакета позволяет осуществлять следующие действия: задавать и проверять последовательность сборки изделий, моделировать операции и потоки материалов для оптимизации производственных процессов, задавать время на выполнение каждой операции, проверять производительность линий и ритмичность производственного процесса, анализировать стоимость продукции и производственных затрат, планировать производственную программу, получать в режиме реального времени информацию о выполнении производственных процессов и отклонений от планов.

Проведя анализ производственного процесса сборки окон, были выявлены статистические параметры моделируемого процесса и построена концептуальная модель работы цехов по обработке заказов и сборке деталей на предприятии «Московские окна». В структуру модели включены очереди, потоки транзактов и обслуживающие устройства. Данная модель построена на основе теории системы массового обслуживания и представлена на рис. 1.

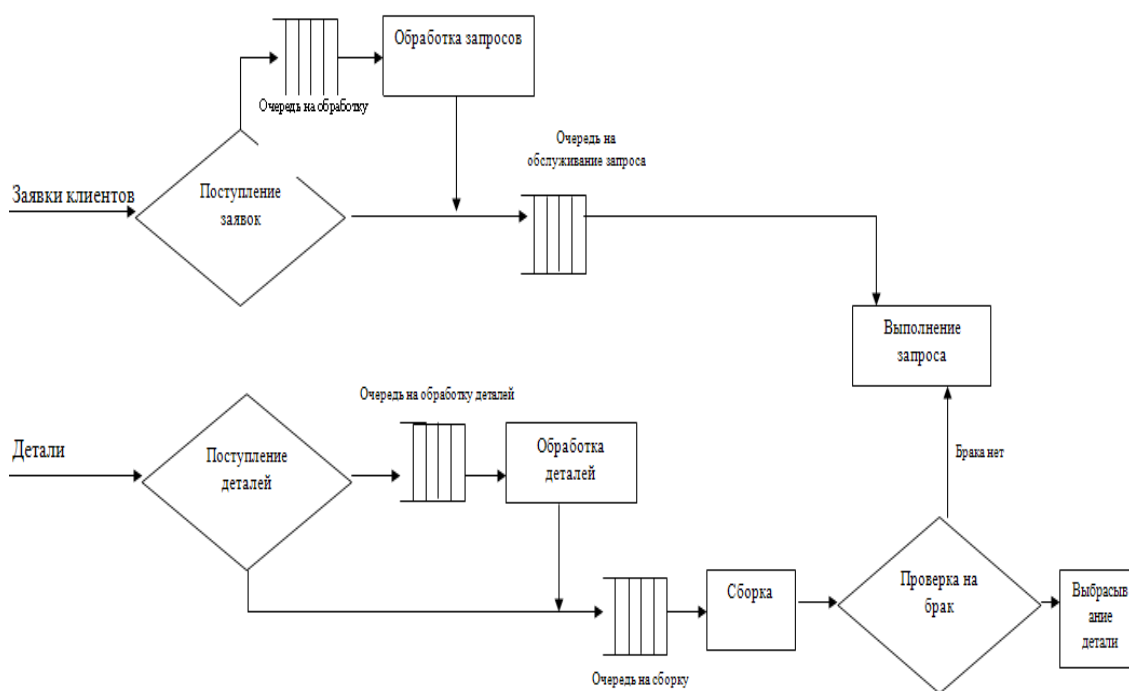


Рис. 1. Концептуальная модель работы цехов предприятия «Московские окна»

В эксперименте варьируемыми параметрами были: количество цехов, длина очередей, время обслуживания запросов, продолжительность процессов обработки и сборки деталей. Критерием оптимизации структуры производства выступила прибыль.

В результате многократных прогонов модели была получена максимальная прибыль для предприятия, которая составила 1 млн 25 тыс. руб. Математическое

ожидание составляет 269 тыс. руб., достигается при наличии 10 рабочих на 1 линии и 8 рабочих на 2 линии. Полученные результаты были представлены менеджменту организации «Московские окна».

Разработанная модель может быть использована для оптимизации производственной структуры других аналогичных предприятий с небольшими модификациями.

В. Г. Захаров

Россия, Пенза, Пензенский государственный университет

МОДЕЛИРОВАНИЕ НА ЭВМ БЕЛОГО ГАУССОВСКОГО ШУМА С ЗАДАНЫМИ МАТЕМАТИЧЕСКИМ ОЖИДАНИЕМ И ДИСПЕРСИЕЙ

Метод моделирования на эвм белого гауссовского шума, основанный на использовании центральной предельной теоремы.

Одним из наиболее распространенных способов формирования гауссовских случайных величин является метод, основанный на использовании центральной предельной теоремы. Данная теорема может быть сформулирована следующим образом: если независимые случайные величины $\xi_1, \xi_2, \dots, \xi_n$ имеют одно и то же распределение и каждая имеет математическое ожидание M_x и дисперсию D_x , то сумма n случайных величин имеет распределение, сколь угодно точно приближающееся к нормальному при $n \rightarrow \infty$ с математическим ожиданием и дисперсией.

$$M_y = n \times M_x; \quad (1)$$

$$D_y = n \times D_x. \quad (2)$$

Практика показывает, что сумма n независимых последовательностей $\{x_i\}$ с равномерным распределением образует последовательность $\{y_i\}$ с распределением, достаточно близким к гауссовскому уже при $n \geq 8$. При этом, если в качестве $\{x_i\}$ фигурирует вырабатываемая стандартной функцией *rand* последовательность с диапазоном распределения $0 \div 1$, математическим ожиданием $M_x = 1/2$ и дисперсией $D_x = 1/12$, то математическое ожидание и дисперсия полученной таким образом гауссовской последовательности в соответствии с (1) и (2) будет определяться как

$$M_y = n/2; \quad D_y = n/12. \quad (3)$$

Поскольку значение n определяет степень близости закона распределения к нормальному, причем $n_{min} \geq 8$, задавать значения M_y и D_y с помощью формул (3) соответствующим выбором n не всегда удобно, а иногда и невозможно.

Попытаемся вывести алгоритм для формирования гауссовской случайной последовательности с заданными математическим ожиданием OM и дисперсией D , не зависящими от n . Для этого в качестве исходных рассмотрим последовательности, равномерно распределенные на некотором интервале $0 \div B$ т.е. имеющие плотность распределения

$$w(x) = \begin{cases} \frac{1}{B}, & 0 \leq x \leq B; \\ 0, & x < 0; x > B. \end{cases}$$

Математическое ожидание и дисперсия такой последовательности будут определяться как

$$M_x = \int_0^B x \times \frac{1}{B} dx = \frac{B}{2}; \quad (4)$$

$$D_x = \int_0^B (x - M_x)^2 \times \frac{1}{B} dx = \frac{B^2}{12}.$$

Если суммировать n подобных последовательностей, то в соответствии с (2) дисперсия результата будет равна

$$D_y = nD_x = \frac{nB^2}{12}. \quad (5)$$

Выражение (5) устанавливает связь между задаваемой дисперсией и величиной B , изменяя значение которой, можно получить требуемое $D[Y]$ при любом n . Из соотношения (5) следует, что

$$B = \sqrt{\frac{12 \times D_y}{n}}. \quad (6)$$

Таким образом, гауссовскую последовательность с дисперсией $D = D_y$ можно сформировать в результате суммирования n независимых случайных последовательностей, равномерно распределенных на интервале $0 \div B$, где B определяется формулой (6). При этом математическое ожидание такой последовательности M_y с учетом (1) и (4) будет равно

$$M_y = \frac{n \times B}{2}$$

или с учетом (6)

$$M_y = \sqrt{3 \times D \times n}. \quad (7)$$

Алгоритм преобразования последовательности вырабатываемых стандартной функцией *rand* случайных чисел $\{x_i\}$ с равномерным распределением на интервале $0 \div 1$ в нужную нам последовательность $\{x_{1i}\}$ с таким же распределением на интервале $[0, B]$ будет описываться выражением

$$x_{1i} = B \times x_i. \quad (8)$$

В результате суммирования n отсчетов x_{1i} в соответствии с изложенным выше получим один отсчет y_{1i} гауссовской последовательности

$$y_{1i} = \sum_{i=1}^n x_{1i},$$

или с учетом (8)

$$y_{1i} = B \times \sum_{i=1}^n x_i.$$

Если в этом алгоритме значения B выбрать в соответствии с (6), т.е. если положить

$$y_{1i} = \sqrt{\frac{12 \times D}{n}} \times \sum_{i=1}^n x_i, \quad (9)$$

то последовательность $\{y_{1i}\}$ будет иметь дисперсию D и математическое ожидание M_y , определяемое формулой (7).

Для того, чтобы получить отсчеты заданной гауссовской последовательности с дисперсией D и математическим ожиданием OM необходимо центрировать последовательность $\{y_{1i}\}$, а затем изменить значения каждой ее реализации на величину OM . При этом алгоритм (9) преобразуется к окончательному виду:

$$y_{1i} = \sqrt{\frac{12 \times D}{n}} \times \sum_{i=1}^n x_i - \sqrt{3 \times D \times n} + OM.$$

Данное выражение позволяет решить поставленную задачу – сформировать последовательность гауссовских случайных отсчетов сигнала $\{y_i\}$ заданными математическим ожиданием OM и дисперсией D , на основе вырабатываемых стандартной функцией *rand* последовательности случайных чисел $\{x_i\}$. Отсчеты сформированной таким образом последовательности $\{y_i\}$ являются независимыми, а следовательно, и некоррелированными. Это дает основание полагать, что отсчеты $\{y_i\}$ представляют собой модель гауссовского белого шума.

М. В. Кононова

Россия, Оренбург, Оренбургский государственный институт менеджмента

ВЕРОЯТНОСТНАЯ АВТОМАТНАЯ МОДЕЛЬ ПЕРЕХОДОВ СОСТОЯНИЙ МЕДИЦИНСКОГО ОБОРУДОВАНИЯ В ПРОЦЕССЕ РЕМОНТА

Резкое увеличение объема медицинской техники обусловило ряд проблем, одной из которых является дисбаланс между требованиями к готовности оборудования и возможностями по его техническому обслуживанию и ремонту. В статье рассматриваются вопросы разработки модели состояний медицинского оборудования при выполнении операций восстановления работоспособности. Предлагается модель состояний медицинского оборудования на основе теории автоматов.

Одной из характерных особенностей современного этапа становления отечественного здравоохранения является рост сложности применяемого дорогостоящего оборудования медицинского назначения для проведения ЛДП в медицинских организациях. По оценкам экспертов Минпромторга РФ, объем поставок медицинской техники в лечебные учреждения за последние десять лет увеличился в 6 раз, причем доля импорта составляет более 60 % и имеет тенденцию к росту. Это привело к двойственному эффекту для государства. С одной стороны, лечебно-профилактические учреждения стали получать высокотехнологичное импортное оборудование с возможностью реализации уникальных лечебных и диагностических манипуляций, но с другой стороны, зачастую наблюдается экономическая неэффективность использования зарубежной техники из-за ее высокой стоимости и низкой окупаемости вследствие неправильной организации потоков больных или простоя; растет зависимость от иностранных компаний в дорогом техническом обслуживании, необходимых расходных материалах и комплектующих, обучении российских специалистов [2].

Поддержание медицинского оборудования в работоспособном состоянии обеспечивает системы технического обслуживания и ремонта (СТОиР), которая предназначена для обеспечения постоянной технической готовности лечебно-диагностического оборудования (ЛДО).

Как указывается в Проекте № 98055635-2ФЗ «Об Обращении Медицинских Изделий» «...Медицинские организации, осуществляющие применение или эксплуатацию медицинских изделий, должны вести учетно-отчетную документацию по техническому обслуживанию медицинских изделий...», а также «...Техническое обслуживание и ремонт медицинского изделия вправе осуществлять: а) производитель, по производимым им медицинским изделиям или уполномоченный представитель производителя; б) лицо, вступившее в саморегулируемую организацию в сфере монтажа, наладки, ремонта и технического обслуживания медицинских изделий; в) специализированная техническая служба медицинского учреждения для собственных нужд». Однако, как показывает анализ современного состояния проблемы организации и проведения работ по ТО и ремонту медицинского оборудования в медицинских организациях и органах управления здравоохранением данному вопросу уделяется недостаточное внимание. В связи с этим наблюдается не рациональное использование медицинской техники в лечебно-диагностическом процессе.

Таким образом, создание СТОиР и обеспечение постоянной технической готовности лечебно-диагностического оборудования к немедленному применению в медицинской технологии является актуальной научно-технической задачей. С практической точки зрения решение этой задачи достигается путем проведения мероприятий по контролю технического состояния, техническому обслуживанию и ремонту медицинской техники. При этом, ремонт и обслуживание медицинского оборудования (МедОб) – это практически всегда сверхсрочная задача, так как отремонтированную технику с нетерпением ждут и пациенты, и персонал.

Технология ремонта медицинской техники – это совокупность приемов и способов восстановления работоспособности (исправности) медицинского оборудования (элемента оборудования), осуществляемых в различных ремонтных органах, а также операции определения состояния МедОб и его составных частей, принятия решения на их восстановление, локализации неисправности до элемента замены, обращения к ЗИП, замены неисправного изделия и контроля работоспособности МедОб. Технология ремонта определяется соответствующим технологическим процессом (ТПР)[1].

В числе наиболее приемлемых способов исследования СТОиР медицинского оборудования выделяется моделирование. При наличии модели возникает целостное понимание работы, какой либо системы, влияния на нее различных факторов, что ведет к возможности управлять рядом из них, а, следовательно, добиваться максимального выполнения поставленных задач при оптимальном использовании ресурсов. Модель обладает большими возможностями прогнозирования в отличие от традиционных методов анализа деятельности, проводимого непосредственно ремонтными органами.

В качестве средства моделирования предлагается рассматривать теорию конечных автоматов. Различают детерминированные и недетерминированные конечные автоматы. Детерминированным конечным автоматом называют следующую пятерку[3]:

$$A = (V, S, P, p_0, F),$$

где $V = \{a_1, \dots, a_m\}$ – входной алфавит (конечное множество символов); $S = \{s_0, s_1, \dots, s_{n-1}\}$ – алфавит состояний автомата (конечное множество символов);

P – функция, отображающая множество $S \times V$ в S и называемая функцией переходов; $p_0 \in S$ – начальное состояние автомата; $F \subseteq S$ – множество состояний, называемых заключительными.

На содержательном уровне функционирование конечного автомата можно представить следующим образом. Имеется бесконечная лента с ячейками, в каждой из которых может находиться один символ из V . На ленте находится цепочка символов $\alpha \in V^*$. Ячейки слева и справа от цепочки не заполнены. Имеется некоторое конечное управляющее устройство с читающей головкой, которое может последовательно считывать символы с ленты, передвигаясь слева направо. При этом устройство может находиться в каком-либо одном состоянии из S . Каждый раз, переходя к новой ячейке, устройство переходит к новому состоянию в соответствии с функцией P .

Недетерминированный конечный автомат есть пятерка вида:

$$A = (V, S, P, p_0, F),$$

где $V = \{a_1, \dots, a_m\}$ – входной алфавит; $S = \{s_0, s_1, \dots, s_{n-1}\}$ – алфавит состояний автомата; P – функция переходов, отображающая множество $S \times V$ в множество подмножеств S ; $p_0 \in S$ – начальное состояние автомата; $F \subseteq S$ – множество заключительных состояний.

Отличие недетерминированного конечного автомата от детерминированного заключается в том, что значениями функции переходов являются не состояния, а множество состояний (или, в терминах команд, возможны различные команды с одинаковыми левыми частями). Это соответствует тому факту, что в диаграмме состояний из одной вершины может исходить несколько дуг с одинаковой пометкой [3].

Язык, допускаемый недетерминированным конечным автоматом A , это:

$$L(A) = \{\alpha \subseteq S_f \in P(p_0, \alpha) \wedge S_f \in F\}.$$

Исходя из стоящей задачи моделирования процесса ремонта МедОб, недетерминированный конечный автомат допускает некий алфавит, сопоставимый с процессом ремонта.

На рис. 1 показана модель состояний конечного недетерминированного автомата соответствующего состояниям ремонтируемого изделия.

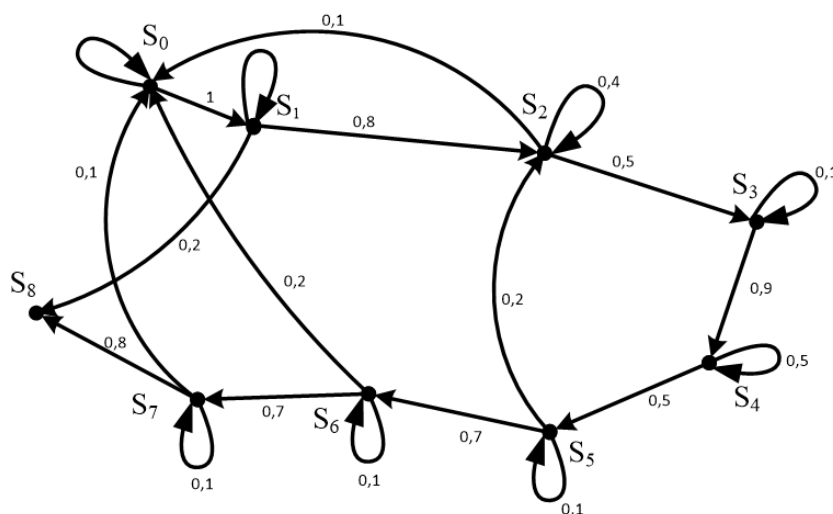


Рис. 1. Модель переходов состояний

Матрица связей S конечного недетерминированного автомата показывает процесс переходов состояний МедОб при ремонте и имеет следующий вид:

$$S = \begin{pmatrix} S_{01}S_{02} & \dots & S_{08} \\ S_{11}S_{12} & \dots & S_{18} \\ & \dots & \\ & \dots & \\ S_{80}S_{81} & \dots & S_{88} \end{pmatrix}$$

Матрица вероятностей свершения переходов конечного недетерминированного автомата имеет вид:

$$P = \begin{pmatrix} P_{01}P_{02} & \dots & P_{08} \\ P_{11}P_{12} & \dots & P_{18} \\ & \dots & \\ & \dots & \\ P_{80}P_{81} & \dots & P_{88} \end{pmatrix}$$

Модель процесса ремонта медицинского оборудования описанная в терминах конечного недетерминированного автомата может естественным образом интерпретироваться сетью Петри, которая имитирует технологию восстановления изделия в целом. В дальнейшем становится возможным проведение анализа технологии ремонта и синтез рациональной структуры СТОиР.

Список литературы

1. Горбачев, Д. В. Видение проекта системы технического обслуживания и ремонта лечебно-диагностического оборудования / Д. В. Горбачев, М. В. Кононова // Научная дискуссия: вопросы юриспруденции, филологии, социологии, политологии, философии, педагогики, психологии, истории, математики, медицины, искусства и архитектуры : материалы Междунар. науч.-практ. конф., 14 декабря 2013. – М., 2013. – С. 278–285.
2. Индилов, Э. В. Формирование стратегии позиционирования изделий медицинского назначения : автореф. дис. ... канд. экон. наук / Индилов Э. В. – М., 2011.
3. Кузин, Л. Т. Основы кибернетики : в 2 т. : учеб. пособие для вузов / Л. Т. Кузин. – М. : Энергия, 1979. – Т. 2. Основы кибернетических моделей. – 584 с.

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Н. П. Вашкевич, В. И. Волчихин, Р. А. Бикташев

Россия, Пенза, Пензенский государственный университет

ВЕРИФИКАЦИЯ СИСТЕМЫ КАНОНИЧЕСКИХ УРАВНЕНИЙ ДЛЯ Q-СОБЫТИЙ, ОПРЕДЕЛЯЮЩИХ СОСТОЯНИЯ ДЕТЕРМИНИРОВАННОГО АВТОМАТА ИСПОЛЬЗУЕМОГО ДЛЯ СТРУКТУРНОЙ РЕАЛИЗАЦИИ АЛГОРИТМА УПРАВЛЕНИЯ ПРОЦЕССАМИ И РЕСУРСАМИ

Предлагается методика контроля правильности структурной реализации систем управления процессами и ресурсами, для варианта построения структурной схемы, основанной на кодировании состояний исходного управляющего автомата.

При построении структурных схем, реализующих алгоритмы управления, обычно используют различные эквивалентные преобразования таких алгоритмов. В процессе таких преобразований могут быть допущены различные ошибки. В связи с этим является актуальной задача контроля правильности формального аналитического описания управляющего алгоритма, полученного на заключительном этапе структурного синтеза, когда управляющий алгоритм будет представлен в стандартной аналитической форме в виде систем канонических уравнений (СКУ) для Q-событий, где Q-события – это события, представляющие переходы в элементах памяти устройства управления. Далее необходимо отметить, что при проектировании управляющих систем уделяется также большое внимание верификации управляющих алгоритмов, представленных в виде программ, когда для их верификации используются автоматные модели [1]. При этом имеется в виду, что контроль правильности аппаратного и программного обеспечения особенно важен, когда он используется в системах промышленной автоматики и в информационных системах управления, работающих в реальном масштабе времени [1].

Рассматриваемая система уравнений для Q-событий представляется в виде системы рекуррентных канонических уравнений, определяемых по результатам кодирования состояний управляющего алгоритма для его структурной реализации. В данном случае имеется в виду, что исходная система управления представляется в виде рекуррентных канонических уравнений функций переходов алгоритма управления в терминах внутренних состояний (a_k) управляющего устройства. Обозначим такую систему уравнений как СКУ детерминированного автомата (ДА) для a_k – событий, которая будет иметь вид:

$$a_k^{[Y_k]}(t+1) = f_k(a_0, a_1, \dots, a_k, \dots, a_n; x_0, x_1, \dots, x_L), \quad (1)$$

где $k = \overline{0, n}$, $a_0(0) = 1$; f_k – булева функция от входных сигналов ($x_0, \dots, x_L$) и полных a -событий (состояний), реализуемых в алгоритме; $[Y_k]$ – множество выходных сигналов, отмечающих состояние a_k (для автомата Мура); a_k – внутреннее состояние управляющего устройства, называемого полным a -событием, так как оно представляется в виде кода из частных совместимых событий, т.е. событий, которые могут одновременно существовать в алгоритме управления.

При этом следует отметить, что система (1) может быть получена в результате выполнения операции детерминации исходной системы управления, если она была представлена недетерминированным автоматом.

Далее отметим, что в результате кодирования внутренних состояний исходного управляющего алгоритма система уравнений для Q -событий представляется в виде системы рекуррентных канонических уравнений (СКУ), которая будет недетерминированной (НД) относительно Q -событий. Такую систему уравнений можно представить следующим образом (2):

$$Q_m(t+1) = f_m(Q_0, Q_1, \dots, Q_R, x_0, x_1, \dots, x_L), \quad (\forall m), m=1, R, \quad (2)$$

где f_m – булева функция от входных сигналов и Q -событий, определяющих функции возбуждения элементов памяти устройства управления.

Данная система (2) будет состоять из R уравнений, если любое состояние a_k исходного управляющего алгоритма представляется конъюнкцией из R букв:

$$a_k = \tilde{Q}_1 \tilde{Q}_2 \dots \tilde{Q}_m \dots \tilde{Q}_R, \quad (3)$$

где R – длина кода состояния, символ $\sim$ означает, что переменная Q_m может быть взята как с отрицанием, так и без отрицания.

Для решения задачи рассматриваемой верификации (контроля правильности системы (2)) воспользуемся тем обстоятельством, что операции детерминизации и кодирования управляющего алгоритма являются обратными друг другу. В связи с этим процедура рассматриваемого контроля может быть организована на основе выполнения операции детерминизации НД СКУ для Q -событий с последующей минимизацией системы уравнений для исходных событий (состояний), представляющих управляющий алгоритм. Учитывая эти обстоятельства, отметим, что если после детерминизации НД СКУ для Q -событий получим СКУ, которая будет соответствовать исходной СКУ до ее кодирования, то это будет свидетельствовать о правильности проверяемой СКУ для Q -событий (2).

Далее отметим, что для выполнения операции детерминизации НД СКУ для Q -событий с целью ее контроля необходимо классический алгоритм детерминизации, рассмотренный в [2], дополнить пунктами, вытекающими из специфики формирования внутренних состояний автомата, представляемых сочетаниями (конъюнкцией) Q -событий и возможным отсутствием полноты входных сигналов, вызывающих переходы в Q -событиях. Это связано с тем, что в кодах состояний автомата некоторые переменные (Q -события) могут быть взяты с отрицанием. Таким образом, в алгоритм детерминизации [2] необходимо ввести следующие дополнения и разъяснения:

1) Если исходная система уравнений для Q -событий состоит из R уравнений, то любое состояние автомата, которое формируется в результате детерминизации такой системы, должно быть представлено сочетанием (конъюнкцией) из R букв (3). Отсюда следует, что если на очередном шаге алгоритма детерминизации будут выводимы только некоторые из Q -событий, определяющие код состояния, то сочетание этих событий необходимо дополнить отрицаниями тех событий, которые не вошли в полученное сочетание.

2) В том случае, если на каком-то шаге алгоритма детерминизации не будет выводимо ни одного из Q -событий, то получим код состояния, который будет соответствовать конъюнкции отрицаний от всех Q -событий. Такое состояние в алгоритме детерминизации, представленном в [2], было названо пустым состоянием. В

данном случае это состояние не будет пустым, так как ему будет соответствовать код $\overline{Q_1} \dots \overline{Q_R}$, принятый при кодировании некоторого вполне определенного состояния автомата.

3) Начало работы алгоритма детерминизации должно определяться начальным состоянием автомата, поэтому его код должен быть зафиксирован в первой строке, формируемой в процессе детерминизации прямой таблицы переходов контролируемого автомата.

4) Если при определении сочетаний частных входных сигналов на каком-либо переходе их совокупность на данном шаге алгоритма детерминизации не удовлетворяет условиям полноты, то эта совокупность входных сигналов должна быть дополнена частными входными сигналами, которые принято называть запрещенными. Эти сигналы определяются путем взятия отрицания от дизъюнкций действующих частных входных сигналов на этом переходе. Такие сигналы обычно помечаются звездочкой. Следует иметь в виду, что для дополнительных входных сигналов, введенных на переходе, соответствующие им состояния автомата не будут неопределенными, как это показано в [2], т.к. выполняемые операции над СКУ для Q-событий относятся к реальному, вполне определенному автомату.

Рассмотрим на простом примере методику детерминизации СКУ для Q-событий, используемую для верификации таких СКУ.

Пусть функции переходов исходного управляющего автомата представлены следующей системой рекуррентных канонических уравнений:

$$\begin{aligned} a_0(t+1) &= x_0, \\ a_1(t+1) &= \overline{a_0 x_1} \vee \overline{a_1 x_2 x_3}, \\ a_2(t+1) &= a_4, \\ a_3(t+1) &= \overline{a_0 x_1 x_2}, \\ a_4(t+1) &= a_0 x_1 x_2 \vee a_1 x_2 x_3 \vee a_2 x_4, \\ a_5(t+1) &= a_3 \vee \overline{a_1 x_2} \vee \overline{a_2 x_4}, \end{aligned} \quad (4)$$

где в правой части уравнений (4) и им подобным время t для простоты записи будет опущено.

Выполняя произвольное кодирование состояний исходного автомата, получим следующие кодовые группы:

$$a_0 = \overline{Q_1} Q_2 Q_3, a_1 = \overline{Q_1} Q_2 \overline{Q_3}, a_2 = Q_1 \overline{Q_2} Q_3, a_3 = Q_1 \overline{Q_2} \overline{Q_3}, a_4 = \overline{Q_1} \overline{Q_2} \overline{Q_3}, a_5 = \overline{Q_1} \overline{Q_2} Q_3.$$

Учитывая принятое кодирование, получим следующую СКУ для Q-событий, представляющую функции возбуждения элементов памяти (5):

$$\begin{aligned} Q_1(t+1) &= (a_2 \vee a_3)(t+1) = a_4 \vee \overline{a_0 x_1 x_2}, \\ Q_2(t+1) &= (a_0 \vee a_1)(t+1) = x_0 \vee \overline{a_0 x_1} \vee \overline{a_1 x_2 x_3}, \\ Q_3(t+1) &= (a_0 \vee a_2 \vee a_5)(t+1) = x_0 \vee a_3 \vee a_4 \vee \overline{a_1 x_2} \vee \overline{a_2 x_4}. \end{aligned} \quad (5)$$

Выполняя в уравнениях (5) подстановку кодов состояний исходного автомата, получим систему уравнений (6), используемую для структурного синтеза:

$$\begin{aligned} Q_1(t+1) &= \overline{Q_1} \overline{Q_2} \overline{Q_3} \vee Q_2 Q_3 x_1 x_2, \\ Q_2(t+1) &= x_0 \vee Q_2 Q_3 x_1 \vee Q_2 \overline{Q_3} x_2 x_3, \\ Q_3(t+1) &= x_0 \vee \overline{Q_3} x_2 \vee Q_1 x_4 \vee \overline{Q_2} \overline{Q_3}. \end{aligned} \quad (6)$$

Для контроля правильности системы уравнений (6) выполняем ее детерминизацию. Для этого по уравнениям (6) строят вспомогательную прямую таблицу переходов (ПТП) для Q -событий (2).

На основании алгоритма детерминизации, представленного в [2], используя вспомогательную ПТП и учитывая требования к формированию полных событий (состояний) автомата и входных сигналов на переходах, получим прямую таблицу переходов детерминированного автомата (табл. 1).

Таблица 1

	Сочетание всех Q -событий в момент времени (t)	Сочетание частных входных сигналов на переходе $X(a_m, a_s)(t)$	Сочетание всех Q -событий в момент времени ($t+1$)
	Полное событие $a_m(t)$		Полное событие $a_s(t+1)$
1	$\overline{Q_1} \overline{Q_2} \overline{Q_3}$ a_0	$\overline{x_1 x_2}$ $(x_1 x_2)^*$ $\overline{x_1}$	$\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_3$ $\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_4$ $\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_1$
2	$\overline{Q_1} \overline{Q_2} \overline{Q_3}$ a_3	$\overline{x_4 x_2}$ $(\overline{x_4 x_2})^*$ $(x_4)^*$	$\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_5$ $\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_5$ $\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_5$
3	$\overline{Q_1} \overline{Q_2} \overline{Q_3}$ a_4	1	$\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_2$
4	$\overline{Q_1} \overline{Q_2} \overline{Q_3}$ a_1	$\overline{x_2 x_3}$ $(x_2 x_3)^*$ $\overline{x_2}$	$\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_1$ $\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_4$ $\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_5$
5	$\overline{Q_1} \overline{Q_2} \overline{Q_3}$ a_2	$\overline{x_4}$ x_4	$\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_5$ $\overline{Q_1} \overline{Q_2} \overline{Q_3} / a_4$

Дадим некоторые пояснения к построенной таблице 1. Частные входные сигналы, отмеченные звездочкой, являются дополнительными входными сигналами, обеспечивающими полноту переходов на соответствующем шаге алгоритма работы автомата. Обозначение сочетаний всех Q -событий принято таким же, что и при кодировании состояний исходного автомата.

По табл. 1 нетрудно построить СКУ для Q -событий, которая должна полностью соответствовать (до обозначений) исходной СКУ (4), представляющей функции переходов исходного автомата. Если такого соответствия нет, то это означает, что система СКУ для Q -событий, на основе которой должна строиться структура устройства управления, имеет ошибки и должна быть скорректирована.

Список литературы

1. Кларк, Э. М. Верификация моделей программ: Model Checking : пер. с англ. / Э. М. Кларк, О. Грамберг, Д. Пелед. – М. : МЦНМО, 2002. – 416 с.
2. Вашкевич, Н. П. Недетерминированные автоматы в проектировании систем параллельной обработки / Н. П. Вашкевич. – Пенза : Изд-во Пенз. гос. ун-та, 2004. – 280 с.

РАЗРАБОТКА МЕНЕДЖЕРА МЬЮТЕКСОВ С АППАРАТНОЙ ПОДДЕРЖКОЙ ДЛЯ ОПЕРАЦИОННЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Рассматриваются вопросы аппаратной реализации средств управления ресурсами операционной системой реального времени, в частности менеджера мьютексов. Предложен вариант реализации устройства управления мьютексами в составе аппаратного менеджера ресурсов.

Известно, что системы реального времени, особенно жесткого, требуют высокой реакции на запрос. Кроме того, они должны гарантировать заданные сроки выполнения. Поэтому перед разработчиками ставится задача увеличения производительности не только аппаратных средств, но и улучшения временных характеристик операционных систем. Традиционные операционные системы, основанные на программном обеспечении, зачастую не обеспечивают необходимых показателей производительности. Кроме того надежность и безопасность операционных систем, реализуемых в программном обеспечении, зачастую не удовлетворяет пользователей. Поэтому предлагается применить аппаратную поддержку ряда компонент ядра операционных систем реального времени, что позволит повысить как производительность, так и показатели надежности и безопасности ОСРВ.

В статье рассматривается устройство под названием менеджер ресурсов. Оно находится в составе блока являющимся аппаратным ускорителем процессора. Фактически этот ускоритель играет роль небольшой аппаратной операционной системы. Его задачами являются, во-первых, управление выполнением нескольких параллельных задач на процессоре. Во-вторых, в его обязанности входит предоставление задачам средств синхронизации и контроля доступа к общим ресурсам.

Современные операционные системы предлагают широкий набор средств для синхронизации взаимодействия параллельных процессов. Наиболее часто применяются двоичный/счетный семафор и мьютекс.

Двоичный семафор может быть использован для оповещения задачи о событии, как часть в механизме синхронизации между задачами, или для защиты одиночного общего ресурса.

Подсчитывающий семафор имеет тот же функционал что и двоичный семафор, но имеет счетчик с большей разрядностью, что обеспечивает управление множеством однотипных ресурсов. Текущее значение счетчика может инициализироваться как максимальным значением, так и равным 0, либо каким-то другим в зависимости от предполагаемого использования.

Мьютекс используется для предоставления задаче эксклюзивного доступа к общему ресурсу. В основном это тот же двоичный семафор, который имеет дополнительный функционал для предотвращения возникновения инверсии приоритетов. В отличие от семафора, который может быть освобожден любой задачей, в независимости от того, кто его захватил, мьютекс может быть освобожден только задачей, которая его захватила. При попытке повторного захвата мьютекса одной и той же задачей, ее выполнение будет остановлено.

Управление ресурсами в операционных системах возлагается на менеджер ресурсов. На рис. 1 представлена архитектура аппаратного менеджера ресурсов, состоящего из двух частей: менеджера мьютексов и менеджера семафоров.

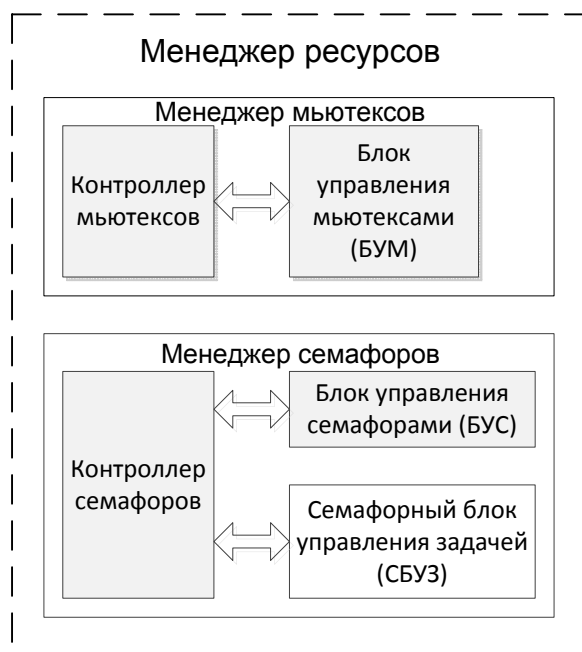


Рис. 1. Архитектура менеджера ресурсов

Менеджер мьютексов состоит из двух компонентов: контроллера мьютексов и блока управления мьютексами (БУМ). Для защиты доступа к мьютексу используется протокол Priority Ceiling [1]. При использовании этого протокола задача гарантированно получит доступ к запрашиваемому мьютексу, по этой причине нет необходимости поддерживать очередь ожидания для каждого мьютекса.

Задача никогда не должна блокировать себя, если у нее есть захваченный мьютекс, так как это не даст выполниться задачам, приоритет которых ниже Ceiling Priority. Принимая во внимание это условие, а так же соблюдая требование освобождения мьютексов в обратном порядке, чем они были захвачены, появляется возможность создать эффективную аппаратную реализацию, которая будет сделана с использованием односвязного списка структур.

БУМ хранит всю необходимую информацию специфичную для каждого мьютекса, такую как: Ceiling Priority, ID владеющей задачи, сохраненный Ceiling Priority и ID следующего мьютекса в очереди захваченных мьютексов. Поле, содержащее ID задачи, используется для предотвращения того что бы задача, не владеющая мьютексом, не смогла его освободить, а поле сохраненный Ceiling Priority используется для быстрого восстановления системного Ceiling Priority, когда освобождается мьютекс из начала очереди. Поле ID следующего мьютекса используется для того, что бы контроллер знал какой мьютекс окажется в начале очереди, после того как текущий мьютекс будет освобожден.

Контроллер мьютексов содержит поля необходимые для того что бы следить за началом и концом связанного списка, а так же системным Ceiling Priority. Помимо полей данных, контроллер мьютексов содержит всю необходимую логику и конечный автомат для поддержки выполнения операций с мьютексами.

Менеджер мьютексов реализуется так, как описано выше. Его структура включает 2 элемента (рис. 2): блок управления мьютексами (БУМ) и контроллер мьютексов.

БУМ содержит все необходимые поля данных для каждого мьютекса. Эти поля сгруппированы в 2 регистра и доступны для API процессора и контроллера мьютексов. Регистры БУМ содержат:

- MCPR – регистр потолочного приоритета мьютекса;
- MCDR – регистр управляющих данных мьютекса.

Регистр MCPR содержит потолочный приоритет мьютекса, который должен назначаться на основе приоритетов задач, который могут захватывать данный мьютекс. Реализация аппаратного планировщика требует, чтобы потолочный приоритет мьютекса был на единицу больше самого высокого приоритета среди задач, которые могут захватывать его. Этот регистр доступен на чтение и на запись для API процессора, и доступен только для чтения контроллеру мьютексов.

Регистр MCDR содержит 4 поля: MU (мьютекс используется) – это 1-е битное поле, в которое записывается «1», если мьютекс захвачен и находится в очереди мьютексов. Поле NMID (ID следующего мьютекса) используется для хранения ID следующего мьютекса в связанном списке. SSCP (сохраненный системный приоритет) поле хранит потолочный приоритет системы до того как мьютекс был захвачен, используется при освобождении мьютекса. Последнее OTID (ID владеющей задачи) поле содержит ID задачи, захватившей мьютекс. Это поле используется для предотвращения освобождения мьютекса другой задачей. Этот регистр доступен контроллеру мьютекса для чтения и записи, а API процессору только на чтение.

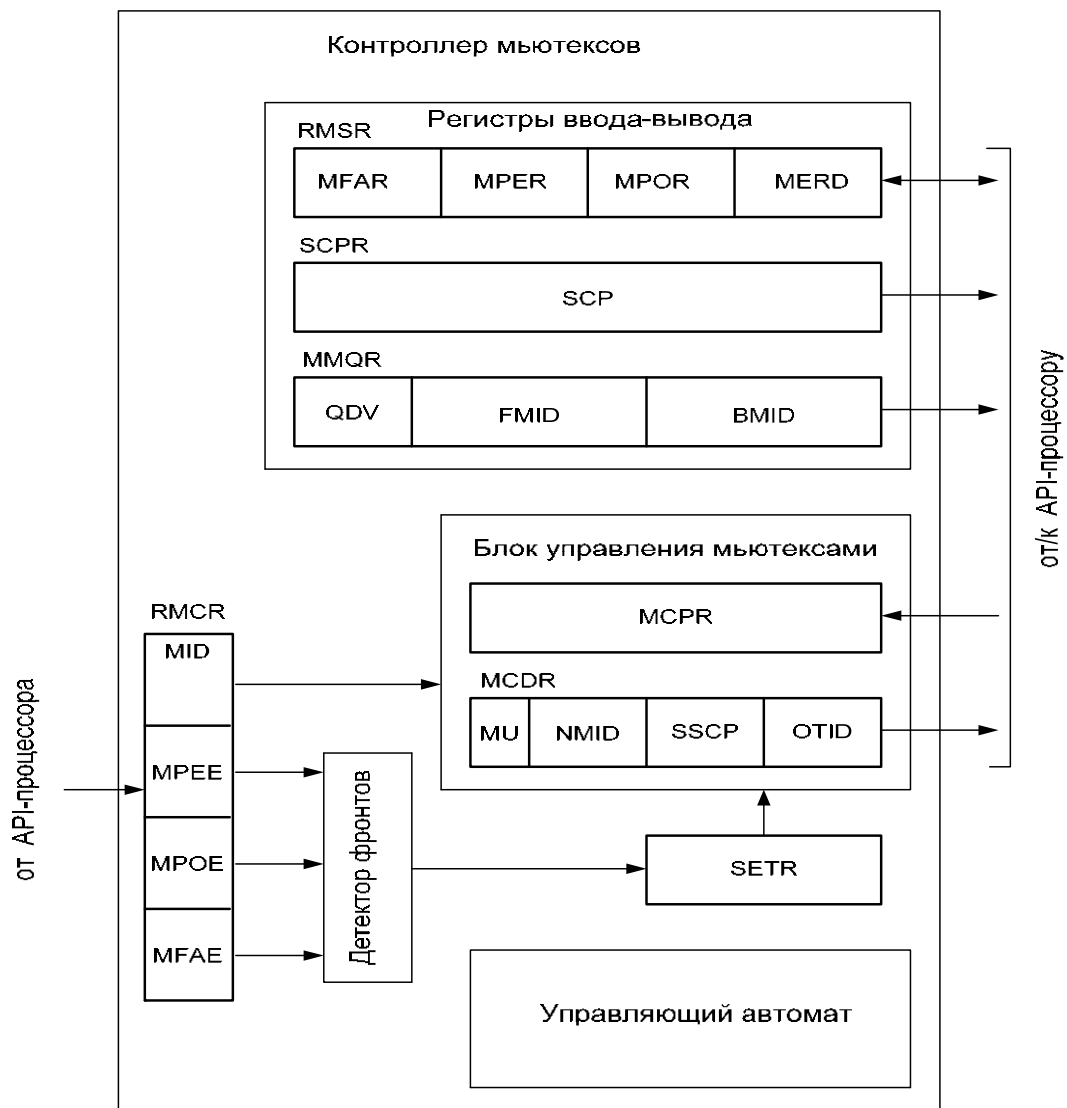


Рис. 2. Архитектура менеджера мьютексов

Контроллер мьютексов осуществляет доступ к нужному мьютексу с помощью сигнала MI, а API процессор может осуществлять доступ посредством записи ID мьютекса в поле MID регистра RMCR.

Контроллер мьютексов содержит всю логику и регистры необходимые для реализации операций над мьютексами: «Освободить все», «Захватить», «Освободить». Регистры, принадлежащие контроллеру мьютексов, включают в себя:

- SCPR – регистр системного потолочного приоритета;
- MMQR – регистр очереди менеджера мьютексов.

Регистр SCPR содержит системный потолочный приоритет. Этот регистр доступен на чтение и на запись для менеджера мьютексов, и только для чтения для API процессора и диспетчера задач.

Регистр MMQR содержит 3 поля: QDV, FMID и BMID. Поле QDV (данные в очереди действительны) это однобитное поле, в которое записывается «1», когда поля FMID и BMID имеют действительные данные. FMID (ID первого мьютекса в очереди) поле содержит ID последнего захваченного мьютекса, мьютекс на который указывает это поле должен быть освобожден раньше всех остальных. Поле MID (ID последнего мьютекса в очереди) указывает на первый захваченный мьютекс; этот мьютекс может быть освобожден самым последним.

Кроме регистров SCPR и MMQR контроллер мьютексов имеет выходные регистры для всех сигналов, а так же входные регистры и детекторы фронтов для сигналов событий. Менеджер мьютексов использует регистр CETR для проверки ID выполняющейся задачи при операции «Освобождение», и сохраняет этот ID задачи в регистр OTID при операции «Захват».

Для управления менеджером мьютексов используются два регистра:

- RMCR – регистр управления диспетчером ресурсов;
- RMSR – регистр статуса диспетчера ресурсов.

MFAE (событие освобождения всех мьютексов), MPPE (событие захвата мьютекса) и MPOE (событие освобождения мьютекса) биты регистра RMCR подключены к детектору фронтов, регистрирующему события нарастания (RE) и спада (FE). Эти события для сигналов MFAE, MPPE и MPOE используются для управления выполнением операций над мьютексами, реализуемых внутри конечного автомата управления мьютексами.

MID (ID мьютекса) поле регистра RMCR содержит ID мьютекса для операций «Захват»/«Освобождение».

MFAR, MPER, MPOR биты записываются контроллером мьютексов в регистр RMSR, как биты подтверждения для каждой из операций: «Освобождение всех», «Захват», «Освобождение». MERD это поле содержащее код ошибки, после совершения операции над мьютексом.

Конечный автомат управления мьютексами фактически состоит из 4-х автоматов. Один главный автомат контролирует общее состояние менеджера мьютексов и три подавтомата, каждый из которых срабатывает для выполнения конкретной операции. На рис. 3 показан главный автомат и три подавтомата для каждой операции. Главный автомат содержит 5 событий: «Инициализация», «Ожидание», «Освобождение всех», «Захват», «Освобождение захваченного мьютекса».

Подавтомат «Освобождение всех» используется для того, что бы все мьютексы были освобождены до того как вся система начнет работать в нормальном режиме. Если этот автомат запустится во время нормальной работы, в то время как один или более мьютексов захвачены, приложение будет работать некорректно, следовательно-

но, в целях уменьшения риска происхождения этого, при вызове API функции, выполняющей данную операцию, таймер тиков, планировщик и диспетчер должны быть отключены. Первым делом автомат перебирает все захваченные мьютексы и сбрасывает бит MU у каждого из них, а затем очищаются SCPR и MMQR регистры контроллера мьютексов.

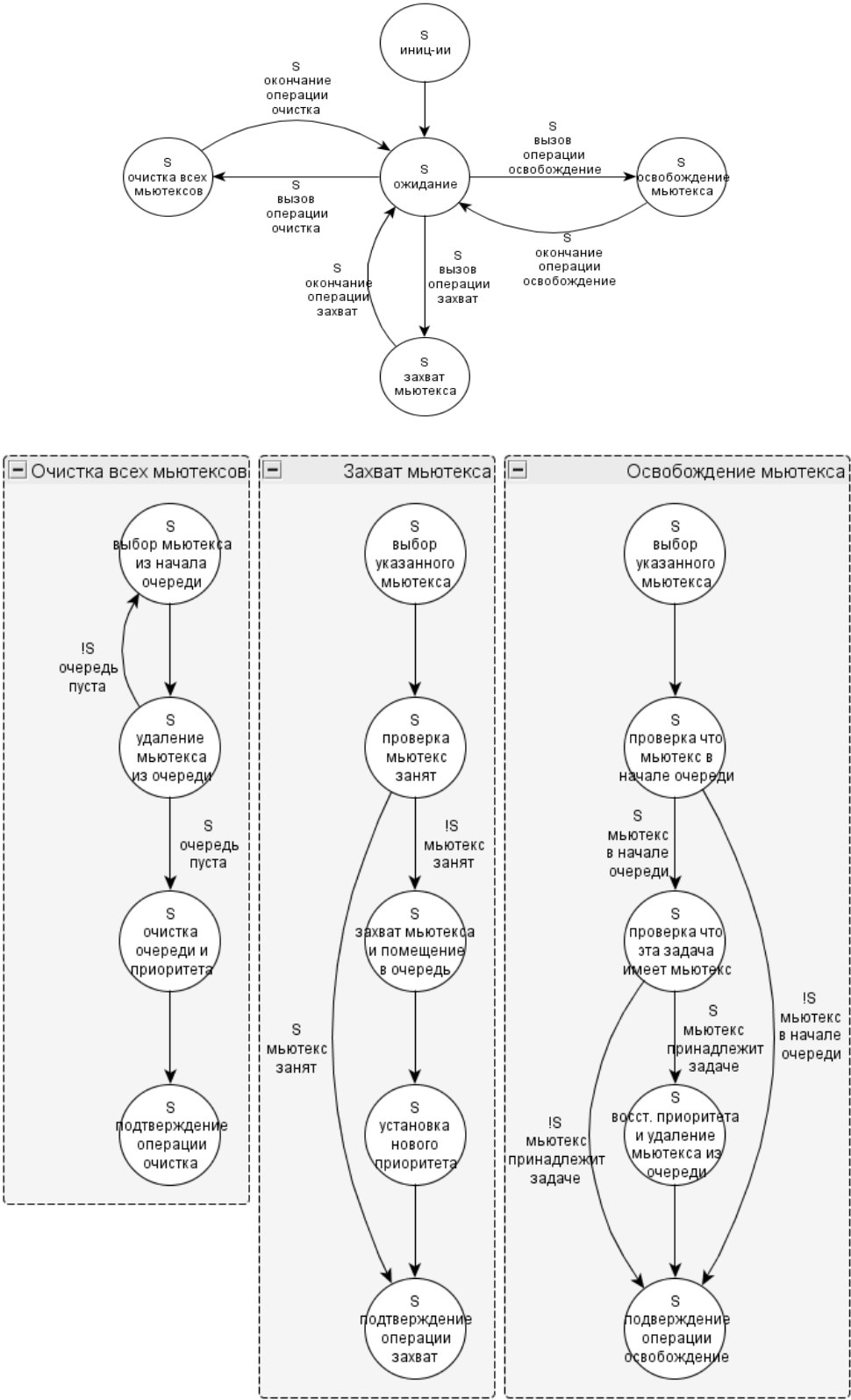


Рис. 3. Главный автомат и подавтоматы менеджера мьютексов

Подавтомат «Захват» используется для захвата мьютекса выполняющейся задачей. Сначала автомат выбирает мьютекс, на который указывает поле MID. Потом он проверяет, что мьютекс является свободным, если мьютекс свободен, он захватывает его и помещает в очередь мьютексов, текущий потолочный приоритет сохраняется и затем обновляется системный потолочный приоритет.

Подавтомат «Освобождение» используется для освобождения мьютекса выполняющейся задачей. Сначала автомат выбирает мьютекс, на который указывает поле MID. Затем проверяет, что мьютекс находится в начале очереди, и что он принадлежит выполняющейся сейчас задаче, если оба условия выполняются, то мьютекс освобождается: он удаляется из очереди и восстанавливается потолочный системный приоритет.

Список литературы

1. Burns, A. Real-Time Systems and Programming Languages – Ada 95, Real-Time Java and Real-Time /Alan Burns, Andy Wellings (March 2001) POSIX. – 3rd ed. – Addison Wesley Longmain, 2001.
2. A scalable single-chip multi-processor architecture with on-chip RTOS kernel / B. D. Theelen, A. C. Verschueren, V. V. Reyes Suárez, M. P. J. Stevens, A. Nuñez // Journal of Systems Architecture: the EUROMICRO Journal – Special issue: Synthesis and verification archive. – 2003. – Vol. 49. – Iss. 12–15, Dec. – P. 619–639.
3. Kohout, P. Hardware support for real-time operating systems / P. Kohout, B. Ganesh, and D. Jacob // Codesign and System Synthesis, 2003. First IEEE/ACM International Conference on. – 2003. – P. 45–52.

ЭЛЕКТРОННЫЙ ДОКУМЕНТООБОРОТ И ЭЛЕКТРОННОЕ ПРАВИТЕЛЬСТВО

Н. Р. Кашапов, Р. А. Файрузов

Россия, Казань, Казанский национальный исследовательский технический университет им. А. Н. Туполева

АНАЛИЗ РЫНКА СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

На рынке систем электронного документооборота имеется достаточно немаленькое количество готовых профессиональных решений. Целью работы является узнать, какие решения являются наиболее эффективными и наиболее используемыми на отечественном рынке систем электронного документооборота.

Развитие человеческой цивилизации сопровождается поражающим воображение увеличением объема создаваемой, обрабатываемой и хранимой информации. Впечатляющие цифры и данные объемов информации говорят только о том, что для любого предприятия или организации вопросы оптимизации документооборота и контроля за обработкой информации имеют ключевое значение. Именно поэтому эффективность управления предприятиями и организациями не в последнюю очередь зависит от корректного решения задач оперативного и качественного формирования электронных документов, контроля их исполнения, а также продуманной организации их хранения, поиска и использования. Потребность в эффективном управлении электронными документами и привела к созданию систем электронного документооборота (СЭД). Российский рынок предлагает несколько профессиональных решений для внедрения и интеграции системы электронного документооборота внутри компании. Далее, мы рассмотрим имеющиеся программные продукты, сравним и выберем наиболее экономически выгодное решение.

IBM Notes/Domino (старое название *Lotus Notes*) – программный продукт, система электронного документооборота с основным акцентом на групповую работу. Разработанный в компании Lotus, а затем перекупленный фирмой IBM, является одним из наиболее отлаженных и стабильных продуктов. Во многих фирмах РФ, распространена старая версия – *Lotus Notes/Domino*. *IBM Notes* включает в себя средства электронной почты, персональные и групповые календарей, службы мгновенных сообщений и среду исполнения приложений делового взаимодействия (средства одновременной работы над документами и управление динамическими процессами). Основной задачей *IBM Notes* является группирование и концентрирование всей информации, относящейся к рабочим процессам и документообороту, в единую сеть. Особенностью системы является объектно-ориентированная архитектура, благодаря которой возможно создание информационных систем, автоматизирующих работу со сложно структурированными данными и неформализованными динамическими процессами.

Основным преимуществом данного продукта является мультиплатформенность, универсальность, наличие большого количества компонентов, позволяющих автоматизировать работу над документами и легкость в обслуживании. Стоимость продукта зависит от типа организации и количества клиентов в системе. Существует как облачная, так и обычная версии программного обеспечения. В целом, учиты-

вая цену, IBM Notes рассчитан на большие компании с большими потоками данных и большим количеством персонала.

Directum – программное обеспечение, внедряющая концепцию Business Process Management с использованием процессов автоматизации и взаимодействием на основе механизма workflow. Документооборот осуществляется на основе глубокой интеграции системы с другими компонентами. Directum не только позволяет поддерживать систему ЭДО, но и так же вести контроль над бизнес-процессами, делать анализ и проводить оптимизацию существующей системы. Программный комплекс состоит из нескольких частей, взаимодополняющих друг друга и создающих общую структуру по документообороту, анализу имеющихся бизнес-процессов.

Положительными качествами данной системы является быстрая развертка, масштабируемость в компаниях различного размера, наличие, как десктопного, так и веб-приложения. Среди минусов можно отметить дороговизну (порядка полумиллиона рублей на 60 рабочих мест), и обязательное наличие в компании специалиста, который будет обслуживать систему и производить обучение персонала.

Microsoft WorkLite SharePoint – профессиональное решение для создания больших и высоконагруженных сетей документооборота с глубокой интеграцией с продуктами корпорации Microsoft. Широкий спектр возможностей sharepoint позволяют запустить масштабную и целостную систему без крупных затрат.

Среди преимуществ можно отметить высокую интеграцию с другими продуктами Microsoft. Но в тоже время, поддержка SharePoint требует высококвалифицированных обученных специалистов.

E1 Ефрат – система, полностью ориентированная на электронный документооборот. Она включает в себя процессы ввода и регистрации документов, работу с электронными документами, контроль исполнения, рассылку документов и внутреннюю почту. Основными достоинствами являются простота, и относительно невысокая стоимость.

СЭД «ДЕЛО» – система, ориентированная на автоматизацию документооборота с возможностью сквозной автоматизации документооборота территориально-распределенных компаний, поддерживающая интеграцию с существующими информационными системами предприятия. СЭД «ДЕЛО» позволяет решать задачи по автоматизации работы с документами компаний различных масштабов и сфер деятельности, поддерживает бумажный, смешанный документооборот, что позволяет плавно перейти к электронному документообороту с минимальными финансовыми и материальными затратами. Система ориентирована на автоматизацию смежных документо-ориентированных процессов (финансовый, договорной документооборот, автоматизация внутренних бизнес процессов, работа с обращениями граждан, оказание госуслуг и т.д.), с возможностью организации юридически значимого документооборота компании. Наличие открытого API-интерфейса обеспечивает возможность интеграции системы «ДЕЛО» с любыми информационными системами и бизнес-программами, используемыми организацией-заказчиком. Цена ниже на 15 % чем у конкурентов (в лице IBM Notes и E1 Ефрат).

Мегаплан – система автоматизации бизнеса, включающая в себя модуль ориентированный на документооборот. Главной особенностью этой системы является гибкость системы и удобство пользования. Однако, как в примере с SharePoint или «Дело», расширить функционал или настроить систему под конкретные условия невозможно. Система создана гибкой и масштабируемой, поэтому в ней нет специализированных возможностей или каких-либо отличительных черт. В экономическом плане, Мегаплан предлагает облачную версию, с небольшой ежемесячной платой и готовое серверное решение, цена которого на уровне конкурентов.

Выше были представлены профессиональные решения, ориентированные на крупный бизнес. В процессе сравнения систем, можно заметить, что функционал, принципы работы и цены практически одинаковы. Самые передовые продукты предлагают облачное исполнение и гарантированную поддержку. Даже маленькие фирмы и компании с штатом менее 100 человек могут себе позволить профессиональное решение на основе облачных технологий. Государственным и образовательным учреждениям предоставляются скидки. Тот факт, что все программные продукты мультиплатформенны и имеют облачные реализации, означает, что проблем с имеющимися системами возникнуть не должно.

Какой продукт выбрать для лучшего роста эффективности компании, должен решать департамент информационных технологий компании отталкиваясь от бюджета и требований системы. Системы имеющие особенности, требующие специфичные решения, требуют интеграции настраиваемой системы, например – SharePoint. Для компаний, ведущих деятельность, не отличающуюся сложностью и специфичностью, стоит обратить на универсальные решения – «Мегаплан» и «Дело», состоящие не только из системы электронного документооборота, но из дополнительных подсистем контроля качества, автоматизации процессов бизнеса и т.д.

В сфере электронного документооборота в малых и средних организациях открытой проблемой остается проблема финансирования и поддержки системы электронного документооборота. С подобной проблемой зачастую сталкиваются учебные заведения и стартапы, которые не имеют стартового капитала, но развивающиеся очень быстро. Решением проблемы все чаще становятся облачные файловые хранилища. Дополненные почтовой службой и дублирующими системами, эффективность простых систем не сильно уступает профессиональным системам электронного документооборота.

В итоге, можно сделать вывод что рынок продукции, связанной с электронным документооборотом насыщен, но не имеет потенциала роста. Программы отлажены и проверены временем, а новые возможности внедряются очень быстро и у всего рынка сразу. Для любых сфер бизнеса можно подобрать продукт, учитывая капитал и потребности фирмы.

И. Г. Кревский

Россия, Пенза, Пензенский государственный университет

ОСНОВНЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ ЭЛЕКТРОННОГО ОБУЧЕНИЯ

Рассмотрены основные тенденции развития Электронного обучения (ЭО). Рассмотрены основные изменения интернет-технологий и ЭО, которые могут произвести революционные преобразования в образовании. Развитие МООС и ЭО в целом, создают невиданные возможности для повышения доступности образования, но могут привести к вытеснению с образовательного рынка многих существующих вузов, несут угрозу потери многими странами суверенитета в области образования.

Дистанционные образовательные технологии (ДОТ) на базе компьютерных сетей развиваются уже более 20 лет. Распространенный англоязычный термин e-learning переключался в российское законодательство и нормативные акты: ст. 16 ФЗ-273 называется «Реализация образовательных программ с применением электрон-

ного обучения и дистанционных образовательных технологий» [1]. С момента возникновения электронное обучение (ЭО) считалось перспективным направлением развития, его использовали в той или иной степени все больше образовательных организаций. Произошедшие за последние годы изменения в развитии интернет-технологий и новые тенденции в ЭО, по мнению многих экспертов, в ближайшее время могут до неузнаваемости изменить мировую систему образования в целом. Основные факторы таких изменений:

1. Качественное повышение скорости доступного населению интернета и, как следствие, превращение его в основной канал передачи любого контента (включая мультимедийный), широкое распространение видеотрансляций и систем видеосвязи, широкое проникновение скоростного интернета.
2. Превращение интернета в основной источник информации, в том числе, бесплатной и свободно распространяемой, развитие поисковых систем. Следствием этого стало исчезновение монополии университетов на обладание нужными для получения образования знаниями.
3. Сокращение государственной поддержки вузов в большинстве стран мира, замедление экономического роста и, как следствие, снижение частных инвестиций. В результате – обострение борьбы университетов за студентов и материальные ресурсы, поиски более экономичных образовательных моделей.
4. Появление и широкое распространение массовых, открытых для всех *online* курсов (МООС – *massive open online courses*).

Интересный анализ экономики образования в США и МООС выполнен С.Л.Тимкиным в ряде работ и записей в блогах, например [2]. В целом, схожие выводы сделаны и в [3]: для наиболее известных и богатых университетов расширение их популярности, благодаря МООС, в конечном счете, может привести к еще большей их популярности и возможности предлагать как элитное (дорогое) образование, так и массовое, относительно дешевое, основанное на МООС и сертификации (за умеренную сумму) пройденных в интернете курсов. Так как создание конкурентоспособного курса МООС и возможность его бесплатного вывода на рынок требуют значительного стартового капитала, а также из-за конкурентного преимущества в виде раскрученных брендов, ведущие университеты имеют лучшие возможности по предоставлению курсов МООС и привлечению к ним внимания слушателей. В то же время для большинства вузов развитие МООС и перспектива дальнейшего признания результатов обучения на этих курсах может значительно потеснить их рыночные позиции.

Надо отметить, что в США развитие МООС пользуется активной поддержкой правительства для развития дешевой альтернативы высшему образованию на основе компетентной модели. В сенате штата Калифорния рассматривается специальный законопроект, обязывающий университеты штата при определенных условиях засчитывать результаты обучения по сертифицированным курсам МООС, против которого решительно выступают отдельные группы и профсоюзы преподавателей. Наиболее популярными платформами МООС на сегодняшний день являются Coursera[4], Udacity [5] и edX [6]. Российские университеты (ВШЭ, МФТИ и Санкт-Петербургский ГУ) начали размещение своих первых курсов на Coursera. Также начали работать и российские площадки МООС. 17 октября 2014 г. подписано соглашение между Минобрнауки России и некоммерческой организацией edX, созданной университетами MIT и Harvard, которое обеспечит возможность использования российскими вузами открытого программного обеспечения edX для создания национальной площадки открытого образования, а также публикации открытых онлайн-курсов на площадке edx.org.

Таким образом, система образования вступает в эпоху глобальной конкуренции. Так как большинство курсов на ведущих площадках МООС представлены на английском языке, «защитой» российского образовательного рынка выступает плохое знание большинством наших сограждан, включая студентов, английского языка. Однако на Coursera уже появился русскоязычный интерфейс. К целому ряду курсов энтузиасты подготовили субтитры на русском языке, разрабатывается сайт с переводом курсов Coursera. По некоторым оценкам на Coursera изучают курсы около 300 000 слушателей из России.

Развитие МООС и ЭО в целом, с одной стороны, создают невиданные ранее возможности для повышения доступности образования, с другой – могут привести к вытеснению с образовательного рынка многих существующих вузов, возможно, несут угрозу потери многими странами суверенитета в области образования. Ведущие вузы получают возможность экспансии в дешевый ценовой сегмент. Например, технический университет Джорджии в Атланте предлагает программу магистра Computer Science в кампусе за 45 тыс. долл., а в интернете – за семь тысяч [3]. Ограничением может выступить стартовый уровень подготовки студентов. Преподавание в ведущих вузах, например, в Гарварде или МГУ характеризуется не только высоким качеством, но и ориентацией на аудиторию с уровнем начальной подготовки и способностей значительно выше среднего. Поэтому несмотря на появление доступа к качественному недорогому образованию, не все смогут им воспользоваться. Едва ли ведущие вузы пойдут на выдачу одинаковых дипломов по результатам изучения в сети как первоклассных, но сложных для восприятия, так и упрощенных вариантов программ MIT или МФТИ. Можно предположить, что именно готовность и способность студентов воспринимать программы на уровне ведущих вузов станет, в конце концов, ограничителем захвата ими образовательных рынков. Хорошую рыночную долю могут получить несколько менее престижных вузов, готовых предложить качественные курсы, ориентированные на массовую аудиторию.

В докладе рассматривается текущая ситуация с развитием ЭО в России, действующая нормативная база и перспективы ее развития с учетом мнений экспертов и проектов документов.

Список литературы

1. Федеральный закон Российской Федерации от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» // СПС «КонсультантПлюс» (дата обращения: 30.10.2014).
2. Тимкин, С. Л. МООК и экономика образования в США. Теория подрывных инноваций применительно к МООК и ее критика / С. Л. Тимкин. – URL: <http://www.slideshare.net/ssuser/454b72/ss-26482104>
3. Сумленный, С. Останется десять университетов – интервью с Эрвином Хеберле / С. Сумленный // Эксперт. – 2013. – 2 дек. – № 48 (878) – URL: <http://expert.ru/expert/2013/48/ostanetsya-desyat-universitetov/>
4. Образовательный портал Coursera. – URL: <https://www.coursera.org/>
5. Образовательный портал Udacity. – URL: <https://www.udacity.com/>
6. Образовательный портал EDX. – URL: <https://www.edx.org/>

А. А. Кручинин

Россия, Пенза, Пензенский государственный университет

ПОДХОД К ФОРМАЛИЗАЦИИ ПРОЦЕССА ЭКСПЕРТНОЙ ОЦЕНКИ НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ И ОПЫТНО-КОНСТРУКТОРСКИХ РАБОТ

Экспертная оценка целевых показателей выполнения НИР и ОКР обычно делается исходя из контекста выполнения работ. Перейдя от анализа документов к анализу функциональных требований можно сформулировать требования к автоматизации данного процесса

В настоящее время во всем мире большое внимание уделяется финансированию наукоемких проектов, как одному из основных показателей развития экономики. При этом, важное значение имеет как оценка значимости результатов исследования, так и защита от мошеннических действий с целью получения финансирования на нецелевые нужды.

С этой целью применяются различные экспертные методы оценки на разных стадиях выполнения: начиная с анализа постановки задачи и способов выполнения на этапе конкурсного отбора (с целью получения финансирования) и заканчивая сдачей результатов работ в ходе испытаний и подтверждения заявленных характеристик. Обычной практикой является контроль выполнения работы на основе заданных целевых показателей.

Существует возможность получения финансирования по работам с не достаточно конкретными целевыми показателями, конечно при условии прохождения конкурсного отбора. Но дело в том, что конкурсный отбор проходит на основании комплексной оценки, в которой доля собственно научных факторов, как правило, составляет менее 40 %. Более того, на момент проведения конкурса не возможно сформулировать в достаточно однозначном виде целевые показатели, что особенно актуально для фундаментальных и поисковых научно-исследовательских работ, где результат заранее не определен.

Наиболее эффективным способом контроля является привлечение финансирующей организацией независимых экспертов-«мониторов». Они могут проводить аудит работы на всех этапах.

Основными приемами работы экспертов-мониторов является контроль порядка ведения работ, состава и полноты документирования на соответствие нормативным документам, как правило, государственным и отраслевым стандартам (например, можно рассмотреть один из регламентов [1]).

При этом, о качестве выполняемой работы обычно можно судить по таким признакам как полнота документирования, структурирование и связанность документации, контекстная связь целевых показателей с постановкой задачи, прозрачность процедуры испытаний (методы испытаний с применением отраслевых норм), этапность работ и соответствие состава документации проводимому этапу.

Таким образом, при таком подходе непосредственно в процессе выполнения работы реализуется проверка некоторых классических критериев научности: используется специализированный научный язык (отраслевой, заданный требованиями к оформлению документов), позволяющий включить его в состав определенной научной теории; в процессе испытаний результатов (эмпирическое подтверждение) достигается проверяемость, доказательность и обоснованность результатов; результаты систематизированы для последующего использования. Отдельно, в явном ви-

де, должны быть раскрыты и другие критерии. Например, рациональность, как правило, явно присутствует в разделах технико-экономического обоснования. Но эти факторы достаточно сложно проверить в ходе работы, так как требуется некоторый период «эксплуатации» результатов.

Не трудно увидеть, что контроль выполнения, как и подтверждения «научности» работы основан на сопоставлении деятельности разработчиков некоторым шаблонам в контексте конкретных областей. Исходя из этого можно сделать вывод о возможности автоматизации процесса аудита, а так же построения систем поддержки циклов разработки в заданных предметных областях на основе некоторых базы знаний.

Принципиально, данная задача решается на множестве информационных систем с применением технологий docflow и workflow обработки, когда определенному документу (или в общем виде – объекту деятельности) задается шаблон и порядок его согласования разными разработчиками-ролями.

С другой стороны, контролируемым параметров является не сам документ, а отраженные в нем целевые показатели - как правило, функциональные требования. С этой точки зрения можно рассматривать документ как набор утверждений о предметной области, которые в процессе работы переходят из документа в документ (или иную форму), обретая более конкретные очертания и, собственно, являются результатами работ.

Здесь удобно использовать термины и методологию, использующуюся в феноменологии. Методические принципы феноменологии сформулированный Э. Гуссерлем подразумеваю следующий процесс [2]:

- необходимо установить первые очевидности, которые лягут в основу достоверного знания;
- очевидности должны быть аподиктическими или ассерторическими: очевидное сейчас может стать сомнительным позднее, оказаться видимостью, иллюзией;
- путем опытов следует от ассерторических очевидностей перейти к абсолютным очевидностям.

Если этот метод наложить на стандартный процесс выполнения исследовательской или конструкторской работы, то получим следующее:

- начальные требования являются ассерторическими, так как нельзя заранее гарантировать их корректность;
- переходя из документа в документ требования обретают смысл функционального объекта с заданными характеристиками или заменяются другими требованиями;
- функциональный объект может быть рассмотрен как результат работы.

Такой подход, в большей степени позволяет оценить «научность» и «обоснованность» исследовательской деятельности. Но вместе с тем это требует и новых автоматизированных информационных систем (АИС) поддержки. Назначением таких АИС является: контроль выполнения НИР (ОКР), поддержка жизненного цикла на этапе разработки, аудит процесса и результата.

Для этого в их состав должны входит следующие компоненты:

- редактор документов как совокупности сущностей;
- редактор алгоритмов уточнения сущностей;
- редактор шаблонов документов;
- редактор ролей;

- библиотека шаблонов документов, библиотека алгоритмов, библиотека сущностей, библиотека ролей, библиотека профилей работ (примеры отношений между шаблоном, алгоритмами и ролями);
- набор инструментов для реверсивного анализа работ.

Естественно, что для реализации таких задач АИС должна использовать свое собственное лингвистическое обеспечение: язык описания документов/сущностей, язык описания алгоритмов уточнения сущностей (например графический – на основе R-схем [3]).

Создание АИС поддержки разработки по приведенным правилам может в значительной степени упростить как аудит процесса исследований и разработки, так и на ранних стадия помочь разработчику выбрать правильный путь развития, что в целом должно снизить риск нецелевого финансирования и повысить рентабельность инновационной деятельности.

Список литературы

1. Регламент представления и рассмотрения отчетной документации по договорам об условиях представления и использования субсидий на реализацию комплексного проекта по созданию высокотехнологичного производства. Утв. Департамент стратегии, анализа и прогноза Министерства образования и науки РФ. 06.05.2013.
2. Гуссерль Э. Идеи к чистой феноменологии и феноменологической философии / Э. Гуссерль. – М. : ДИК, 1999. – Т. 1. – С. 293
3. ГОСТ 19.005–85. Единая система программной документации. Р-схемы алгоритмов и программ. Обозначения условные графические и правила выполнения.

В. А. Ушаков

Россия, Санкт-Петербург, Санкт-Петербургский государственный университет аэрокосмического приборостроения

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ДЕКАНАТА В ХОДЕ ПОДГОТОВКИ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ БАКАЛАВРА

Описывается проблема повышения эффективности деятельности деканата путем разработки автоматизированного рабочего места декана.

В ходе подготовки выпускной квалификационной работы бакалавра разработано приложение «Автоматизированное рабочее место декана» (рис. 1) [1–2] – это электронный журнал, содержащий информацию: данные о студентах, их академические задолженности и учебные планы групп. Целью данного приложения является упрощение и автоматизация работы деканата, а именно анализ академических задолженностей студентов и составление статистических отчетов деканата.

Данное приложение написано в Delphi 7 на языке Object Pascal [3]. Оно получает исходные данные из автоматизированной информационной системы (АИС) «Образование» – номера групп, анкетные данные студентов, учебные планы. Выгрузка из АИС осуществляется через Microsoft Excel. Оперативная информация, такая как данные об итоговой аттестации, вводится вручную.

Автоматизированное рабочее место учебной части Сер

Файл Отчет о движении контингента студентов Заполнить сводку движения контингента ОШИБКИ !!! О программе

Фильтр

- ☒ все студенты
- ☐ по № факультета
- ☐ по № курса
- ☐ по направлению
- ☐ по № группы
- ☐ по бюджет/контракт
- ☐ по н/ж/м
- ☐ по ак отпуску
- ☐ отчисленные студенты

Отобразить таблицу на графике

Выполнить

Движение контингента

ФИО	группа	курс	Ср. балл	Итого	посл. сдача	1 семестр	2 семестр	3 семестр	4 семестр	б/к	н/ж
1. Абрамов Александр Петрович	5210М	2	5	3	20.06.2013	3	0	0		б	м
2. Бердина Марина Юрьевна	5210М	2	4,56	0	20.06.2013	0	0	0		б	ж
3. Васильев Игорь Анатольевич	5210М	2	3,23	0	20.06.2013	0	0	0		б	м
4. Ежова Елена Геннадьевна	5210М	2	3,92	0	20.06.2013	0	0	0		б	ж
5. Мишра Ольга Владимировна	5210М	2	3,01	4	01.01.2013	2	2	0		б	ж
6. Котов Вадим Петрович	5210М	2	3,72	1	20.06.2013	0	0	1		б	м
7. Маркелова Наталья Викторовна	5210М	2	4,01	0	20.06.2013	0	0	0		б	ж
8. Оленев Валентин Леонидович	5210М	2	5	0	20.06.2013	0	0	0		б	м
9. Семенов Лодмила Борисовна	5210М	2	4,91	3	01.01.2013	2	1	0		б	ж
10. Филь Юрий Юрьевич	5210М	2	3,47	1	20.06.2013	0	0	1		б	м
11. Вересов Игорь Васильевич	5230М	2	5	0	10.10.2013	0	0	0		б	м
12. Грабарь Анатолий Григорьевич	5230М	2	3,11	2	01.01.2013	1	0	1		б	м
13. Ершов Николай Владимирович	5230М	2	4,12	1	15.06.2013	1	0	0		б	м
14. Изотов Олег Альбертович	5230М	2	4,96	3	01.01.2013	0	1	2		б	м
15. Коробкова Светлана Николаевна	5230М	2	4	4	18.01.2013	0	2	2		б	ж
16. Куляев Анатолий Николаевич	5230М	2	3,19	0	15.06.2013	0	0	0		б	м
17. Манзее Виктор Яковлевич	5230М	2	4,5	1	01.01.2013	0	0	1		б	ж
18. Михалев Сергей Александрович	5230М	2	3,01	0	15.06.2013	0	0	0		б	м
19. Пашиков Валерий Павлович	5230М	2	3,5	2	01.01.2013	0	0	2		б	м
20. Поляков Юрий Сергеевич	5230М	2	3	2	01.01.2013	0	1	1		б	м
21. Семенов Андрей Владимирович	5230М	2	4,96	1	01.01.2013	0	1	0		б	м
22. Сешиков Алексей Юрьевич	5230М	2	4	0	24.05.2013	0	0	0		б	ж
23. Усикова Ирина Васильевна	5230М	2	3,19	1	15.06.2013	1	0	0		к	ж
24. Хохлова Надежда Ивановна	5230М	2	4,5	3	18.01.2013	3	0	0		б	ж
25. Юдин Павел Юрьевич	5230М	2	3,01	0	15.06.2013	0	0	0		б	м
26. Жилиева Юлия Константиновна	5310М	1	5	0	01.01.2015	0				б	ж
27. Иванушкин Антон Олегович	5310М	1	4	0	01.01.2016	0				б	м
28. Карнаукова Лиза Евгеньевна	5310М	1	5	0	01.01.2017	0				б	ж
29. Кондратский Максим Олегович	5310М	1	4	0	01.01.2018	0				б	м
30. Кривоша Елена Николаевна	5310М	1	3,5	0	01.01.2019	0				б	ж
31. Куц Андрей Владимирович	5310М	1	4	0	01.01.2020	0				б	м
32. Позовик Евгений Олегович	5310М	1	5	0	27.02.2025	0				б	м
33. Матвеев Николай Валентинович	5310М	1	5	0	01.01.2022	0				б	м
34. Новиков Дмитрий Вадимович	5310М	1	5	0	01.01.2023	0				б	м
35. Палушин Андрей Сергеевич	5310М	1	3,5	0	01.01.2024	0				б	м

Показать долги на: 30.03.2014

Рис. 1. Главное окно приложения «АРМ декана»

В главном окне АРМ декана (см. рис. 1) можно произвести фильтрацию студентов; просмотреть и отсортировать сводную таблицу и график академических задолженностей, анкетные данные об обучающихся, в том числе историю приказов; добавить данные и результаты аттестации и сессии; заполнить форму движения контингента студентов и увидеть академические задолженности не только на текущую дату, но и на выбранную дату в прошлом. Все операции, выполняемые в приложении, возможны в микро и макро вариантах.

Для анализа академических задолженностей, имеются сводная таблица (см. рис. 1) с общей информацией об обучающихся, количеством академических задолженностей, датой последней сдачи; и график для более наглядного отображения интегрального взгляда на успеваемость студентов [4].

Мониторинг успеваемости студентов по дисциплинам позволяет видеть успеваемость всех студентов по выбранной дисциплине даже если они учатся в разных группах [5]. Это необходимо для контроля усвоения знаний студентами и проверки правильности организации пересдач.

Кроме того, наше приложение позволяет автоматически формировать статистические отчеты разного вида. Например, «Сводка движения контингента студентов» (рис. 2), которая составляется для периодической отчетности ВУЗа перед министерством образования и науки Российской Федерации и выгружается в Microsoft Excel [6].

контракт

Сводка движения контингента студентов за январь 2012 года																												
факультет	курс	прибыло							выбыло							перевод							окончили и сдали госэж.	всего выбыло	состоит к концу месяца	по полу		
		состоит на 1 число	принято	восстановлено	перевод из других ВУЗов	из филиалов	перевод с др. факультетов	перевод с курса на курс	перевод с бюдж. формы обуч. на контр. формы обуч. на контр.	итого	неуспевающие	болезнь	нарушение дисциплины	нарушение условий договора	по собственному желанию	прочие причины	итого	перевод с контр. формы обуч. на бюдж.	с курса на курс	на другой факультет	в другие ВУЗы	итого				в т.ч. в отпуску	мужчины	женщины
а	1	13								0							0					0		0	13		13	0
	2	28								0							0					0		0	28		28	2
	3	43								0	5						5					0		5	38	1	33	5
	4	65								0	1						1			3		3		4	61		51	10
	5	50						3		3							0					0		0	53		48	5
	6	0								0							0					0		0	0		0	0
всего		199	0	0	0	0	0	3	0	3	6	0	0	0	0	0	6	0	3	0	0	3	0	9	193	1	171	22

бюджет

факультет	курс	прибыло							выбыло							перевод							окончили и сдали госэж.	всего выбыло	состоит к концу месяца	по полу		
		состоит на 1 число	принято	восстановлено	перевод из других ВУЗов	из филиалов	перевод с др. факультетов	перевод с курса на курс	перевод с контр. формы обуч. на бюдж. формы обуч. на контр.	итого	неуспевающие	болезнь	нарушение дисциплины	нарушение условий договора	по собственному желанию	прочие причины	итого	перевод с бюдж. формы обуч. на контр.	с курса на курс	на другой факультет	в другие ВУЗы	итого				в т.ч. в отпуску	мужчины	женщины
а	1	136								0							0					0		0	136		108	28
	2	127								0					2		2					0		2	125	1	90	35
	3	90								0							0					0		0	90	2	66	24
	4	83								0							0					0		0	83	2	53	30
	5	97								0							0					0		0	97		62	35
	6	25								0							0					0		0	25		16	9
всего		558	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	0	0	0	0	0	0	2	556	5	395	161

Секретарь факультета № 5

Е.А. Копеева

749

Секретарь факультета № 5

Е. А. Колеватых

749

Рис. 2. Сводка движения контингента студентов

Для обеспечения безопасности, хранимой в программе информации используется парольная защита и ведется история изменений данных, таких как движение контингента студентов. А так же, внедрена система автоматического поиска ошибок.

Приложение «Автоматизированное рабочее место декана» позволяет повысить эффективность и оперативность деятельности сотрудников деканата. Оно протестировано в институте информационных систем и защиты информации Санкт-Петербургского государственного университета аэрокосмического приборостроения. В настоящий момент программа находится на регистрации фонде алгоритмов и программ.

Список литературы

1. Ушаков, В. А. АРМ для учебной части / В. А. Ушаков // Шестьдесят шестая международная студенческая научная конференция ГУАП : сб. докл. : в 2 ч. – СПб. : СПбГУАП, 2013. – Ч. I. Технические науки. – С. 342–346.
2. Ушаков, В. А. Разработка автоматизированного рабочего места декана / В. А. Ушаков // Решетневские чтения : материалы XVII Междунар. науч. конф., посвящ. памяти генер. конструктора ракет.-космич. систем акад. М. Ф. Решетнева (12–14 нояб. 2013 г., Красноярск) : в 2 ч. / под общ. ред. Ю. Ю. Логинова. – Красноярск : Сиб. гос. аэрокосмич. ун-т, 2013. – Ч. 2. – С. 263–265.
3. Корняков, В. Н. Программирование документов и приложений MS Office в Delphi / В. Н. Корняков. – СПб. : БХВ-Петербург, 2006. – 496 с.
4. Ушаков, В. А. Мониторинг успеваемости студентов с использованием автоматизированного рабочего места учебной части / В. А. Ушаков, И. У. Бхаттачарджи // XL Гагаринские чтения : науч. тр. Междунар. молодежной науч. конф., 7–11 апреля : в 9 т. – М. : МАТИ, 2014. – Т. 4. – С. 246–248.
5. Ushakov, V. Implementation of program part at automated workplace for a teaching department / V. Ushakov // Proceedings of 15th conference of Open Innovations Association FRUCT, St-Petersburg, April, 2014. – St-Petersburg, 2014. – P. 214–220.
6. Ушаков, В. А. Программно-аппаратное автоматизированное рабочее место декана / В. А. Ушаков // Шестьдесят седьмая международная студенческая научная конференция ГУАП : сб. докл. : в 2 ч. – СПб. : ГУАП, 2014. – Ч. I. Технические науки. – С. 220–224.

ПАРАДИГМА НЕРАВНОВЕСНОГО ПРОГРАММИРОВАНИЯ И ДРУГИЕ СОВРЕМЕННЫЕ ПАРАДИГМЫ

Формулируются основные положения парадигмы неравновесного программирования, проводится сравнительный анализ парадигмы неравновесного программирования с другими современными парадигмами.

Современное программное обеспечение должно не только выполнять полезную вычислительную работу, направленную на решение общесистемных задач или удовлетворение потребностей пользователя, но и поддерживать свою работоспособность в условиях многопользовательской многозадачной среды. Характерными особенностями таких сред является невозможность перебора пространств возможных состояний и событий, а также не всегда однозначная взаимосвязь между событиями и переходами в новые состояния ввиду наличия предыстории функционирования вычислительной среды и ее отдельных компонентов. Поэтому традиционные детерминированные методы задания реакции программы на внешние воздействия среды оказываются неэффективными, так как не всегда адекватны, а тем более не обеспечивают адаптации программ к конкретной изменяющейся вычислительной среде, накоплению и использованию опыта. Тем более это верно для распределенных вычислительных сред, которые могут строиться на базе систем с большой территориальной протяженностью, например, глобальных компьютерных сетей.

Таким образом, на повестке дня стоит вопрос о переходе к программным системам, способным к самоадаптации, самоменеджменту, самовосстановлению или даже самопрограммированию. Соответственно, появляется необходимость в концепциях, в рамках которых будут предложены методы и средства создания таких программ.

Предложенная автором в работах [1–2] парадигма неравновесного программирования может быть сформулирована в виде совокупности следующих основных положений:

- программа представляет собой динамический объект переменной структуры в виде непрерывно изменяющейся совокупности отдельных независимых программных единиц, участвующих в непрерывных процессах распада-образования связей друг с другом;
- в свою очередь, каждая независимая программная единица (цифровой организм) является результатом непрерывного образования-распада связей между глобальными данными и программными компонентами, которые также непрерывно изменяются или заменяются;
- программа функционирует в условиях замкнутой среды (искусственного мира) и может быть результатом эволюции некоторой первичной зародышевой структуры, например, единственного исходного цифрового организма, в соответствии с правилами искусственного мира;
- целью программы является как можно более дальний уход от модельного аналога термодинамического равновесия и отсутствия свободной энергии, заданного как отсутствие ресурса доступного процессорного времени, то есть как можно

дольшее функционирование и выполнение полезной работы, за которую начисляются баллы доступного процессорного времени.

Концептуально парадигма неравновесного программирования происходит от моделей теории искусственной жизни, в частности, искусственной химии [3], моделей молекулярной биологии, а также генетических алгоритмов и генетического программирования. Технологическим базисом для парадигмы неравновесного программирования являются объектно-ориентированное, агент-ориентированное и компонентное программирование.

В некоторых аспектах концептуально близкими к данной парадигме можно считать порождающее (*generative programming*) [4] и ментальное программирование (*intentional programming*) [4], а также искусственные иммунные сети (*artificial immune systems*) [5–7]. Далее будет проведен сравнительный анализ этих парадигм и парадигмы неравновесного программирования.

Сущность порождающего программирования [4] состоит в том, что программа может быть сгенерирована на основе множества имеющихся классов, шаблонов (паттернов) проектирования и типовых конструктивных образцов сходных по целевому назначению приложений. Такой подход технологичен, обеспечивает возможность использования уже имеющихся удачных решений, способствует повышению скорости и качества разработки. Однако сгенерированная программа остается неизменной, она неспособна к адаптации.

Неравновесное программирование также исходит из того, что программа (программная система) может быть сгенерирована на основании некоторых родовых образцов с учетом наилучших предыдущих решений при использовании компонентного подхода. Однако при этом вся сгенерированная программная система или отдельная независимая программная единица в ее составе не являются точной копией исходного образца, хотя и наследуют генетически его основные свойства. Программа в целом и программные единицы по отдельности существуют как непрерывно изменяющиеся совокупности, поэтому они сохраняют способность к самообучению и адаптации, а фактически – к самопрограммированию. С другой стороны, программа и составляющие ее отдельные программные единицы практически никогда не находятся в фиксированном («твердом») состоянии.

Парадигма ментального программирования [4] близка к неравновесному программированию тем, что согласно этой парадигме программа описывается не в терминах машино-ориентированного языка программирования, а в терминах *намерений* (*intentions*) программиста. Таким образом, программа задается на уровне семантики предметной области. Затем на основе выделенных намерений может быть сгенерирован исходный текст на каком-либо распространенном языке программирования высокого уровня.

В случае неравновесного программирования семантика предметной области также имеет первостепенное значение, однако вместо намерений, которые остаются неизменными, для задания программы используется совокупность зародышевой структуры и правил, которым подчиняется существование искусственного мира и населяющих его цифровых организмов. Благодаря этому зародышевая структура может развиваться в различных направлениях, порождая разные программы, которые в свою очередь могут изменяться в соответствии с изменениями среды.

Как и ментальное программирование, неравновесное программирование использует некоторые принципы, характерные для баз данных. В ментальном программировании каждый объект (переменная или оператор) снабжаются идентификатором, что позволяет провести «нормализацию» кода и избежать избыточности.

В неравновесном программировании независимые программные единицы – цифровые организмы – являются результатом сборки данных и программных компонентов, которые могут храниться в базе данных. В свою очередь, все программные решения могут сохраняться в базе данных как последовательности номеров компонент, участвующих в сборке. Одновременно с этим базы данных могут выступать и как исходные данные, подлежащие обработке множеством цифровых организмов. Так проявляется характерная для неравновесного программирования двойственность кода и данных.

Так же, как и неравновесное программирование, концепция искусственных иммунных систем [5–7] исходит из того, что программы могут порождаться в некоторой среде в результате процессов самопрограммирования. Самопрограммирование возможно благодаря тому, что исходные и результирующие программы представляют собой текстовые строки, которые могут видоизменяться в соответствии с заданными грамматиками. Такой подход роднит искусственные иммунные системы с классическими моделями искусственной химии [3]. Трудность, с которой неизбежно приходится сталкиваться при подобном подходе, состоит в том, что не исключается возможность появления программных последовательностей с нежелательными свойствами. Во избежание этого в модель приходится вводить дополнительные служебные программы, такие как антитела (*antibodies*), В-квинны (*B-quinines*), Т-помощники (*T-helper fraglets*) и супрессоры (*suppressor fraglets*) [6]. В результате модель оказывается слишком сложной.

В отличие от искусственных иммунных систем, в неравновесном программировании проверка программного кода на допустимость выполняется гораздо проще, путем анализа кодирующей числовой последовательности, или цифровой ДНК. Проверка может проводиться разными способами: на наличие или отсутствие заданного значения в указанной позиции последовательности, на сочетание значений, находящихся в разных позициях последовательности, и т.д. Для задания правил, которым должна или не должна удовлетворять допустимая последовательность, могут использоваться регулярные выражения.

Таким образом, парадигма неравновесного программирования соответствует современным тенденциям развития технологий программирования для распределенных систем и находится в тесной связи с многими другими парадигмами и концепциями, обладая рядом преимуществ по сравнению с ними.

Список литературы

1. Кольчугина, Е. А. Модель эволюционирующего программного обеспечения / Е. А. Кольчугина // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2006. – № 6 (27). – С. 78–86.
2. Кольчугина, Е. А. Неравновесное программирование / Е. А. Кольчугина // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2009. – № 3 (11). – С. 25–31.
3. Fontana, W. Algorithmic Chemistry / W. Fontana // Artificial Life II, SFI Studies in the Sciences of Complexity / ed. by C. G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen. – Redwood City, CA : Addison-Wesley, 1991. – Vol. X. – P. 159–209.
4. Чарнецки, К. Порождающее программирование: методы, инструменты, применение. Для профессионалов / К. Чарнецки, У. Айзенекер. – СПб. : Питер, 2005. – 731 с.
5. Fraglets Home Page. – URL: <http://www.fraglets.net> (дата обращения 03.04.2014).

6. Чарнецки, К. Programming by Equilibria / К. Чарнецки, У. Айзенекер // 15th Kolloquium Programmiersprachen und Grundlagen der Programmierung (KPS 2009), October 2009 / Technische Universität Wien. – 2009. – Vol. 2009-X-2. – P. 37–46.

7. Schreckling, D. An Artificial Immune System Approach for Artificial Chemistries Based on Set Rewriting / D. Schreckling, T. Marktscheffel // Artificial Immune Systems. Proceedings of 9th International Conference, ICARIS 2010. Edinburgh, UK, July 26–29, 2010 – Edinburgh, 2010. Lecture Notes in Computer Science. – Vol. 6209. – Berlin-Heidelberg : Springer Verlag, 2010. – P. 250–263.

Е. А. Кольчугина

Россия, Пенза, Пензенский государственный университет

ПАРАДИГМА НЕРАВНОВЕСНОГО ПРОГРАММИРОВАНИЯ И ИНДУСТРИЯ РАЗРАБОТКИ ПРОГРАММ

Выделяются и рассматриваются изменения, которые может претерпеть индустрия разработки программного обеспечения в случае внедрения принципов парадигмы неравновесного программирования.

Современная индустрия программирования и информационных технологий функционирует таким образом, чтобы заставить пользователя постоянно приобретать новое программное обеспечение. При этом не всегда речь идет о программных продуктах принципиально нового назначения, решающих новые задачи или удовлетворяющих новые информационные потребности пользователя. Как правило, потребителю предлагаются новые версии уже существовавших ранее продуктов. При этом функциональные преимущества новых версий перед старыми не всегда очевидны для пользователя. Однако установка новых версий с более высокими программно-аппаратными требованиями может в свою очередь потребовать дополнительных затрат, связанных с приобретением сопутствующего системного программного обеспечения, а иногда и аппаратных средств.

Подобный поход вызывает также проблемы, связанные с несовместимостью отдельных прикладных программ, выполняющихся на одной и той же программно-аппаратной платформе. Эти проблемы являются результатом несогласованности действий разработчиков и отсутствия у них полной информации о структуре и функциях операционной системы, что особенно характерно для операционных систем с закрытой архитектурой. Перечисленные проблемы характерны не только для коммерческих, но и для свободно распространяемых прикладных программных продуктов.

Традиционным решением проблем версионной несовместимости и способом устранения ошибок на стадии эксплуатации программного обеспечения является принуждение пользователя постоянно скачивать и устанавливать обновления имеющихся у него программ. В результате пользователь должен тратить дополнительное время на то, чтобы отслеживать актуальность версий программного обеспечения и поддерживать целостность программно-аппаратной среды. Помимо временных затрат, подобный процесс требует еще либо наличия у пользователя некоторых профессиональных навыков, либо регулярного обращения к специалистам, ими обладающими.

Отказ следовать общей тенденции и непрерывно следить за актуальностью версий приводит к тому, что через некоторое время пользователь может лишиться возможности использовать и воспроизводить тот или иной информационный контент, так как параллельно с изменением версий программного обеспечения происходит изменение формата представления контента.

Таким образом, доступ к информационным сервисам и компьютерным технологиям, уже очевидно необходимый в современном обществе для получения образования, поиска работы, пользования госуслугами, для улучшения качества жизни на бытовом уровне, может оказаться недоступным малоимущим слоям населения в виду высокой стоимости приобретения и владения.

Концепция неравновесного программирования, предложенная в настоящей работе, помогает хотя бы отчасти решить перечисленные проблемы.

В случае неравновесного программирования программа представляет собой не статический, а динамически устойчивый объект, который собирается на основе программных компонентов, находящихся в локальной памяти автономного компьютера или вычислительного узла сети. Пользователь может приобрести целостный набор базовых компонентов и неограниченно долго его использовать. Сама программа представляет собой тактированно реактуализируемый результат сборки этих компонентов на основе цифровой ДНК. При этом основное значение имеют типы компонентов и связанная с ними семантика, задаваемые номерами локусов, а не номера версий компонентов, задаваемые значениями локусов. Правила задания значений по умолчанию позволяют указывать альтернативные номера версий компонентов, если актуальные версии в системе отсутствуют. Это, с одной стороны, хотя бы отчасти снимает проблему устаревания и несовместимости версий, а с другой стороны, продлевает срок жизни программных решений, так как они представлены уже на более высоком концептуальном уровне.

Изменяются также основные идеи, связанные с понятиями создания программы, ее приобретения и распространения.

Сама программа будет представлять собой совокупность рецепта ее сборки, правил организации среды, подходящей для программы, и правил развития программы в этой среде. Если создание правил организации среды и развития программы может представлять сложность для неквалифицированных пользователей и начинающих программистов, то создание сборок может принять массовый характер, с возможностью свободного обмена сборками между их создателями и пользователями. Удачные новые решения на уровне сборок могут быть получены даже начинающими разработчиками, находящимися в стесненных экономических условиях.

В целом, процесс прикладного программирования и администрирования прикладных систем будет напоминать животноводство и растениеводство. Как и в этих областях деятельности, в прикладном программировании важную роль будет играть формирование подходящей среды, подбор генетического материала для исходных популяций и поддержание существования искусственного биоценоза.

На долю профессиональных специалистов в области программной инженерии достанется преимущественно разработка сред, библиотек компонентов и совокупности правил развития программ. Именно эти составляющие будут вносить основной вклад в стоимостные характеристики программного обеспечения.

Применение парадигмы неравновесного программирования создает потребность в развитии в таких областях математики, как теория динамических систем, теория хаоса, качественная теория дифференциальных уравнений, так как на основе

результатов, достигнутых в рамках этих направлений, могут быть получены новые математические модели, востребованные при реализации парадигмы.

С другой стороны, системы неравновесного программирования сами могут стимулировать исследования в перечисленных областях математики, так как позволяют проводить вычислительные эксперименты с наглядной демонстрацией результатов.

Е. А. Кольчугина

Россия, Пенза, Пензенский государственный университет

ИДЕНТИФИКАЦИЯ ЦИФРОВЫХ ОРГАНИЗМОВ

Предложен способ идентификации цифровых организмов путем генерации уникальных имен, обладающих семантикой.

Необходимость идентификации цифровых организмов связана с тем, что они способны эволюционировать в процессе своего жизненного цикла. Решение задачи исследования эволюции отдельного организма требует наличия средств, позволяющих отыскать информацию о конкретном организме. К этим средствам относятся в первую очередь средства работы с темпоральными базами данных, в которых фиксируются цифровые ДНК отдельных организмов, снабженные их идентификаторами и временными метками.

Ранее уже были введены некоторые характеристики, позволяющие идентифицировать организм на уровне общего генезиса и родовидовой принадлежности:

- идентификаторы разработчика и владельца организма;
- идентификатор вида цифрового организма;
- номер слоя, к которому принадлежит организм.

Однако необходим и персональный идентификатор организма, уникальный хотя бы в пределах вида.

В экспериментах с искусственными мирами цифровых организмов, рассмотренных далее, была использована система идентификации, отчасти вдохновленная произведением С. Лема [1].

В большинстве описываемых далее экспериментов особи каждого из видов делились на два пола, и каждая особь, порожденная в процессе моделирования, имела двух родителей. Имена особей формировались путем конкатенации имен родителей с прибавлением дополнительного числового идентификатора. Пара предков имела идентификаторы «A1» («самец») и «E1» («самка»). Имена условных «самцов» начинались с идентификатора «отца», за которым следовал идентификатор «матери», буква «s» («son») и номер потомка у данного «отца». Имена условных «самок» начинались с идентификатора «матери», за которым следовал идентификатор «отца», буква «d» («daughter») и номер потомка у данного «отца». Такая система наименований гарантировала уникальность идентификаторов, но приводила к быстрому росту длины имен, что вызывало замедление в обработке идентификаторов.

В связи с этим был введен дополнительный идентификатор: номер поколения. Номер поколения для потомка определялся как максимум из номеров поколений

его родителей. Если номер поколения потомка делился на 10 без остатка, то имя потомку давалось по укороченной схеме. Для «самцов» идентификатор начинался с буквы «А» за которой следовал номер поколения «отца», а затем, после разделителя, номер поколения «матери». Для «самок» применялась аналогичная схема, только идентификатор начинался с буквы «Е».

В ходе экспериментов при использовании данной схемы идентификации совпадений имен отмечено не было. Кроме того, идентификаторы, построенные с использованием подобной схемы, сами по себе являются информативными, указывая на «пол» цифрового организма, его приблизительный возраст и время рождения, а также родственные связи с другими организмами.

Список литературы

1. Лем, С. Не буду служить // Библиотека XXI века : сб.: пер. с польск. / С. Лем. – М. : АСТ, 2004. – 602 с.

СОДЕРЖАНИЕ

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ И ПАРАЛЛЕЛЬНЫЕ СИСТЕМЫ

Вашкевич Н. П., Калиниченко Е. И.

ФОРМАЛЬНЫЙ ЯЗЫК НА ОСНОВЕ КОНЦЕПЦИИ
НЕДЕТЕРМИНИЗМА ДЛЯ ОПИСАНИЯ АЛГОРИТМОВ
УПРАВЛЕНИЯ ПРОЦЕССАМИ И РЕСУРСАМИ
В ПАРАЛЛЕЛЬНЫХ СИСТЕМАХ ОБРАБОТКИ ИНФОРМАЦИИ.....3

Вашкевич Н. П., Калиниченко Е. И.

ПРОЕКТИРОВАНИЕ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ
В ЗАДАЧАХ ИДЕНТИФИКАЦИИ 6

Шалагин С. В.

ОЦЕНКА СЛОЖНОСТИ РАСПРЕДЕЛЕННОГО ВЫЧИСЛЕНИЯ
НЕЛИНЕЙНОЙ ПОЛИНОМИАЛЬНОЙ ФУНКЦИИ НАД ПОЛЕМ $GF(2^k)$
НА МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ 9

Федюнин Р. Н., Войнов А. С., Сенокосов И. В.

ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ АЛГОРИТМОВ
УМНОЖЕНИЯ БОЛЬШИХ ДАННЫХ С ПОМОЩЬЮ МАРКОВСКИХ
ПРОЦЕССОВ 13

Федюнин Р. Н.

СИСТОЛИЧЕСКИЕ СТРУКТУРЫ СРАВНИТЕЛЬНОГО АНАЛИЗА
МАССИВА И ЭТАЛОНА 18

АППАРАТНО-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СИСТЕМ УПРАВЛЕНИЯ

Чулков В. А., Нестеренко С. А.

ЛОКАЛЬНАЯ ФАЗОВАЯ ИНТЕРПОЛЯЦИЯ В ТЕХНИКЕ
ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ 22

Чулков В. А., Нестеренко С. А.

НЕПРЕРЫВНАЯ ФАЗОВАЯ ИНТЕРПОЛЯЦИЯ В ТЕХНИКЕ
ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ 24

Уваров Д. А.

СТАНДАРТ ИЕС 61499, ФУНКЦИОНАЛЬНЫЕ БЛОКИ ИЕС 61499,
РАСПРЕДЕЛЕННАЯ МОДЕЛЬ СИСТЕМЫ.....28

Тумасов С. В., Таранцев Е. К., Бычков А. С.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ SDR-РАДИОСРЕДСТВОМ 32

Смагин М. С.

ПЕЧАТНАЯ ЭЛЕКТРОНИКА КАК ПЕРСПЕКТИВНОЕ
АППАРАТНО-ТЕХНОЛОГИЧЕСКОЕ НАПРАВЛЕНИЕ 35

Скопинцев Р. Г.

АППАРАТНАЯ РЕАЛИЗАЦИЯ НА ПЛИС АССОЦИАТИВНОЙ
ПАМЯТИ С ПОИском БЛИЖАЙШИХ ПОИСКОВЫХ СЛОВ 39

Рыбаков А. В.	
УЛУЧШЕНИЕ ОЦЕНКИ СЛОЖНОСТИ АСИМПТОТИЧЕСКИ ОПТИМАЛЬНЫХ ПО НАДЕЖНОСТИ КЛЕТОЧНЫХ СХЕМ	42
Пикулева Н. И.	
ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В CASE-СРЕДСТВЕ SILVERRUN	45
Пащенко В. В., Пащенко Д. В., Кучин А. В.	
ДАТЧИКИ СИСТЕМ КОНТРОЛЯ И УПРАВЛЕНИЯ.....	48
Трокоз Д. А., Максимова К. С., Бальзанникова Е. А.	
СПОСОБ ТРАНСЛЯЦИИ ПОСЛЕДОВАТЕЛЬНЫХ ФУНКЦИОНАЛЬНЫХ СХЕМ В ДИАГРАММЫ ФУНКЦИОНАЛЬНЫХ БЛОКОВ	51
Бальзанникова Е. А., Трокоз Д. А., Пащенко Д. В.	
АЛГОРИТМ ТРАНСЛЯЦИИ ИЗ ФУНКЦИОНАЛЬНОГО ЯЗЫКА «Т» В ПРОГРАММНЫЕ МОДУЛИ НА ЯЗЫКЕ «С»	54
Барабаш К. О., Чечель П. К., Пащенко Д. В.	
КЛАССИФИКАЦИЯ ДАТЧИКОВ ДЛЯ ЭЛЕКТРОННОГО КАТАЛОГА.....	57
Пашков Н. А.	
МНОГОКРИСТАЛЬНЫЕ СИСТЕМЫ НА БАЗЕ ПЛИС. ОПТИМИЗАЦИЯ АЛГОРИТМА ПРОГРАММИРОВАНИЯ ПОДЧИНЕННЫХ ПЛИС ПО ИНТЕРФЕЙСУ SPI.....	60
Мартьяшин Г. В., Максимова К. С., Пащенко Д. В.	
ГЕНЕРАЦИЯ КОДА ДЛЯ РАСПРЕДЕЛЕННОЙ АРХИТЕКТУРЫ С ПРИМЕНЕНИЕМ МНОГОУРОВНЕВОЙ ТРАНСЛЯЦИИ	64
Алеева А. Р., Пащенко Д. В., Бычков А. С.	
ОБЪЕКТИВНЫЙ КОНТРОЛЬ СИСТЕМ УПРАВЛЕНИЯ ОБЪЕКТАМИ НАЗЕМНОЙ ИНФРАСТРУКТУРЫ РАКЕТНО-КОСМИЧЕСКОЙ ТЕХНИКИ.....	67
Башев А. С., Афонин А. Ю.	
ПРЕИМУЩЕСТВА ИСПОЛЬЗОВАНИЯ ЯЗЫКА LUA ДЛЯ РАЗРАБОТКИ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ.....	71
Бондаренко Л. Н.	
О СЖАТИИ ТЕЛЕМЕТРИЧЕСКОЙ ИНФОРМАЦИИ ДЛЯ ЕЕ ОБРАБОТКИ МЕТОДАМИ ПРОНИ И РУТИСХАУЗЕРА	74
Каргин С. П.	
О СИНТЕЗЕ СХЕМ ИЗ НЕНАДЕЖНЫХ ЭЛЕМЕНТОВ В P_4	78
Гурин Е. И., Скопинцев Р. Г.	
ПОСТРОЕНИЕ ВСТРАИВАЕМЫХ СИСТЕМ С ПРОЦЕССОРНЫМИ ЯДРАМИ НА ПЛИС ФИРМЫ XILINX	80
Клименко А. В.	
АЛГОРИТМ АВТОМАТИЗИРОВАННОЙ РАЗРАБОТКИ ЭЛЕКТРОННЫХ УСТРОЙСТВ НА БАЗЕ ПОЛУЗАКАЗНЫХ ИНТЕГРАЛЬНЫХ СХЕМ НА ОСНОВЕ ПЛИС ПРОТОТИПИРОВАНИЯ	83
Турчак О. А., Князев В. Н.	
МОДИФИКАЦИЯ АЛГОРИТМА КОДИРОВАНИЯ РЕЧИ MELP	86
Кутузов В. В.	
СИСТЕМА ВЕРИФИКАЦИИ АЛГОРИТМОВ УПРАВЛЕНИЯ, ПРЕДСТАВЛЕННЫХ НА ЯЗЫКЕ СНДА	89

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ

Волчихин В. И., Зинкин С. А.

ОПЕРАЦИОННАЯ СЕМАНТИКА РАЗВЕРТЫВАНИЯ
МОБИЛЬНЫХ ОБЪЕКТОВ В ВИРТУАЛИЗИРОВАННЫХ СЕТЯХ
ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ 92

Якупов А. Ю., Малкин А. А., Колесникова Ю. С.

АППАРАТНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА
СЖАТИЯ ИЗОБРАЖЕНИЙ БЕЗ ПОТЕРЬ
С ЭФФЕКТИВНЫМ ИСПОЛЬЗОВАНИЕМ ПАМЯТИ,
ОРИЕНТИРОВАННАЯ НА КОСМИЧЕСКОЕ ПРИМЕНЕНИЕ 97

Башев А. С., Афонин А. Ю.

МЕТОДЫ ХРАНЕНИЯ И СОЗДАНИЯ СМЕШАННОГО ЛАНДШАФТА 100

Убиенных А. Г., Убиенных Г. Ф.

ОЦЕНКА ПОМЕХОУСТОЙЧИВОСТИ АЛГОРИТМОВ
ДЕКОДИРОВАНИЯ КАСКАДНОЙ СИСТЕМЫ КОДИРОВАНИЯ
НА ОСНОВЕ СВЕРТОЧНЫХ КОДОВ 104

Архипов Л. В.

СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 107

Савченко В. В., Васильев Р. А.

АВТОМАТИЧЕСКАЯ ОЦЕНКА КАЧЕСТВА РЕЧИ
ПО КРИТЕРИЮ МИНИМУМА ТРЕБУЕМОЙ ИЗБЫТОЧНОСТИ
РЕЧЕВОГО СИГНАЛА..... 110

Войнова Е. А., Токарев А. Н.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ И ВЫБОР СУБД
ДЛЯ СИСТЕМЫ УПРАВЛЕНИЯ НАЗЕМНОЙ ИНФРАСТРУКТУРОЙ
РАКЕТНО-КОСМИЧЕСКОЙ ТЕХНИКИ..... 113

Токарев А. Н., Войнова Е. А.

ХРАНЕНИЕ ДРАЙВЕРОВ ДЛЯ ПОДКЛЮЧАЕМЫХ МОДУЛЕЙ
В СИСТЕМЕ УПРАВЛЕНИЯ НАЗЕМНОЙ ИНФРАСТРУКТУРОЙ
РАКЕТНО-КОСМИЧЕСКОЙ ТЕХНИКИ..... 117

Горбачев Д. В.

МЕТОДИЧЕСКИЙ ПОДХОД К РАЗРАБОТКЕ
ИНФРАСТРУКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ
ЛЕЧЕБНО-ДИАГНОСТИЧЕСКОГО ПРОЦЕССА
ПРИ ОКАЗАНИИ МЕДИЦИНСКОЙ ПОМОЩИ..... 120

Графов О. Б.

МЕТОД И АЛГОРИТМЫ МЯГКОГО ДЕКОДИРОВАНИЯ КОДОВ
РИДА-СОЛОМОНА ДЛЯ КОРРЕКЦИИ ОШИБОК
В АРХИВНОЙ ОПТИЧЕСКОЙ ПАМЯТИ 125

Егоров С. И., Кривонос А. В.

АЛГОРИТМЫ КОРРЕКЦИИ ОШИБОК
ДЛЯ ОПТИЧЕСКИХ ДИСКОВ DVD..... 128

Жерновой С. А.

ПРОТОКОЛЫ МАРШРУТИЗАЦИИ БЕСПРОВОДНЫХ МОБИЛЬНЫХ
СЕТЕЙ (MANET)..... 133

Карамышева Н. С.

ОРГАНИЗАЦИЯ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ В ВЫЧИСЛИТЕЛЬНОЙ СЕТИ
С ТОПОЛОГИЕЙ ТИПА «ЛОГИЧЕСКОЕ КОЛЬЦО» 136

Карпов А. Е.

РАСШИРЕНИЕ ВОЗМОЖНОСТЕЙ УНИФИКАЦИИ ФУНКЦИЙ
НА ОСНОВЕ ЭКВИВАЛЕНТНЫХ КОНСТРУКЦИЙ ЦИКЛОВ 142

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И РАСПОЗНАВАНИЕ ОБРАЗОВ

Федотов Н. Г., Сёмов А. А.

К ВОПРОСУ ИЗВЛЕЧЕНИЯ ПАРАМЕТРОВ 3D ОБЪЕКТОВ
ИЗ РЕЗУЛЬТАТОВ СКАНИРОВАНИЯ 146

Стас Т. Т., Козлов А. Ю.

ПРИМЕНЕНИЯ ФУНКЦИИ МНОГОМЕРНОГО НОРМАЛЬНОГО
РАСПРЕДЕЛЕНИЯ В ПОСТРОЕНИИ АДАПТИВНОГО АЛГОРИТМА
РАСПОЗНАВАНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ФРАЗЫ 149

Бердичевский Е. Г., Вознюк Д. В.

ПЕРСПЕКТИВЫ ИНФОРМАЦИОННО-ИНТЕЛЛЕКТУАЛЬНЫХ
СИСТЕМ В ТЕХНИЧЕСКОЙ ЭСТЕТИКЕ 152

Орешкин К. А.

ИСПОЛЬЗОВАНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ БАЗЫ ДАННЫХ
INTERSYSTEMS CASHE В КАЧЕСТВЕ ПЛАТФОРМЫ
ДЛЯ РЕАЛИЗАЦИИ АКТИВНОЙ БАЗЫ ДАННЫХ 155

Зинкин С. А., Шеланков О. Е.

ОСНОВНЫЕ ПОЛОЖЕНИЯ
ПО ФОРМАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ДИНАМИЧЕСКИХ
ПРОЦЕССОВ НА ОСНОВЕ СОБЫТИЙНО-ИЕРАРХИЧЕСКОГО
ФРЕЙМОВОГО МЕТОДА ПРЕДСТАВЛЕНИЯ ЗНАНИЙ 158

Лезин И. А., Маркелов Д. Е.

ИЗВЛЕЧЕНИЕ ЗНАНИЙ
ИЗ БОЛЬШИХ МАССИВОВ ДАННЫХ 160

Мангалова Е. С.

ПОСТРОЕНИЕ АНСАМБЛЯ НЕПАРАМЕТРИЧЕСКИХ ОЦЕНОК
С ИЗМЕНЯЮЩИМИСЯ ПАРАМЕТРАМИ РАЗМЫТОСТИ
ДЛЯ РЕШЕНИЯ ЗАДАЧ ВОССТАНОВЛЕНИЯ РЕГРЕССИИ
И РАСПОЗНАВАНИЯ ОБРАЗОВ 163

Бобырь М. В., Кулабухов С. А.

ПРОГНОЗИРОВАНИЕ РЕЗУЛЬТАТА ЧИСЛЕННЫХ ВЫЧИСЛЕНИЙ
В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ 167

Михайлов Э. В., Богуш Р. П.

ОБРАБОТКА ИЗОБРАЖЕНИЙ ПЛАТЕЖНЫХ КВИТАНЦИЙ
ПРИ АВТОМАТИЗИРОВАННОМ ВВОДЕ ДАННЫХ
С ПРИМЕНЕНИЕМ МОБИЛЬНЫХ УСТРОЙСТВ 169

Вершинин И. С.

ПОМЕХОУСТОЙЧИВОСТЬ АНАЛИЗА
АССОЦИАТИВНО-ЗАЩИЩЕННЫХ НЕСЖАТЫХ
БИНАРНЫХ СЦЕН 173

Горбаченко В. И., Жуков М. В. ИСПОЛЬЗОВАНИЕ СЕТЕЙ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ ДЛЯ ИДЕНТИФИКАЦИИ СВОЙСТВ СРЕДЫ ПО РЕЗУЛЬТАТАМ ДИСКРЕТНЫХ ИЗМЕРЕНИЙ	177
Слепцов Н. В. ВЫБОР ЭФФЕКТИВНОЙ СХЕМЫ КОДИРОВАНИЯ ПРИ ЭВОЛЮЦИОННОМ ПОИСКЕ	180
Бойков И. В., Руднев В. А., Бойкова А. И., Баулина О. А., Столяров Р. В. ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ НА НЕЙРОННЫХ СЕТЯХ ХОПФИЛДА	183

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ В ПРОИЗВОДСТВЕ И УПРАВЛЕНИИ

Drozдов D., Dubinin V., Patil S., Vyatkin V. PLANT ABSTRACTION FOR IEC 61499 CLOSED-LOOP MODEL CHECKING	194
Косников Ю. Н. РЕКОНСТРУКЦИЯ ПРОСТРАНСТВЕННОГО РЕЛЬЕФА НА ОСНОВЕ ЦВЕТОВОГО КОДИРОВАНИЯ ИЗОПОВЕРХНОСТЕЙ.....	202
Тараник М. А., Копаница Г. Д. НЕОБХОДИМОСТЬ ФОРМИРОВАНИЯ НОВЫХ ИНТЕЛЛЕКТУАЛЬНЫХ ПОДХОДОВ ДЛЯ РЕШЕНИЯ УПРАВЛЕНЧЕСКИХ ЗАДАЧ В ЗДРАВООХРАНЕНИИ	205
Николаева Е. А., Советкина Г. И., Дубравин А. В. РАЗРАБОТКА ПРИЛОЖЕНИЯ ИМПОРТА ДАННЫХ	207
Силич В. А., Силич М. П., Аксенов С. В. КОСВЕННАЯ ОЦЕНКА ФАКТОРОВ ЭНЕРГЕТИЧЕСКОЙ ЭФФЕКТИВНОСТИ ТЕРРИТОРИАЛЬНЫХ ОБРАЗОВАНИЙ	210
Сасева А. В. ИМИТАЦИОННОЕ РЕШЕНИЕ ЗАДАЧИ ИЗГОТОВЛЕНИЯ РЕКЛАМНЫХ СТЕНДОВ.....	213
Петрунина Е. В. ГЕТЕРОГЕННАЯ ИНФОРМАЦИОННО-МОДЕЛИРУЮЩАЯ СИСТЕМА СЛОЖНЫХ ЭРГАТИЧЕСКИХ КОМПЛЕКСОВ.....	215
Зыков М. С., Огнев И. В., Горьков А. Г. СПОСОБ ФОРМИРОВАНИЯ БАЗЫ ДАННЫХ ДЛЯ ФРАГМЕНТАРНОГО МЕТОДА СЖАТИЯ ИНФОРМАЦИИ.....	219
Дамбаева С. В., Мердыгеев Б. Д. СТРУКТУРЫ ОНТОЛОГИИ.....	222
Цукарев Э. В. МОДЕРНИЗАЦИЯ ИНИЦИАТИВНОГО ОПРОСА НА СЕРВЕРЕ КОНСОЛИДАЦИИ ТЕХНОЛОГИЧЕСКИХ ДАННЫХ	227
Чечель П. К., Краснов А. А., Дубравин А. В. РАЗРАБОТКА УНИВЕРСАЛЬНОГО КАТАЛОГА ЦИФРОВЫХ И АНАЛОГОВЫХ ДАТЧИКОВ, ПРОИЗВОДИМЫХ НА ТЕРРИТОРИИ РФ И СНГ: СТРУКТУРА И СРЕДСТВА РЕАЛИЗАЦИИ.....	230

Макарычев П. П.

ПРОЦЕДУРА ПОИСКА ОПТИМАЛЬНОГО РЕШЕНИЯ
МЕТОДОМ ПОСЛЕДОВАТЕЛЬНЫХ УСТУПОК 233

Демин Е. С., Вяткин В. В., Дубинин В. Н.

ИСПОЛЬЗОВАНИЕ СЕРВИСНО-ОРИЕНТИРОВАННЫХ
АРХИТЕКТУР В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ
НА ОСНОВЕ СТАНДАРТА IEC 61499 237

Адамова А. Д.

РАЗРАБОТКА КОМПЛЕКСНОГО АЛГОРИТМА ПРИЕМА
СИГНАЛОВ В ВСТРОЕННЫХ
ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМАХ 242

Belinskaya N. S.

DETERMINATION OF THE REFLUX AMOUNT
IN THE STABILIZATION COLUMN OF HYDRODEWAXING
UNIT USING HYSYS MODELLING ENVIRONMENT 246

Семенов А. Д., Михайлов П. Г., Будаговский Д. А.

МОДЕЛЬНОЕ ПРОГНОЗИРУЮЩЕЕ УПРАВЛЕНИЕ 248

Гурьянов Л. В., Авдонин И. В., Башев Б. С.

ИСПОЛЬЗОВАНИЕ CAN ШИНЫ ДЛЯ УПРАВЛЕНИЯ
СОСТОЯНИЕМ АВТОМОБИЛЯ 252

Drozdov D., Dubinin V., Patil S., Vyatkin V.

REFACTORING SMV MODELS BY REPLACING ASYNCHRONOUS
PROCESSES WITH SYNCHRONOUS ONES AND ITS APPLICATION
TO PETRI NETS AND IEC 61499 FUNCTION BLOCKS DOMAINS 255

Брякин Л. А., Кучин А. В.

ДАТЧИК ХОЛЛА НА ПОЛЕВОМ ТРАНЗИСТОРЕ 261

Дудоров А. В., Дубинин В. Н., Дёмин Е. С.

РАЗРАБОТКА ОНТОЛОГИИ ТЕХНОЛОГИЧЕСКИХ СХЕМ
ДЛЯ СИСТЕМ АВТОМАТИЗАЦИИ 262

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Войнова Е. А.

ФОРМАЛИЗАЦИЯ ПРОЦЕССОВ УПРАВЛЕНИЯ
ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ
ЯЗЫКА СОБЫТИЙНЫХ НЕДЕТЕРМИНИРОВАННЫХ АВТОМАТОВ 267

Гавришев А. А., Бурмистров В. А., Анзин И. В.

К ВОПРОСУ ОБ ИСПОЛЬЗОВАНИИ ПСЕВДОСЛУЧАЙНЫХ
ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ ПРЕДОТВРАЩЕНИЯ
НЕСАНКЦИОНИРОВАННОГО ДОСТУПА К РАДИОКАНАЛУ
АВТОМОБИЛЬНОЙ СИГНАЛИЗАЦИИ 270

Гольшевский Е. О.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ПРОВЕДЕНИЯ СПЕЦИАЛЬНОЙ
ПРОВЕРКИ МЕТОДОМ АКТИВНОГО КОНТРОЛЯ ЦЕПЕЙ ПИТАНИЯ
С ИСПОЛЬЗОВАНИЕМ АППАРАТУРЫ ТЕСТОВОГО ПРИЕМНИКА
«RONDE&SCHWARZ» 272

Ефимов Б. И. АНАЛИЗ ЭФФЕКТИВНОСТИ УГРОЗ БЛОКИРОВАНИЯ И ПОДМЕНЫ ОТВЕТОВ ЭКСПЕРТОВ В СИСТЕМАХ ПРИНЯТИЯ РЕШЕНИЙ.....	275
Зефирова С. Л., Щербакова А. Ю. АНАЛИЗ СОСТОЯНИЙ ИНЦИДЕНТОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРИ ИХ ОБРАБОТКЕ	278
Легошин В. В., Жукова Н. Б. СРАВНЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ АВТОРЕГРЕССИОННОГО СПЕКТРАЛЬНОГО ОЦЕНИВАНИЯ ПРИМЕНИТЕЛЬНО К ЗАДАЧЕ СИНХРОНИЗАЦИИ В СИСТЕМАХ СКРЫТОЙ ПЕРЕДАЧИ ИНФОРМАЦИИ.....	282
Вовк С. В. МОДЕЛИРОВАНИЕ И ФИЛЬТРАЦИЯ СИГНАЛОВ В СРЕДЕ SCILAB	285
Яговитов Д. С., Липилин О. В. ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ТЕХНОЛОГИИ IP-ТЕЛЕФОНИИ ПРИ ИСПОЛЬЗОВАНИИ IP-АТС «ASTERISK»	288
Арефьев Д. В. СРЕДСТВА ЗАЩИТЫ АППАРАТУРЫ ПЕРЕДАЧИ ДАННЫХ ОТ УГРОЗЫ ПРОНИКНОВЕНИЯ НЕЗАШИФРОВАННОЙ ИНФОРМАЦИИ В КАНАЛ СВЯЗИ.....	291
Карпушин Д. А. ПОВЫШЕНИЕ ДРУЖЕСТВЕННОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ, ИСПОЛЬЗУЮЩИХ СТРОГУЮ ИДЕНТИФИКАЦИЮ И АУТЕНТИФИКАЦИЮ ПОЛЬЗОВАТЕЛЯ.....	294

СЕТИ ЭВМ И ТЕЛЕКОММУНИКАЦИИ

Перепёлкин Д. А., Иванчикова М. А. ЗАДАЧА ФОРМИРОВАНИЯ ОПТИМАЛЬНЫХ МАРШРУТОВ ПЕРЕДАЧИ ДАННЫХ КОРПОРАТИВНЫХ СЕТЕЙ В УСЛОВИЯХ ДЕЙСТВИЯ НА ТЕРРИТОРИИ РЕГИОНА НЕСКОЛЬКИХ ПРОВАЙДЕРОВ СВЯЗИ С РАЗЛИЧНЫМИ ЗОНАМИ ПОКРЫТИЯ АБОНЕНТОВ	297
Перепёлкин Д. А. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ МНОГОПУТЕВОЙ АДАПТИВНОЙ МАРШРУТИЗАЦИИ В КОРПОРАТИВНЫХ СЕТЯХ С БАЛАНСИРОВКОЙ НАГРУЗКИ НА ОСНОВЕ ДАННЫХ О ПАРНЫХ ПЕРЕСТАНОВКАХ МАРШРУТОВ.....	301
Семёнов А. О., Конов Н. Н. ФОРМАЛЬНОЕ ПРЕДСТАВЛЕНИЕ СТОХАСТИЧЕСКОГО АЛГОРИТМА ДИСПЕТЧЕРИЗАЦИИ ОЧЕРЕДЕЙ В ТЕЛЕКОММУНИКАЦИОННОМ ОБОРУДОВАНИИ	306
Семёнов А. О. КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ СТОХАСТИЧЕСКОГО АЛГОРИТМА ДИСПЕТЧЕРИЗАЦИИ ОЧЕРЕДЕЙ В ТЕЛЕКОММУНИКАЦИОННОМ ОБОРУДОВАНИИ.....	310

Сакерин А. В., Родионов А. С., Мигов Д. А.	
ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ В РАСЧЕТЕ ПОКАЗАТЕЛЕЙ НАДЕЖНОСТИ СЕТЕЙ.....	313
Жукова Н. Б.	
МЕТОД СОКРАЩЕНИЯ ДЛИТЕЛЬНОСТИ ПРОЦЕДУРЫ ПЕРВОНАЧАЛЬНОЙ СИНХРОНИЗАЦИИ В СИСТЕМАХ С ПРОГРАММНОЙ ПЕРЕСТРОЙКОЙ РАБОЧЕЙ ЧАСТОТЫ.....	316
Мельников Н. Э.	
СИСТЕМА ПЕРЕДАЧИ ДАННЫХ НА ПРИМЕРЕ ВЫСОКОЧАСТОТНЫХ РАДИОМОДЕМОВ.....	319
Ерёмкина Р. А.	
ОРГАНИЗАЦИЯ КОММУТАТОРА FAST ETHERNET С ПОДДЕРЖКОЙ QOS	322
Кизилов Е. А., Коннов Н. Н., Механов В. Б.	
МОДЕЛИРОВАНИЕ КОММУТАТОРОВ ETHERNET В ПРОМЫШЛЕННЫХ СЕТЯХ.....	326
Бессонов С. Н.	
НАЗНАЧЕНИЕ БЕСПРОВОДНЫХ МНОГОРАНГОВЫХ САМООРГАНИЗУЮЩИХСЯ СЕТЕЙ, ТЕХНОЛОГИИ И ПРИНЦИПЫ ПОСТРОЕНИЯ	329
Mustafa Sadeq Jaafar	
ON MODIFICATION ACK-BASED SCHEME USING FSA FOR AD-HOC NETWORKS	334
Ильина Ю. Д., Иванов А. П.	
ОБЗОР МЕТОДОВ АНАЛИЗА СИГНАЛОВ ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ	340
Кашиев Е. Д., Иванов А. П., Тикин М. С.	
ВЛИЯНИЕ ШУМА В РАДИОКАНАЛЕ НА ВЫПОЛНЕНИЕ АЛГОРИТМА ОБЕСПЕЧЕНИЯ ЦЕЛОСТНОСТИ СИГНАЛОВ УСТРОЙСТВА ПРЕОБРАЗОВАНИЯ СИГНАЛОВ	343
Крупский А. С.	
МЕТОД РАСПЕРЕДЕЛЕНИЯ НАГРУЗКИ ПО ВРЕМЕНИ ПРИ РЕШЕНИИ ЗАДАЧ В ОДНОРОДНОЙ ВОСЬМИСВЯЗНОЙ СЕТИ	345

МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Волчихин В. И., Дубинин В. Н.	
МЕТОДИКА МОДЕЛИРОВАНИЯ НЕДЕТЕРМИНИРОВАННЫХ АВТОМАТОВ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА LD	348
Трофимова О. В.	
ПОЛУМАРКОВСКАЯ МОДЕЛЬ ПЕРЕХОДОВ СОСТОЯНИЙ ПАЦИЕНТА (БОЛЬНОГО) В ПРОЦЕССЕ ЛЕЧЕНИЯ ЗАБОЛЕВАНИЯ	354
Тимченко В. С.	
ПРИМЕНЕНИЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДЛЯ ОЦЕНКИ ПЕРЕРАБАТЫВАЮЩЕЙ СПОСОБНОСТИ ГРУЗОВОГО ФРОНТА	357

Такташкин Д. В.	
КОМПЛЕКСНАЯ МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ПРОЦЕССА ОХЛАЖДЕНИЯ РЕЖУЩЕГО ИНСТРУМЕНТА.....	360
Васильев И. С., Попов С. А.	
ОЦЕНИВАНИЕ SPICE-ПАРАМЕТРОВ ТРАНЗИСТОРОВ ПРИ ЗАДАННОЙ ТОЧНОСТИ РАСЧЕТА ЭЛЕКТРИЧЕСКИХ РЕЖИМОВ ЭЛЕКТРОННЫХ УСТРОЙСТВ	363
Болодурина И. П., Парфёнов Д. И.	
МОДЕЛИРОВАНИЕ ВИРТУАЛЬНЫХ ПОТОКОВ ОБЛАЧНОЙ СИСТЕМЫ И ОПТИМИЗАЦИЯ ДОСТУПА К ВЫЧИСЛИТЕЛЬНЫМ РЕСУРСАМ.....	367
Зинкин С. А., Шеланков О. Е.	
ОБОСНОВАНИЕ СОЗДАНИЯ И ПРИМЕНЕНИЯ АЛГЕБРЫ СОБЫТИЙНО-ИЕРАРХИЧЕСКИХ ФРЕЙМОВ	370
Михайлова С. Е., Пучков А. Ю.	
СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ	374
Медовщикова С. А., Паршин А. М., Пащенко Д. В.	
МОДЕЛЬ РАСЧЕТА ПРОИЗВОДСТВЕННЫХ ВОЗМОЖНОСТЕЙ РЕМОНТНО-ВОССТАНОВИТЕЛЬНЫХ ПОДРАЗДЕЛЕНИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ	375
Мартышкин А. И.	
ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СМО С ПОМОЩЬЮ MATLAB+SIMULINK	378
Белинская Н. С.	
ПРИМЕНЕНИЕ КОМПЬЮТЕРНОЙ МОДЕЛИРУЮЩЕЙ СИСТЕМЫ ДЛЯ ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОГО СООТНОШЕНИЯ ВОДОРОД/СЫРЬЕ В ПРОЦЕССЕ ГИДРОДЕПАРАФИНИЗАЦИИ ДИЗЕЛЬНЫХ ТОПЛИВ.....	382
Балашова И. Ю.	
КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ДЛЯ ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ ЭЛЕКТРОННОГО ОБУЧЕНИЯ.....	384
Волынская К. И.	
МОДЕЛИ И ПРЕДСТАВЛЕНИЯ ОНТОЛОГИИ ПРОЕКТИРОВАНИЯ И РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	388
Убиенных Г. Ф., Убиенных А. Г.	
ЧИСЛЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНЫХ КРАЕВЫХ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ НА ОСНОВЕ РБФ-АППРОКСИМАЦИИ	391
Убиенных Г. Ф., Убиенных А. Г.	
МОДЕЛИРОВАНИЕ ЦЕНООБРАЗОВАНИЯ ОПЦИОНОВ НА ОСНОВЕ АППРОКСИМАЦИИ РЕШЕНИЯ УРАВНЕНИЯ БЛЭКА–ШОУЛЗА ФУНКЦИЯМИ РАДИАЛЬНОГО БАЗИСА.....	394
Малинова Е. А., Пучков А. Ю.	
ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РАБОТЫ ЦЕХОВ ПО СБОРКЕ ОКОН НА ПРЕДПРИЯТИИ «МОСКОВСКИЕ ОКНА».....	398

Захаров В. Г. МОДЕЛИРОВАНИЕ НА ЭВМ БЕЛОГО ГАУССОВСКОГО ШУМА С ЗАДАНЫМИ МАТЕМАТИЧЕСКИМ ОЖИДАНИЕМ И ДИСПЕРСИЕЙ	400

Кононова М. В. ВЕРОЯТНОСТНАЯ АВТОМАТНАЯ МОДЕЛЬ ПЕРЕХОДОВ СОСТОЯНИЙ МЕДИЦИНСКОГО ОБОРУДОВАНИЯ В ПРОЦЕССЕ РЕМОНТА.....	402

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Вашкевич Н. П., Волчихин В. И., Бикташев Р. А. ВЕРИФИКАЦИЯ СИСТЕМЫ КАНОНИЧЕСКИХ УРАВНЕНИЙ ДЛЯ Q-СОБЫТИЙ, ОПРЕДЕЛЯЮЩИХ СОСТОЯНИЯ ДЕТЕРМИНИРОВАННОГО АВТОМАТА, ИСПОЛЬЗУЕМОГО ДЛЯ СТРУКТУРНОЙ РЕАЛИЗАЦИИ АЛГОРИТМА УПРАВЛЕНИЯ ПРОЦЕССАМИ И РЕСУРСАМИ.....	406

Бикташев Р. А., Вашкевич Н. П., Левин Д. А. РАЗРАБОТКА МЕНЕДЖЕРА МЬЮТЕКСОВ С АППАРАТНОЙ ПОДДЕРЖКОЙ ДЛЯ ОПЕРАЦИОННЫХ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ	410

ЭЛЕКТРОННЫЙ ДОКУМЕНТООБОРОТ И ЭЛЕКТРОННОЕ ПРАВИТЕЛЬСТВО

Кашанов Н. Р., Файрузов Р. А. АНАЛИЗ РЫНКА СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА	416

Кревский И. Г. ОСНОВНЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ ЭЛЕКТРОННОГО ОБУЧЕНИЯ	418

Кручинин А. А. ПОДХОД К ФОРМАЛИЗАЦИИ ПРОЦЕССА ЭКСПЕРТНОЙ ОЦЕНКИ НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ И ОПЫТНО-КОНСТРУКТОРСКИХ РАБОТ	421

Ушаков В. А. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ДЕКАНАТА В ХОДЕ ПОДГОТОВКИ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ БАКАЛАВРА	423

Кольчугина Е. А. ПАРАДИГМА НЕРАВНОВЕСНОГО ПРОГРАММИРОВАНИЯ И ДРУГИЕ СОВРЕМЕННЫЕ ПАРАДИГМЫ	426

Кольчугина Е. А. ПАРАДИГМА НЕРАВНОВЕСНОГО ПРОГРАММИРОВАНИЯ И ИНДУСТРИЯ РАЗРАБОТКИ ПРОГРАММ.....	429

Кольчугина Е. А. ИДЕНТИФИКАЦИЯ ЦИФРОВЫХ ОРГАНИЗМОВ	431

Научное издание

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ

СБОРНИК НАУЧНЫХ СТАТЕЙ
XI Международной научно-технической
конференции

г. Пенза, 25–27 ноября 2014 г.

Компьютерная верстка *М. Б. Жучковой*

Подписано в печать 21.11.2014.
Формат 60×84¹/₈. Усл. печ. л. 52,62.
Тираж 50. Заказ № 1042.

Издательство ПГУ.
440026, Пенза, Красная, 40.
Тел./факс: (8412) 56-47-33; e-mail: iic@pnzgu.ru