

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**ВЕСТНИК II МЕЖВУЗОВСКОЙ
КОНФЕРЕНЦИИ МОЛОДЫХ
УЧЕНЫХ**

Сборник научных трудов

Том 1



**САНКТ-ПЕТЕРБУРГ
2005**

Выпуск содержит материалы II межвузовской конференции молодых учёных, посвященной 100-летию первого выпуска механико-оптического и часового отделения Ремесленного училища цесаревича Николая – предшественника Санкт-Петербургского государственного университета информационных технологий, механики и оптики.

Конференция была проведена 28–31 марта 2005 г. Санкт-Петербургским государственным университетом информационных технологий, механики и оптики.

ПРОГРАММНЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

Председатель

Ректор СПбГУ ИТМО, д.т.н., профессор **В.Н. Васильев**

Сопредседатели

Проректор по развитию, д.т.н., проф. **В.О. Никифоров**
Проректор по УО и АР, д.ф.-м.н., проф. **Ю.Л. Колесников**
Декан факультета ППО, д.т.н., проф. **В.Л. Ткалич**

Члены программного комитета

Д.ф.-м.н., проф. **Л.Н. Андреев**, д.т.н., проф. **В.А. Зверев**, д.т.н., проф. **В.А. Иванов**, д.т.н., проф. **В.К. Кирилловский**, д.т.н., проф. **А.Г. Коробейников**, д.т.н., проф. **Д.Д. Куликов**, д.т.н., проф. **С.М. Латыев**, д.т.н., проф. **В.М. Мусалимов**, д.ф.-м.н., проф. **Н.В. Никоноров**, д.т.н., проф. **Э.Д. Панков**, д.т.н., проф. **Э.С. Путилин**, д.ф.-м.н., проф. **В.С. Сизиков**, д.т.н., проф. **А.М. Скворцов**, д.т.н., проф. **С.Б. Смирнов**, д.т.н., проф. **С.К. Стафеев**, д.т.н., проф. **В.А. Тарлыков**, д.т.н., проф. **А.А. Шалыто**, д.т.н., проф. **А.В. Шарков**, д.т.н., проф. **Е.Б. Яковлев**, к.т.н., доц. **В.М. Домненко**, к.т.н., доц. **М.Я. Марусина**, к.т.н., проф. **Б.С. Падун**, к.т.н., проф. **М.И. Потеев**, к.ф.н., доц. **В.Н. Садовников**, к.т.н., доц. **И.Н. Тимошук**, к.т.н., доц. **Б.Д. Тимченко**, к.т.н. **Т.В. Точилина**, к.т.н., доц. **Е.Ф. Чуфаров**, к.т.н., доц. **Е.В. Шалобаев**

ОРГАНИЗАЦИОННЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

Председатель

Зам. проректора по НР **Л.М. Студеникин**

Зам. председателя

К.т.н. **Т.В. Точилина**

Члены организационного комитета

К.В. Богданов, **П.А. Борисов**, **Н.Н. Валентик**, **Н.Ф. Гусарова**, **И.Н. Жданов**, **С.Ю. Кerpелева**, **Н.В. Когай**, **Д.В. Лукичев**, **А.Г. Метляков**, **Н.Б. Нечаева**, **М.В. Никитина**, **А.В. Перепелкин**, **Т.А. Прудентова**

ISBN 5-7577-0259-1

© Санкт-Петербургский государственный университет
информационных технологий, механики и оптики,
2005

Уважаемый читатель!

Вы держите в руках сборник научных трудов II Межвузовской конференции молодых учёных, прошедшей в марте 2005 г. в Санкт-Петербургском государственном университете информационных технологий, механики и оптики. Участие в конференции приняли студенты и аспиранты Санкт-Петербургского государственного университета, Санкт-Петербургского государственного морского технического университета, Санкт-Петербургского государственного университета кино и телевидения, Пермского государственного университета, Ставропольского государственного университета.

Материалы, вошедшие в сборник, отражают содержание докладов, прозвучавших на 11 секциях конференции. Они могут представлять интерес для молодых специалистов, чьи научные интересы лежат в сфере оплотехники, оптического приборостроения и материаловедения, лазерной физики, физики кристаллов и полупроводников, механики и мехатроники, измерительной техники, технологии приборостроения, информационных технологий, САПР, системного анализа и теории управления, защиты информации, медицинской техники, томографии, а также целого ряда гуманитарных областей: технологий и методик образования, экономики и менеджмента, философии, социологии, политологии.

Работы, вошедшие в сборник, отражают широкий спектр научно-технических исследований, которые ведутся как в вузах, участвующих в конференции, так и в целом ряде сотрудничающих с ним научных и производственных организаций: ВНЦ ГОИ им С.И. Вавилова, Физико-техническом институте им. А.Ф. Иоффе, Институте аналитического

приборостроения РАН, Институте мозга человека РАН, Высокогорном геофизическом институте Роскомгидромета (г. Нальчик), ОАО «ЛОМО» и других.

*Доктор технических наук,
профессор*

В.Л. Ткалич

АЛГОРИТМЫ УКЛАДКИ ДИАГРАММ СОСТОЯНИЙ**М.А. Коротков****Научный руководитель – к.ф.-м.н., доцент Ф.А. Новиков**

В статье приведен обзор методов плоской укладки графов, поставлена задача плоской укладки диаграмм состояний *UML*. Описан алгоритм с физической моделью (метод отжига) и один из аналитических алгоритмов.

Введение

Программный пакет с открытым исходным кодом *UniMod* [1] обеспечивает разработку и выполнение автоматически-ориентированных программ. Он позволяет создавать и редактировать диаграммы классов и состояний *UML* [2], которые соответствуют графу переходов и схеме связей конечного автомата [3]. После создания диаграмм существует возможность выполнить их, при этом содержимое диаграмм преобразуется в *XML*-описание, которое передается интерпретатору, также входящему в пакет *UniMod*. Такой подход является реализацией парадигмы автоматного программирования [4].

Во многих современных редакторах для текстовых языков программирования существует возможность автоматического форматирования программного кода. Если программным кодом является не текст, а набор диаграмм, задача форматирования текста преобразуется в задачу укладки диаграмм. Более подробно задача будет описана ниже. В рамках проекта *UniMod* основной интерес представляет укладка диаграмм состояний языка *UML*.

При укладке диаграммы необходимо учитывать ряд критериев, которые могут противоречить друг другу, например, критерии минимизации площади, занимаемой диаграммой, и наличия достаточного количества свободного места («воздуха»). Каждый критерий оценивает «качество» диаграммы для того или иного применения (отображения на мониторе, печати т.д.). Такие критерии называются **эстетиками**. Задавшись некоторым набором эстетик, можно построить штрафную функцию (которая тем больше, чем больше расхождения между изображением диаграммы и критериями) и пытаться минимизировать её, перемещая элементы, или разработать алгоритм, последовательно модифицирующий исходную диаграмму так, чтобы результат соответствовал выбранным критериям.

1. Постановка задачи

Рассмотрим диаграмму состояний конечного автомата. Она включает в себя состояния и переходы (и у тех, и у других есть дополнительные, связанные с ними, сущности – метки, но мы сейчас не будем останавливаться на этом вопросе). Диаграмме состояний (без вложенных состояний) можно сопоставить граф [5] (при укладке диаграммы состояний нам не важна ориентация дуг). Каждому состоянию соответствует вершина, а переходу – ребро. Нам необходимо несколько сузить поле деятельности, зафиксировав представление элементов и тип носителя, на котором мы планируем изображать диаграмму.

1.1. Выбор типа носителя

Диаграмма может быть изображена на плоскости, или, например, может быть построена объемная модель [6]. Для представления на мониторе современного персонального компьютера наиболее естественным представляется выбор плоского носителя.

1.2. Выбор представления элементов

Необходимо выбрать вид элементов графа. Будем изображать вершину как прямоугольник, а ребро – как ломаную линию с конечным числом изломов.

Выбор именно такого вида элементов объясняется исключительно субъективными эстетическими соображениями, а также сложившимися в области построения диаграмм традициями. Такой выбор диктует использование декартовой системы координат. Вершину принято изображать как прямоугольник со сторонами, ориентированными параллельно координатным осям. Для изображения вершины достаточно знать координаты левого верхнего угла, ширину и высоту, а для изображения ребра – координаты его начала, конца и всех точек излома.

1.3. Понятие укладки

Укладкой графа $G = (V; E)$ в декартовой системе координат $(X; Y)$ назовем множество $L = (G; F_V; F_E)$, где F_V – функция из множества вершин в множество параметров, необходимых для представления вершины в выбранной системе координат (в нашем случае $F_V: V \rightarrow X \times Y \times R \times R$, где последняя пара параметров – ширина и высота прямоугольника). F_E – функция из множества ребер в множество параметров, необходимых для представления ребра в выбранной системе координат (в нашем случае $F_E: V \rightarrow (X \times Y)^n, n \in N$, где параметр n – количество изломов – вообще говоря, меняется от ребра к ребру). Для простоты будем говорить, что в уложенном графе заданы геометрические параметры каждого элемента.

На задаче построения визуального изображения диаграммы по известным параметрам ее элементов в заданной системе координат (по заданной укладке) останавливаться здесь не будем.

Две укладки L, L' назовем **изоморфными**, если одна получается из другой путем параллельного переноса всех элементов на определенный вектор. Укладки на рис.1, например, являются различными. Далее мы, для простоты, будем говорить об изоморфных укладках как об одинаковых.

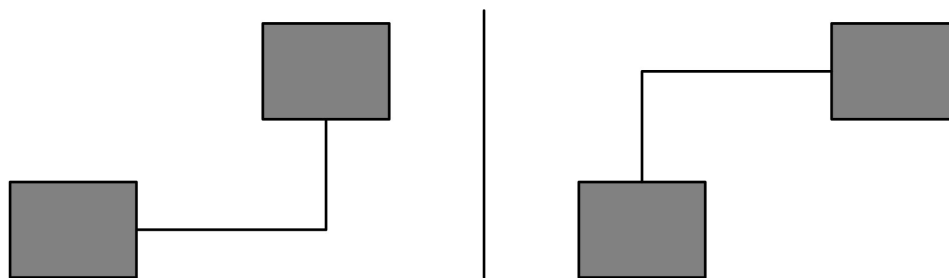


Рис. 1. Различные укладки

Таким образом, алгоритм укладки диаграммы должен строить пару функций укладки $F_V; F_E$, точнее, соответствующие наборы геометрических параметров для каждой вершины.

Укладку на плоскости мы будем называть **плоской**. Данный термин не является общепринятым, в литературе (например, [7]) чаще говорят о **плоском представлении** графа. Укладку, в которой ребра представляют собой ломаные, состоящие только из горизонтальных и вертикальных отрезков, будем называть **плоской ортогональной** (или просто **ортогональной**).

1.4. Оценка качества укладки

Теперь необходимо оценить качество укладки (точнее, качество изображения, полученного по некоторой укладке). Наиболее точный ответ способен дать человек. По-

пробуем выделить набор эстетик, по которым мы будем оценивать качество укладки. Основная задача – обеспечить «читаемость» графа (однозначность представления информации):

- количества пересечений ребер, прохождений ребер по вершинам и наложения вершин должны быть минимальны;
- площадь, занимаемая выпуклой оболочкой уложенного графа (или площадь минимального прямоугольника, включающего в себя уложенный граф), должна быть минимальной;
- доля свободного места на диаграмме не должны быть меньше некоторого предела;
- количество изломов и суммарная длина ребер должны быть минимальны.

К примеру, укладка слева на рис. 2 значительно лучше читается, чем укладка справа.

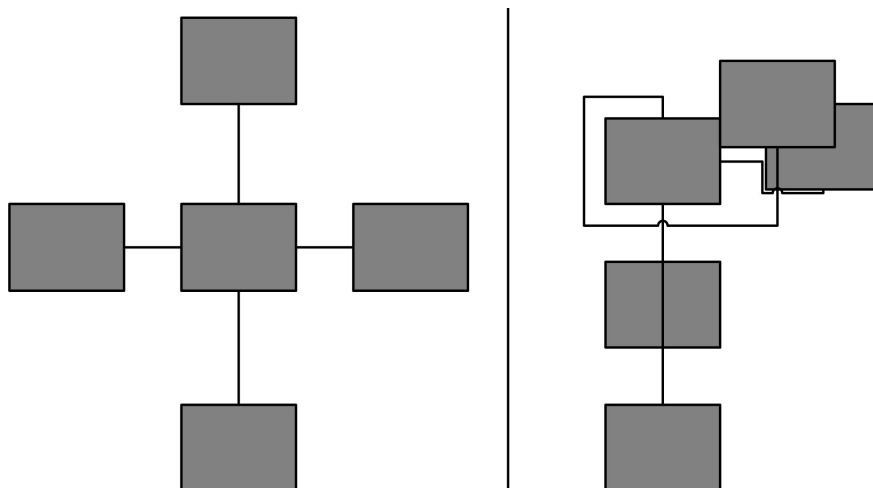


Рис. 2. Укладки графа

Граф, соответствующий диаграмме состояний, отличается от обычного графа возможностью вложения состояний друг в друга, наличием псевдосостояний, а также выделенных начальных и конечных состояний. Для оценки качества его укладки необходимо учесть дополнительные критерии:

- размер всех простых состояний (состояний, не имеющих вложенных состояний) необходимо максимально приблизить к заданному оптимальному размеру;
- расстояние между начальным и конечным состоянием должно быть достаточно велико.

Сравним две укладки на рис. 3. Укладка справа является предпочтительной, поскольку на ней автомат «визуально» выполняется слева направо (от начального состояния к конечному). Заметим, что приведенные эстетик не являются строгими и лишь дают понять, какими соображениями мы будем руководствоваться, оценивая результаты работы программ.

Обратим внимание на то, что в принятые в диаграмме состояний *UML* начальное и конечное состояние не имеют прямоугольной формы. Для простоты заменим их объемлющими прямоугольниками.

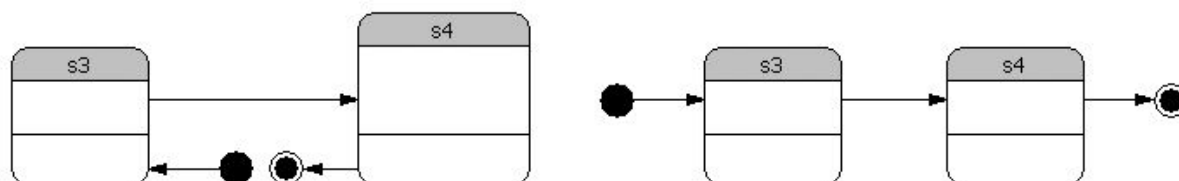


Рис. 3. Укладки диаграммы состояний

2. Типы алгоритмов укладки графов

Задача укладки графа не имеет и не может иметь универсального решения, в связи с тем, что набор критериев, применяемых для оценки качества укладки, зависит от типа диаграммы. В [8] приведена классификация алгоритмов, применяемых для решения данной задачи. Рассмотрим две большие группы, принципиально отличающиеся подходом к решению:

- алгоритмы с физическим аналогом [9];
- аналитические алгоритмы [7].

2.1. Алгоритмы с физическим аналогом

Эта группа алгоритмов ставит в соответствие графу некоторую физическую модель, например, систему пружин, которые стремятся сжаться до некоторой заданной длины (такие алгоритмы называют «пружинным методом») или систему стержней и шарниров, с вершинами – одноименными зарядами.

Для описания физической модели вводится понятие штрафной функции, задающей потенциальную энергию системы (такие алгоритмы еще называют методами минимизации потенциала). Задача укладки преобразуется, таким образом, в задачу нахождения минимума этой функции, которая решается с помощью сдвига на случайный вектор каждой вершины графа и проверки изменения значения штрафной функции. Сдвиг вершин выполняется в цикле, условием выхода из которого является либо достижение локального минимума, либо достижение максимума числа допустимых итераций.

Наиболее популярная подгруппа группы алгоритмов с физическим аналогом – **методы отжига**. Они выделяются тем, что «колебания» системы затухают с каждой итерацией.

Для минимизации штрафной функции также можно использовать генетические алгоритмы. В [10] приводится обоснование такого подхода. Генетические алгоритмы строятся для минимизации тех же штрафных функций и отличаются, в основном, более высоким быстродействием.

Нахождение минимума потенциальной энергии также можно выполнить, решив систему дифференциальных уравнений, задающих ее. Для этого необходимо построить гладкую штрафную функцию, что является нетривиальной задачей. В [9] приведена ситуация, в которой малое изменение положения вершин должно приводить к резкому скачку штрафной функции.

На рис. 4 приведены две различные укладки одного и того же графа. Критерий «читаемости» графа, заключающийся в том, что пользователю должно быть понятно, какую пару вершин связывает каждое ребро, является для нас одним из важнейших. Следовательно, штрафная функция должна резко возрастать при нарушении условия читаемости. Теперь обратим внимание на то, что укладка слева на легко «читается», тогда как укладка справа «нечитаема» из-за наложения ребер. С другой стороны, координаты элементов обеих упадок отличаются незначительно. Это означает, что небольшой сдвиг вершин графа должен приводить к резкому скачку штрафной функции.

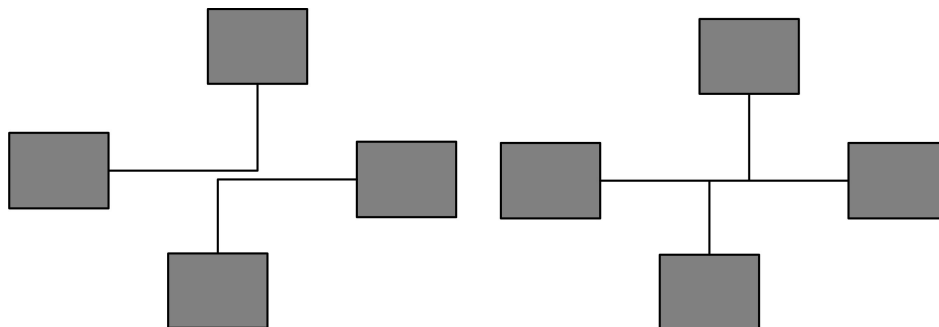


Рис. 4. Скачок штрафной функции

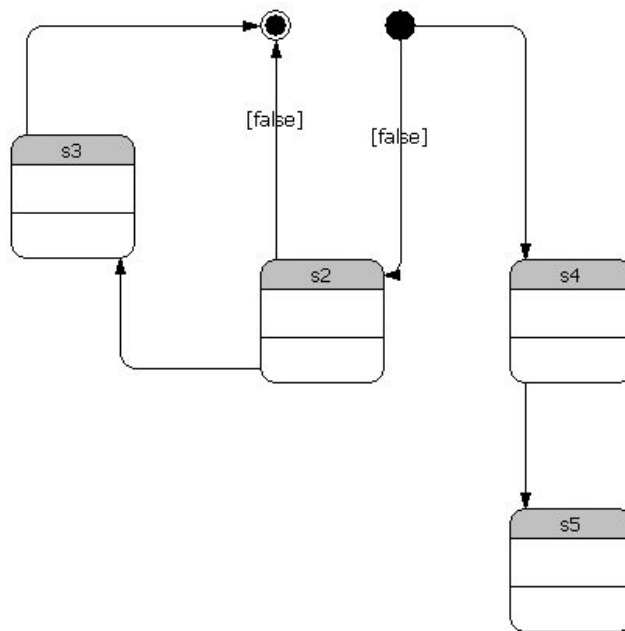


Рис. 5. Уложенная диаграмма

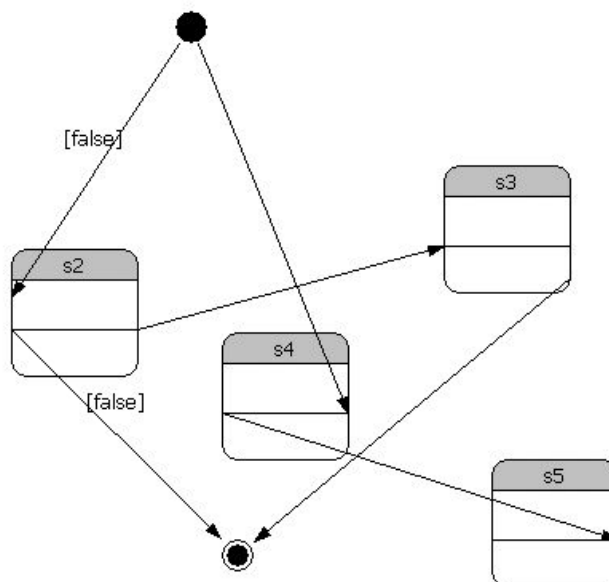


Рис. 6. Исходная диаграмма

Простота реализации метода отжига стала причиной реализации этого алгоритма для укладки диаграммы состояний в рамках проекта *UniMod*. Был выбран следующий подход: построим первоначальную укладку с помощью алгоритма отжига, а затем ортогонализуем ее (вообще говоря, с помощью аналитического алгоритма). Получившийся алгоритм можно назвать смешанным. При этом в процессе ортогонализации могут появиться дополнительные пересечения ребер, не учтенные алгоритмом отжига.

На рис. 5 приведена диаграмма состояний, полученная из диаграммы, приведенной на рис. 6, с помощью модифицированного алгоритма отжига.

2.2. Аналитические алгоритмы

Аналитические алгоритмы, в отличие от алгоритмов с физическим аналогом, представляют собой последовательность различных преобразований графа, приводящую к построению укладки. Это позволяет получать с их помощью гарантированный результат, удовлетворяющий выбранным критериям, так как критерии используются не для построения штрафной функции, а для построения самого алгоритма укладки.

К минусам аналитических алгоритмов можно отнести сложность их построения. Кроме того, они плохо поддаются модификации, и для постановки экспериментов приходится вносить серьезные изменения в сам алгоритм, а не в набор параметров, как это можно делать в случае использования, например, алгоритмов, базирующихся на штрафных функциях.

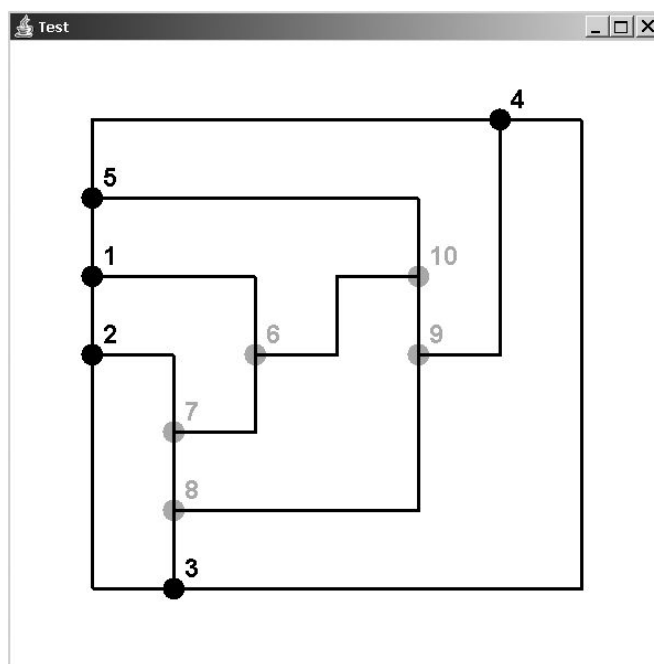


Рис. 7. Укладка графа

На практике (рассмотрено три реализации: [11–13]) аналитические алгоритмы позволяют относительно быстро получать наиболее качественные и красивые графы, в связи с этим для укладки диаграммы состояний в проекте *UniMod* было принято решение использовать эту группу алгоритмов в качестве основной.

Наиболее эффективным [14] в группе аналитических алгоритмов является алгоритм GIOTTO ([7, 15]). В [16] приводится обоснование применения модификации алгоритма GIOTTO для решения задачи укладки диаграммы состояний. Основная часть работы [16] посвящена укладке реберных меток, однако данная проблема этой в статье не затрагивается. В [7] изложены основы алгоритма GIOTTO. Было решено реализовать его и адаптировать к задаче укладки диаграммы состояний. Вторая задача на данный момент не решена окончательно. На рис. 7 приведен пример графа, уложенного с помощью модифицированного алгоритма GIOTTO.

Заключение

В настоящей работе исследована проблема плоской укладки диаграмм состояний *UML*. Поставлена задача укладки, дан обзор наиболее популярных методов.

На базе аналитического алгоритма плоской укладки графа GIOTTO, являющегося наиболее эффективным [14], разработан алгоритм, позволяющий уложить произвольную диаграмму состояний языка *UML*. Ценность алгоритма заключается в том, что он разработан специально для укладки *UML*-диаграммы состояний. Это позволило учесть специфические особенности диаграмм данного типа при их плоской укладке.

Кроме того, в работе описан смешанный алгоритм, сочетающий в себе простоту метода отжига и оригинальный аналитический алгоритм построения ортогональной укладки.

Литература

1. Гуров В.С., Мазин М.А. Веб-сайт проекта UniMod. <http://unimod.sourceforge.net/>
2. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. М.: ДМК. 2000.
3. Шальто А.А. Switch-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука. 1998.
4. Шальто А.А., Туккель Н.И. Танки и автоматы // ВУТЕ/Россия. 2003. № 2, с. 69–73. [http://is.ifmo.ru/\(раздел «Статьи»\)](http://is.ifmo.ru/(раздел%20«Статьи»)).
5. Новиков Ф.А. Дискретная математика для программистов. СПб.: Питер. 2001.
6. Tamassia R. Advances in the Theory and Practice of Graph Drawing. <http://www.cs.brown.edu/people/rt/papers/ordal96/ordal96.html>
7. Battista G., Eades P., Tamassia R., Tollis I. Graph Drawing. Algorithms for the Visualization of Graphs. New Jersey: Prentice Hall. 1999.
8. Sugiyama K. Graph Drawing and Applications for Software and Knowledge Engineers. Singapore: Mainland Press. 2002.
9. Frutcherman T., Reingold E. Graph Drawing by Force-directed Placement, Software – Practice And Experience, 1991, Vol. 21, p1129–1164.
10. Makinen E., Seiranta M. Genetic algorithms for drawing bipartite graphs. International Journal of Computer Mathematics, 1994, Vol 53, No 3, p 157–166.
11. Веб-сайт проекта AGD (Algorithms for Graph Drawing). <http://www.ads.tuwien.ac.at/AGD/>
12. Веб-сайт проекта GDT (Graph Drawing Toolkit). <http://www.dia.uniroma3.it/~gdt>
13. Веб-сайт проекта Graph Layout Toolkit компании Tom Sawyer Software <http://www.tomsawyer.com/tsl/tsl.java.php>
14. Battista G., Garg A., Liotta G., Tamassia R., Tassinari E., Vargiu F., An experimental comparison of four graph drawing algorithms. Computational Geometry, 1997, 7(5-6), p303–325.
15. Tamassia R., Battista G., Batini C. Automatic graph drawing and readability of diagrams. IEEE Transactions on Systems Man Cybernetics, 1998, 18(1), p 61–79.
16. Klaw G., Mutzel P. Automatic layout and labeling of state diagrams. Materials of Graph Drawing conference, 1997.

ВЕРОЯТНОСТНАЯ МОДЕЛЬ СИСТЕМЫ ПОИСКА И НАВЕДЕНИЯ

А.В. Кучер

Научный руководитель – д.т.н., профессор А.В. Демин

В статье рассмотрен подход к построению математической модели сложной многорежимной системы с использованием счетных цепей Маркова.

Введение

Рассматривается алгоритм анализа состояний многорежимных систем применительно к выбору проектных решений. Особенность анализа состояний системы заключается в вероятностно-временной ограниченности всего цикла её работы, что может привести к неоднозначности оценки правильности выбранного проектного решения. В этой связи в статье дан подход к построению семейства вероятностно-временных характеристик и критерий выбора наиболее оптимальной. Выбранная вероятностно-временная характеристика и берётся за основу структуры системы, которая в дальнейшем и будет реализована. В качестве критерия предложен минимум «расстояния» между желаемой и реализуемой вероятностно-временной характеристики, т.е:

$$\min\{p_{\text{ж}} - p_{\text{ф}}\}, \min\{t_{\text{ж}} - t_{\text{ф}}\},$$

при условии $t_{\text{ж}} \geq t_{\text{ф}}$ и $p_{\text{ж}} \leq p_{\text{ф}}$, где $t_{\text{ж}}$ – желаемое время, необходимое для всего цикла работы системы, $t_{\text{ф}}$ – фактическое время работы системы, $p_{\text{ж}}$ – желаемая вероятность решения задачи (наведение управляемого объекта на цель), $p_{\text{ф}}$ – фактическая вероятность решения задачи системой при условии, что фактическая вероятность должна быть не меньше желаемой, а время поиска – не больше желаемой характеристики.

Предложенный подход был реализован при создании системы поиска и наведения.

При моделировании режимов работы системы поиска и наведения предполагались следующие условия:

- последовательность режимов – в каждый момент времени система находится только в одном состоянии;
- возможность возврата в один из режимов;
- ограничение во времени на работу всей системы;
- обеспечение необходимости задания вероятности перехода.

Математическая модель анализа режимов работы объекта технического проектирования

Система поиска и наведения (СПН) имеет пять режимов работы [2,3]:

- поиск объекта по вызову (имеется предварительное целеуказание);
- захват объекта (попадание его в поле зрения – поле захвата);
- совмещение линии визирования с направлением на цель с последующим сопровождением для выработки параметров её движения;
- повторный поиск цели при срыве одного из режимов;
- наведение управляемого объекта на цель (этот режим может быть конечным, т.е. цикл работы СПН считается завершенным, либо продолжение режима сопровождения (например, гидирование звезды)).

Функционирование СПН в каждом из этих основных режимов зависит от многих факторов и носит случайный характер. Для анализа работы СПН и выработке требований к отдельным ее режимам работы с точки зрения достижения конечного результата (наведение управляемого объекта на цель) рассмотрим построение математической модели переходов с режима на режим на основе марковских цепей [4,5]. При построении модели зафиксируем каждый из режимов работы СПН в качестве состояний марковской цепи, а именно [1, 6]:

- состояние 1 (V_1) – работа системы в режиме поиска;
- состояние 2 (V_2) – работа системы в режиме захвата;
- состояние 3 (V_3) – работа системы в режиме сопровождения;
- состояние 4 (V_4) – работа системы в режиме повторного поиска при срыве режимов сопровождения или захвата цели;
- состояние 5 (V_5) – работа системы в режиме наведения на цель.

Исходя из особенностей функционирования СПН, указанный выше, а также обеспечения режима реального масштаба времени, анализ необходимо проводить на модели дискретной марковской цепи. Для этого введем интервал дискретизации τ и номер интервала дискретности $m = 0, 1, 2, \dots$ – целое число. Тогда время исполняемого цикла определяется соотношением

$$t = \sum_1^5 w_i t_i = \tau \sum_1^5 w_i m_i, \quad (1)$$

где m_i – количество интервалов дискретности работы системы в i -м режиме, w_i – весовой коэффициент i -го режима работы. Весовые коэффициенты введены из расчета приведения в соответствие временных затрат на каждый режим работы СПН. Поскольку самым трудоемким по времени исполнения является режим поиска, то и значимость присваиваемого ему весового коэффициента – самая большая. Но при назначении коэффициентов необходимо исходить из того, что $\sum_{i=1}^5 w_i = 1$.

Определим переходы СПН из одного режима работы в другой, для чего построим граф переходов марковской цепи из состояния в состояние, соответствующий функциональным переходам (см. рис.1). В соответствии с графом переходов марковской модели составим разностное уравнение, описывающее изменение в дискретные моменты времени вероятности пребывания СПН в том или ином состоянии [6], для чего введем следующие обозначения:

- $P_1(m)$ – вероятность работы СПН в режиме поиска в пределах зоны (состояние V_1);
- $P_2(m)$ – вероятность работы СПН в режиме захвата (состояние V_2);
- $P_3(m)$ – вероятность работы СПН в режиме сопровождения (состояние V_3);
- $P_4(m)$ – вероятность работы СПН в режиме повторного поиска (состояние V_4);
- $P_5(m)$ – вероятность работы СПН в режиме наведения (состояние V_5);
- $P(m)$ – вектор распределения вероятностей по соответствующим состояниям:
 $P(m) = [P_1(m); P_2(m); P_3(m); P_4(m); P_5(m)]^T$
- $P_{ij}(m)$ – вероятность перехода из состояния i в j ($i=1, 2, \dots, 5; j=1, 2, \dots, 5$).

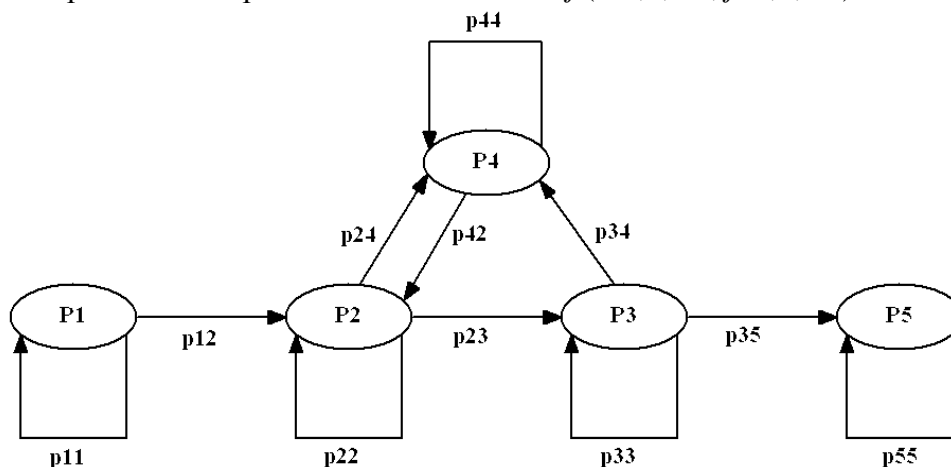


Рис.1. Граф перехода из состояния в состояние по вероятности

Для распределения требований по вероятности работы СПН в том или ином режиме, а также переходных вероятностей в соответствии с графом марковской модели должны выполняться следующие соотношения:

$$\begin{cases} \sum_{i=1}^5 P_i(m) = 1; \\ P_{11}(m) + P_{12}(m) = 1; \\ P_{22}(m) + P_{23}(m) + P_{24}(m) = 1; \\ P_{33}(m) + P_{34}(m) + P_{35}(m) = 1; \\ P_{44}(m) + P_{42}(m) = 1; \\ P_{55}(m) = 1. \end{cases} \quad (3)$$

Таким образом, разностное уравнение, описывающее изменение во времени вектора распределения вероятностей $P(m)$, примет вид:

$$P^T(m+1) = P^T(m) \cdot \pi(m); \quad P^T(0), \quad (4)$$

где $\pi(m)$ – матрица переходных вероятностей состояний (размер 5×5); $P(0)$ – начальное распределение вероятностей нахождения системы в состояниях 1–5.

Матрица $\pi(m)$ в соответствии с графом (рис.1) имеет следующий вид:

$$\pi(m) = \begin{bmatrix} P_{11}(m) & 1-P_{11}(m) & 0 & 0 & 0 \\ 0 & P_{22}(m) & P_{23}(m) & P_{24}(m) & 0 \\ 0 & 0 & P_{33}(m) & P_{34}(m) & P_{35}(m) \\ 0 & P_{42}(m) & 0 & P_{44}(m) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

Решением уравнения (4) является вектор распределения вероятностей состояний на m -м шаге:

$$P^T(m) = P^T(0) \cdot \prod_{i=0}^{m-1} \pi(i), \quad (6)$$

причем каждая компонента вектора $P_i(m)$ определяет вероятность того, что в данный момент времени СПН работает в том или ином режиме, а в целом система работает в режиме наведения как конечном режиме. В этой связи $P_5(m)$ может быть выбрана в качестве вероятностного критерия качества функционирования системы (или в качестве критерия для сравнения оценки качества функционирования различных вариантов построения СПН). Однако следует отметить, что в практических случаях выполнять вычисления по формуле (6) нецелесообразно. Значение $P(m)$ проще вычислять на основе рекуррентной процедуры, которую легче реализовать на ЭВМ. Получение аналитических зависимостей для анализа системы в целом, особенно при высоких порядках, в случае нестационарной модели марковской цепи практически невозможно. В наиболее часто встречающихся случаях почти любая система может считаться стационарной за некоторый промежуток времени (рабочее время). Поэтому проведем ряд упрощающих предположений относительно марковской модели, т.е. будем рассматривать ее как для стационарной системы. Тогда элементы $P_{ij}(m) = P_{ij}$ ($i=1,2,\dots,5$; $j=1,2,\dots,5$) переходной матрицы $\pi(m)$ можно считать постоянными и не зависящими от времени, т.е. $\pi(m) = \pi$. В соответствии с введенными упрощениями операцию произведения матриц, характеризующих нестационарные процессы, можно заменить операцией возведения в степень m матрицы переходных вероятностей с постоянными элементами, а формула (6) примет следующий вид:

$$P^T(m) = \pi^m P^T(0). \quad (7)$$

При этом значение вероятности P_5^* , определяющее время достижения системой этого состояния, может быть принято за пороговое значение, а время достижения этого состояния t_n будет характеризовать быстродействие системы как комплекса. Иными словами, за время t_n с вероятностью P_5^* система (СПН) осуществит успешное решение задачи (например, наведение в упрежденную точку). Однако следует заметить, что изменение $P_5(m)$ во времени, а следовательно, и t_n как время достижения значения пороговой вероятности P_5^* при заданном распределении начальных вероятностей $P_i(0)$ являются функцией значений переходных вероятностей P_{ij} , которые, в свою очередь, зависят от количественных показателей системы в отдельных режимах. Тем самым, изменяя значения переходных вероятностей P_{ij} , можно изменять количественные показатели системы в отдельных режимах и выбрать в качестве критерия значение времени ее работы, отвечающее тактико-техническим характеристикам системы для исходной ситуации, когда $P^T(0) = [1;0;0;0;0]$ (работа системы с режима поиска).

Анализ влияния P_{ij} на время $t_n(P_5^*)$ (если в качестве исходных параметров заданы P_5^* и t_n), позволяет предъявить требования и к динамике системы в виде влияния требуемых значений переходных вероятностей на показатель t_n .

Алгоритм расчета

Алгоритм расчета применяется для вычисления вероятностей состояния системы поиска и наведения в каждый момент времени работы системы. Исходными данными являются значения вероятностей переходов из состояния в состояние, а также количество интервалов дискретизации, определяющее время работы системы. Работа системы начинается с первого режима, что соответствует вероятности этого состояния, равному единице. Значение вероятностей состояний на каждом интервале дискретизации рассчитывается в соответствии с формулой (4). Переход из режима в режим осуществляется при условии, что вероятность состояния, в котором находится система, становится меньше вероятности состояния, в которое может перейти система.

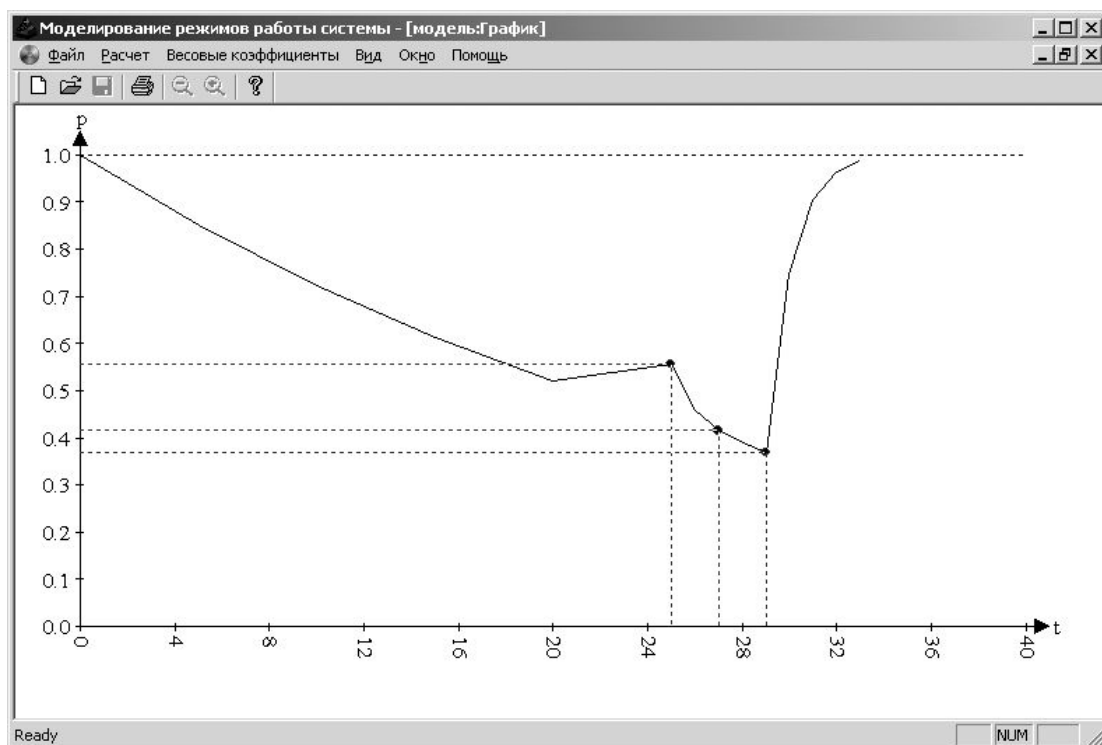


Рис. 2. График вероятности работы системы поиска и наведения

В результате работы программы были получены значения вероятностей состояний на каждом дискретном интервале времени. На рис. 2 представлен график зависимости переходных процессов системы от времени работы системы с учетом весовых коэффициентов состояний. На рис. 3 представлены результаты моделирования в табличной форме. В таблице указывается номер интервала дискретизации и значения вероятностей состояний. Интервалы, в которые система переходит из состояния в состояние, отмечены подчеркиванием.

Interval	p1	p2	p3	p4	p5	sum
0.	1.00000	0.00000	0.00000	0.00000	0.00000	1.00000
1.	0.85000	0.15000	0.00000	0.00000	0.00000	1.00000
2.	0.72250	0.27750	0.00000	0.00000	0.00000	1.00000
3.	0.61413	0.38588	0.00000	0.00000	0.00000	1.00000
4.	0.52201	0.47799	0.00000	0.00000	0.00000	1.00000
5.	0.44371	0.55629	0.00000	0.00000	0.00000	1.00000
6.	0.37715	0.51159	0.05563	0.05563	0.00000	1.00000
7.	0.32058	0.49366	0.10651	0.07925	0.00000	1.00000
8.	0.27249	0.48264	0.15534	0.08952	0.00000	1.00000
9.	0.23162	0.47175	0.20283	0.09380	0.00000	1.00000
10.	0.19687	0.45904	0.24899	0.09509	0.00000	1.00000
11.	0.16734	0.44431	0.29365	0.09469	0.00000	1.00000
12.	0.14224	0.42790	0.33661	0.09325	0.00000	1.00000
13.	0.12091	0.41028	0.37772	0.09110	0.00000	1.00000
14.	0.10277	0.39191	0.41686	0.08846	0.00000	1.00000
15.	0.08735	0.37317	0.41436	0.08551	0.03960	1.00000
16.	0.07425	0.35439	0.41025	0.08214	0.07897	1.00000
17.	0.06311	0.33572	0.40466	0.07856	0.11794	1.00000
18.	0.05365	0.31733	0.39777	0.07488	0.15638	1.00000
19.	0.04560	0.29935	0.38972	0.07116	0.19417	1.00000
20.	0.03876	0.28190	0.38069	0.06746	0.23119	1.00000
21.	0.03295	0.26506	0.37081	0.06382	0.26736	1.00000
22.	0.02800	0.24891	0.36023	0.06027	0.30259	1.00000
23.	0.02380	0.23346	0.34910	0.05683	0.33681	1.00000
24.	0.02023	0.21875	0.33754	0.05351	0.36997	1.00000
25.	0.01720	0.20479	0.32566	0.05032	0.40204	1.00000
26.	0.01462	0.19157	0.31357	0.04727	0.43298	1.00000
27.	0.01243	0.17908	0.30137	0.04436	0.46276	1.00000
28.	0.01056	0.16731	0.28914	0.04159	0.49140	1.00000
29.	0.00898	0.15623	0.27696	0.03897	0.51886	1.00000

Рис. 3. Таблица распределения вероятностей состояний системы на каждом интервале дискретизации m

Заключение

В результате работы была разработана и реализована на ЭВМ математическая модель системы поиска и наведения. Проведенные экспериментальные исследования реальной системы поиска и наведения подтверждают корректность разработанных алгоритмов и моделей проектных решений. Предложенный подход можно применить для моделирования систем поиска и наведения и выбора оптимального проектного решения на этапе проектирования, что позволяет сократить экономические и трудовые затраты на изготовление макета системы и проводимые с ним испытания.

Литература

1. Дж.Кемени, Дж.Снелл, А.Кнепп. Счетные цепи Маркова: Пер. с англ.-М.: Наука, Гл. ред. физ.-мат. лит., 1987. 416 с.
2. Демин А.В., Копорский Н.С., Немолонов О.Ф. Вероятностная модель режимов работы систем поиска и наведения. // Известия ВУЗов. Приборостроение. 1999. Т.42 № 5-6. С.14–18.
3. Демин А.В., Копорский Н.С., Немолонов О.Ф. Имитационное моделирование систем поиска и наведения. Санкт-Петербургский государственный институт точной механики и оптики. Россия. г.Санкт-Петербург. 1998.-II с:25 ил.-9. Библиогр.11. I назв.-Рус.-Деп. В ВИНТИ № 3840-В98 от 23 декабря 1998г.
4. Преснухин Л.Н., Соломонов Л.А., Четвериков В.Н. Шаньгин В.Ф. Основы теории и проектирования вычислительных приборов и машин управления. М.: Высшая школа, 1970. 293 с.
5. Пузырев В.А., Данилевич А.Б., Кочетов Н.В. и др. Применение автоматизированного проектирования для управляющих систем в радиоэлектронике. // Семинар IFAC. Тезисы докладов. - Институт проблем управления, 1980., С.2. (41).
6. Тихонов В.И. Статистическая радиотехника. М.: Радио и связь, 1982.

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ОПТИЧЕСКОГО ПОИСКА

А.В. Кучер

Научный руководитель – д.т.н., профессор А.В. Демин

В статье рассмотрено построение имитационной модели оптического поиска с использованием простого для понимания и реализации средства – массива чисел. Приведен алгоритм модели и результаты моделирования на ЭВМ.

Введение

В работе рассмотрен алгоритм анализа характеристик технической системы поиска в зависимости от ее параметров применительно к выбору проектного решения. Основными характеристиками системы являются время поиска и вероятность правильного обнаружения объекта. Учитывая ограничения системы по времени работы в зоне поиска и предполагая, что данному временному режиму работы и параметрам может отвечать не одно схемотехническое решение, очевидна неоднозначность при выборе решения. В связи с этим в статье рассматривается подход к построению имитационной модели, которая учитывает различные стратегии поиска, возможные события системы и критерии оценки ситуации. При моделировании накапливается статистика, по результатам которой можно выбрать, в соответствии с критерием выбора, оптимальное проектное решение. Выбранная временная характеристика и берется за основу структуры системы, которая в дальнейшем будет реализована. В качестве критерия предложен минимум между математическим ожиданием (МО) расчетного времени поиска для модели с учетом стратегии поиска и имитируемой ситуации и фактически полученным МО времени поиска.

В настоящее время существует ряд работ, посвященных проблемам поиска, но проблема построения модели системы поиска и оптимизации вероятностно-временных характеристик на этапе проектирования системы с использованием имитационного моделирования в целом так и не раскрыта. В работах [1, 2] рассматривается модель системы поиска и наведения как единого целого, что ограничивает применение данной модели для использования принятия проектного решения в других классах систем, использующих систему поиска как подсистему.

Таким образом, целью данной работы является сокращение трудозатрат на поиск схематических решений, наиболее полно соответствующих желаемым характеристикам и параметрам системы, а также сокращение трудозатрат при последующем проведении натурных испытаний. В работе предложенный подход построения имитационной модели рассматривается на примере системы оптического поиска.

Сущность имитационного моделирования

Имитационное моделирование – это численный метод проведения на ЭВМ экспериментов с математическими моделями (ММ), описывающими поведение системы в течение заданного или формируемого периода времени. Здесь фигурируют такие ММ, с помощью которых результат нельзя заранее вычислить или предсказать, поэтому для определения результата необходим эксперимент на модели при исходных данных.

Под процессом имитации будем понимать конструирование модели, ее испытание и применение модели для изучения некоторого явления или проблемы.

При построении системы исследователя интересует, прежде всего, вычисление функционала, характеризующего поведение объекта имитации. Наиболее важный функционал – это показатель эффективности системы, с помощью которого исследователь может оценить различные принципы управления системой, сравнение вариантов схемотехнических решений, определение степени влияния изменений параметров и начальных условий имитации ее поведения.

Основные понятия оптического поиска

Задача оптического поиска объекта в заданной области Ω n -мерного евклидового пространства возникает тогда, когда необходимо обнаружить и определить его местоположение в этой области с помощью систем оптического поиска (СОП).

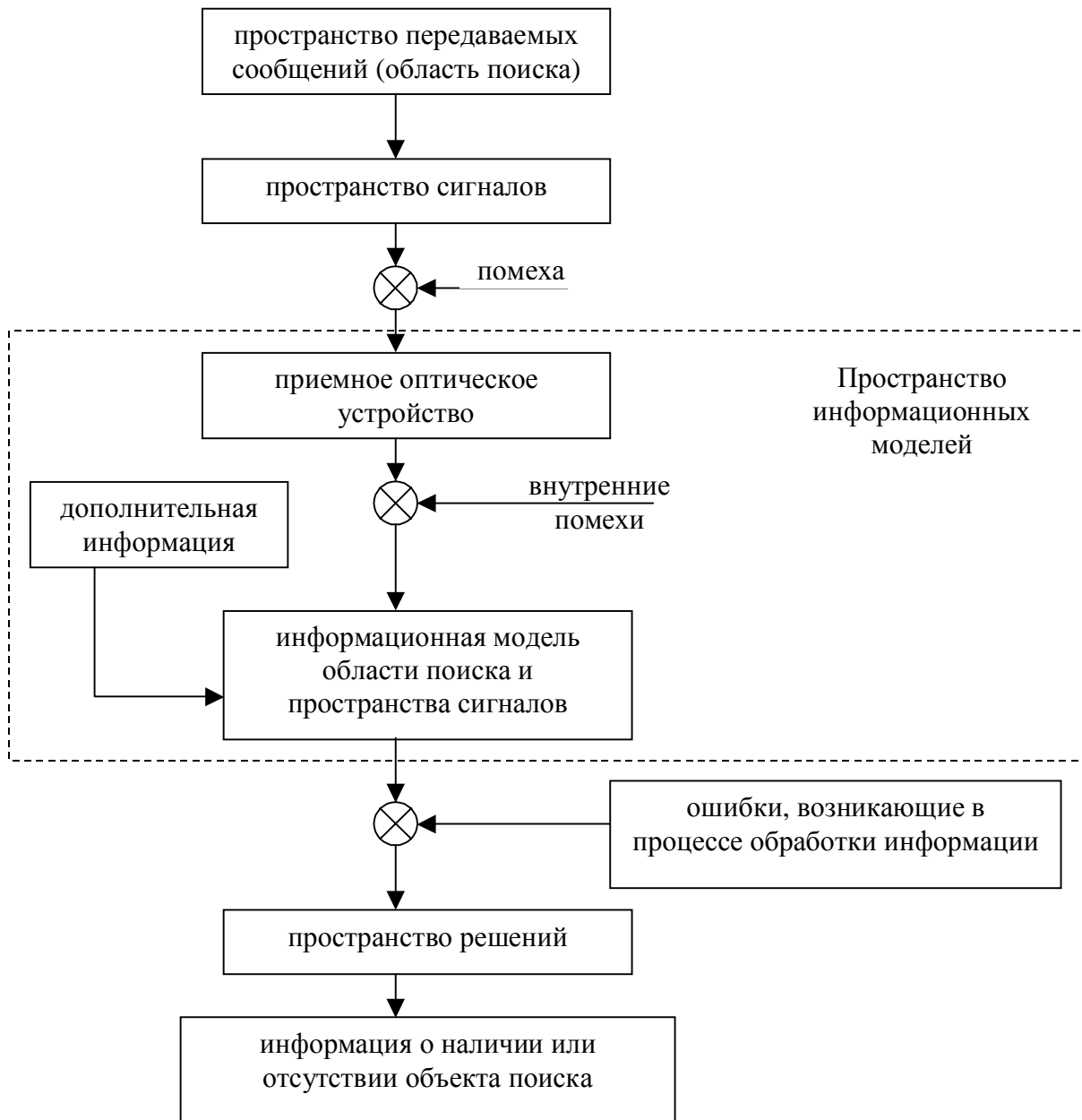


Рис.1. Общая схема получения и обработки информации при оптическом поиске и обнаружении объекта

Движение к объекту не является поиском, если к ней ведет один, заранее известный путь. Множество возможных путей, необходимость выбора среди них – существенная черта любого процесса поиска и, в частности, оптического. Оптический поиск осуществляется в соответствии с некоторым планом, представляющим собой правило выбора последовательности действий – алгоритмом поиска. Способ построения этого правила отражает принципиальную сущность процесса поиска. С каждым выбранным вариантом действий при поиске связан определенный результат как исход этого действия. В ряде случаев этот результат несет в себе только элементарную информацию: объект обнаружен или нет.

Оптическим поиском называется процесс обработки оптической информации наблюдателем, разворачивающийся во времени и направленный на решение задачи обнаружения объекта относительно выбранной системы отсчета при наличии помех и за конечное время. При этом оптический поиск осуществляется в два этапа:

- задание такой кинематики (траектории и скорости) поискового движения, которая обеспечивает попадание объекта в поле зрения системы оптического поиска при известных ограничениях на область и время поиска – этап кинематического обнаружения;
- реализация правила принятия решения о наличии или отсутствии объекта в Ω_n при наблюдении в заданном интервале на фоне помех – этап физического обнаружения.

Если рассматривать оптический поиск как процесс получения и обработки информации, то общая схема этого процесса может быть представлена в виде рис.1.

Вероятностные характеристики системы оптического поиска

Для системы оптического поиска характерны следующие три вероятностные характеристики:

- вероятность обнаружения объекта;
- вероятность ложной тревоги;
- вероятность пропуски объекта.

Тем самым с точки зрения событийности система поиска должна принимать следующие решения:

- объект обнаружен в поле зрения,
- объект не обнаружен в поле зрения.

Но, поскольку поиск процесс является вероятностным, а также в случае, когда решение об обнаружении или необнаружении принимается на пределе чувствительности решающего правила, возможны еще два варианта событий:

- объект пропущен, но он есть в поле зрения;
- ложная тревога, т.е. имеется сообщение об обнаружении объекта, но объекта в поле зрения нет.

Тогда для построения имитационной модели оптического поиска сделаем следующие допущения:

- способы поиска:
 - одноэлементный приемник, т.е. решение принимается по всему полю зрения одновременно;
 - многоэлементный приемник, т.е. поле зрения разбито на число элементов и анализ по элементам производится последовательно (последовательный поиск) или одновременно (параллельный поиск). Если N – число элементов приемника, тогда поле зрения i -го элемента $2\omega_i = \frac{2\omega}{N}$, а время анализа i -го элемента $t_i = \frac{t}{N}$, где t – время анализа всего поля зрения 2ω .
- стратегии поиска:
 - равномерный поиск. Распределение поисковых усилий по просматриваемому пространству происходит равномерно. При этом тщательность просмотра и скорость просмотра в любой из его ячеек одинакова и определяется уровнем расходуемых поисковых усилий: $\varphi(x) = \Phi/X$, где X – область поиска искомого объекта, Φ – затраты поисковых усилий на поиск объекта в области X , $\varphi(x)$ – функция распределения усилий по обследуемой области. Просмотр ячеек осу-

ществляется последовательно одним или несколькими каналами (параллельный поиск);

– априорный поиск. Наличие предварительных сведений о возможном местоположении искомого объекта указывает на то, что тщательность просмотра отдельных участков поля при поиске должна соотноситься с этими сведениями, т.е. распределение поисковых усилий должно быть неравномерным, $\varphi(x) \neq \text{const}$ при $x \in X$, где X представляет собой область поиска искомого объекта, а $\varphi(x)$ – функция распределения усилий по обследуемой области. В простейшем случае неравномерный поиск можно обеспечить квантованием поисковых усилий на 2 уровня 0 и $\bar{\varphi}$. При этом все сводится к тому, что часть исследуемой области $x \in X_1$ просматривается равномерно, где X_1 – область предварительных сведений появления искомого объекта, а в оставшейся части $x \in X - X_1$ никакой поиск не производится, т.е.

$$\varphi(x) = \begin{cases} \bar{\varphi} = \frac{\Phi}{X_1}, & x \in X_1; \\ 0, & x \in X - X_1 \end{cases};$$

– апостериорный поиск. Отличительной особенностью этого алгоритма является его многостадийность. В общем виде многостадийная процедура заключается в следующем: все пространство поиска просматривается равномерно, причем уровень усилий, приходящихся на ячейку, недостаточен для обнаружения объекта с заданной достоверностью, но позволяет сделать некоторое заключение о вероятности наличия или отсутствия объекта в данной ячейке. По результатам равномерного просмотра отбираются ячейки, сигнал в которых превышает выбранный порог. Эти ячейки составляют множество X_2 . На второй стадии просматриваются только отобранные ячейки, и среди них выявляются те, где сигнал превысил некоторый другой порог, и т.д. Возможны различные варианты, но в общем случае выделяются 2 характерные черты:

- на первой стадии отсутствуют априорные данные о положении искомого объекта;
- каждая из последующих стадий строится по результатам предыдущих.

Алгоритм

1. Величину поля зрения системы определим в следующем виде:

$$\|2\omega\| = n.$$

2. Введем операцию разбиения поля зрения на ячейки:

$$K = \left\lceil \frac{2\omega}{\alpha} \right\rceil + 1,$$

где α – наименьший размер объекта на дальнем рубеже.

3. Введем операцию разбиения ячейки на элементы k , число которых определяется в соответствии с критерием Джонсона об опознании объекта, т.е. $k = \{3; 5; 7\}$. Время анализа в ячейке задается в виде разбиения. Например, элемент массива представляется в виде разбиения из k чисел, т.е. $1=0,1+0,2+0,3+0,4-0,6+\dots-0,5$. Каждый элемент разбиения равен условной единицы времени поиска.

4. Величину области поиска определим массивом чисел

$$\Omega = \left\{ \left\lceil \frac{X}{2\omega} \right\rceil + 1 \right\} \times \left\{ \left\lceil \frac{Y}{2\omega} \right\rceil + 1 \right\}$$

где X, Y – координаты области поиска.

На рис. 2 представлено разбиение области поиска на поля зрения.

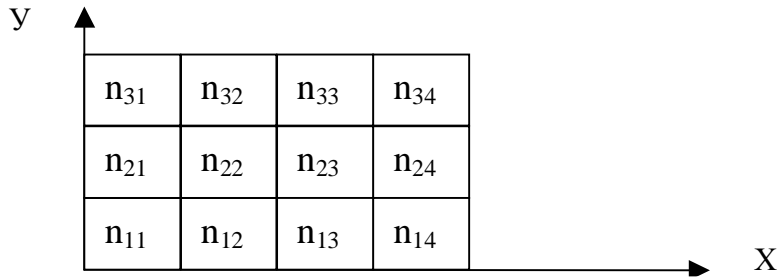


Рис. 2. Разбиение области поиска на поля зрения

Каждое поле зрения разбивается на ячейки, как показано на рис.3.

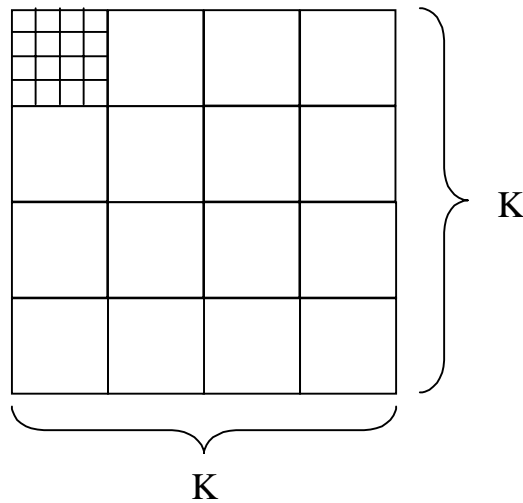


Рис.3. Разбиение поля зрения на ячейки

Таким образом, минимальный массив чисел, подлежащих обработке в ЭВМ, будет иметь размер $N=K^2 \times \Omega$, где $\Omega=n_x \times n_y$, $n_x = \left\lceil \frac{X}{2\omega} \right\rceil + 1$, $n_y = \left\lceil \frac{Y}{2\omega} \right\rceil + 1$. Зададим область поиска в виде массива $AA = a_1, a_2, \dots, a_N$. Каждая ячейка a_i разбивается на элементы, количество которых равно k . Зависимость k от выбранной стратегии поиска описана ниже.

Равномерный поиск – $k = \text{const}$.

Априорный поиск – $k \neq \text{const}$.

Элемент массива	$a[1]$...	$a[p1-r]$	$a[p1]$	$a[p2]$...	$a[p2+r]$...	$a[N]$
Количество элементов разбиения	3	3	5	5	7	7	7	5	5	3	3

В таблице индексы $p1$ и $p2$ определяют область возможных положений искомого объекта, а r – некоторое целое число.

Апостериорный поиск – 1-й шаг $k = \text{const}$, 2-й шаг $k \neq \text{const}$.

5. В соответствии с принятой теорией поиска при моделировании будут инициироваться следующие события:

- объект находится в поле зрения и обнаружен – правильное обнаружение;
- объект находится в поле зрения, но не обнаружен – пропуск объекта поиска;
- объект не находится в поле зрения и не обнаружен;

d) объект не находится в поле зрения, но проходит информация о его обнаружении – ложная тревога.

Задается количество экспериментов, а также параметры системы, а именно – вероятность ложной тревоги $P_{лт}$ и вероятность пропуска объекта $P_{пц}$. Вероятности используются для изменения инициирования событий ложной тревоги и пропуска объекта генератором случайных чисел. Задается также P_k – вероятность попадания объекта в область. Тогда вероятность обнаружения объекта будет определяться по формуле:

$$P_o = P_k (1 - P_{лт} - P_{пц}).$$

В результате моделирования, когда в ЭВМ будет вестись протокол истинности ситуации (объект есть в поле зрения или его нет) и будет набрана статистика по ситуациям, можно получить вероятностные оценки.

б. Ситуации а), б), с) и d) будем имитировать следующим образом.

б.1. Пронумеруем числа, определяющие область поиска. Всем числам массива присваивается значения, равное M , где M – целое положительное число

$$a_i = M, \quad i = 1..N.$$

б.2. Введем операцию представления элемента ячейки в следующем виде

$$M = \sum_1^{k_i} \frac{M}{k_i}.$$

б.3. Введем операцию выбора случайным образом числа из Ω .

б.4. Введем операцию «вбрасывания числа» $\chi = \begin{cases} -v \\ 0 \\ +v \end{cases}$ случайным образом в элемент ячейки.

При этом получаем модели следующих ситуаций:

$$\left\{ \begin{array}{l} M_1 = \sum_1^k \left(\frac{M}{k}\right) \\ M_2 > \sum_1^k \left(\frac{M}{k}\right) + \chi \\ M_3 = \sum_1^k \left(\frac{M}{k}\right) + \chi \\ M_4 < \sum_1^k \left(\frac{M}{k}\right) + \chi \end{array} \right., \quad (1)$$

где M_1 – объекта в ячейке нет (χ не вбрасывается); M_2 – объект в ячейке есть ($\chi = +v$); M_3 – ложная тревога ($\chi = 0$); M_4 – пропуск объекта поиска ($\chi = -v$). Индекс элемента Number массива («номер» ячейки) генерируется случайным образом. Выражение (1) фактически определяет правило принятия решения о наличии или отсутствии объекта поиска.

7. Введем критерий выбора оптимального проектного решения в следующем виде: $\min \{p_o - p_k\}$ и $\min \{M[T_r] - M[T_m]\}$, (2)

где p_o – фактическое распределение плотности вероятности обнаружения объекта, p_k – распределение плотности вероятности объекта в области, $M[T_r]$ – МО времени поиска, полученное в результате моделирования (фактическое), $T_r = \sum_1^{N_o} t_i$ – фактическое время

поиска для эксперимента, $M[T_m]$ – расчетное МО времени поиска для модели,

$T_m = \sum_1^{N_o} t_i$ – расчетное время поиска для эксперимента, N_o – количество ячеек просмотр-

ра, необходимое для получения ответа о наличии либо отсутствии объекта в области поиска, t_i – время просмотра i -й ячейки, i – индекс ячейки, принадлежащей множеству просмотренных ячеек.

8. Индекс ячейки $i=1$. Счетчик условных единиц времени CountTime=0.

9. Для ячейки a_i вычисляется:

а) сумма M_i

$$M_i = \sum_1^{k_i} \frac{M}{k_i}.$$

Если $i = \text{Number}$, тогда $M_i = M_i + \chi_i$;

б) длина m_i ,

$$m_i = k_i.$$

Если $i = \text{Number}$, тогда $m_i = m_i + 1$;

с) счетчик времени CountTime увеличивается на m_i .

10. Если $M_i \geq M$, тогда п.11. В противном случае п.13.

11. Если $m_i \geq k_i$, тогда объект найден.

12. Определяется местоположение объекта в виде 2-х координат ячейки в поле зрения по следующему алгоритму:

А. Индекс поля зрения n_i определим по формуле

$$n_i = \left\lfloor \frac{i}{K^2} \right\rfloor.$$

В. Индекс элемента в массиве, определяющий начало n_i -й ячейки I_n , равен

$$I_n = n_i \times K^2.$$

С. Индексы ячейки в поле зрения вычисляются по следующим формулам:

$$K_x = \left\lfloor \frac{(i - K_n)}{K} \right\rfloor + 1,$$

$$K_y = (i - K_n) - K_x \times K.$$

Д. Индексы поля зрения в области поиска вычисляются по формулам

$$n_{xi} = \left\lfloor \frac{n_i}{n_x} \right\rfloor, n_{yi} = \left\lfloor \frac{n_i}{n_y} \right\rfloor.$$

Процедура поиска прекращается.

13. Переход на следующую ячейку $i=i+1$.

14. Если $i \leq N$, тогда переход к п.8.

15. Если требуется осуществить повторный поиск в области, то переход к п.7.

16. Объект не обнаружен. Процедура поиска прекращается.

Количество эксп.	Вероятность обнаружения объекта	Вероятность ложной тревоги	Вероятность пропуска объекта	Вероятность достоверного ответа	M[t _i] в усл. Ед. времени	M[t _m] в усл. ед. времени
100	0.52	0.04	0.02	0.94	883.65	874.86
1000	0.55	0.027	0.015	0.953	822.22	830.2
5000	0.49	0.012	0.01	0.97	847.60	846.57

Таблица 1. Результаты моделирования – таблица истинности событий модели

Результаты моделирования системы оптического поиска приведены в табл. 1.

Заключение

Алгоритм анализа характеристик системы поиска в зависимости от ее параметров, рассмотренный выше, применительно к выбору проектного решения позволяет промоделировать систему оптического поиска, описываемую исходными параметрами, такими как поле зрения, область поиска, размер объекта и стратегия поиска. Таблица истинности, сформированная на основе правила принятия решения (1), позволяет определить вероятность достоверного ответа, даваемого системой. Для каждого эксперимента рассчитывается теоретическое время поиска, требуемое системе для формирования ответа, и фактическое время поиска. По вычисленным МО экспериментов и критерия выбора оптимального проектного решения (2) можно выбрать решение, отвечающее заданным временным и вероятностным характеристикам.

Литература

1. Демин А.В., Копорский Н.С., Немолочнов О.Ф. Имитационное моделирование систем поиска и наведения. Санкт-Петербургский государственный институт точной механики и оптики. Россия. г. Санкт-Петербург. 1998.-II с:25 ил.-9. Библиогр.11. I назв.-Рус.- Деп. В ВИНТИ № 3840-В98 от 23 декабря 1998г.
2. Демин А.В., Копорский Н.С., Немолочнов О.Ф. Вероятностная модель режимов работы систем поиска и наведения. // Известия ВУЗов. Приборостроение. 1999. Т. 42, № 5-6. С.14–18.
3. Демин А.В., Копорский Н.С., Немолочнов О.Ф., Чиченова Е.А. Имитационное моделирование технических объектов. // Известия ВУЗов. Приборостроение. 2003. Т.46. № 2. С.73–79.
4. Ивахненко А.Г., Юрачковский Ю.П. Моделирование сложных систем по экспериментальным данным. М.: Радио и связь, 1987. 120 с.
5. Максимей И.В. Имитационное моделирование на ЭВМ. М.: Радио и связь. 1988. 232 с.
6. Технология системного моделирования / Е.Ф.Аврамчук, А.А.Вавилов, С.В.Емельянов и др. / Под общей ред. С.В.Емельянова и др. М.: Машиностроение; Берлин: Техник, 1988. 520 с.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ АНОМАЛИЙ В КОМПЬЮТЕРНЫХ СЕТЯХ

М.П. Шалаев

Научный руководитель – д.т.н., профессор И.П. Гуров

Рассмотрены возможности определения потенциально аномальных воздействий на сеть, разработана методика мониторинга и диагностики состояния сети, выполнены экспериментальные исследования предложенных критериев и методов.

Введение

Оперативное определение проблем и превентивные меры по поддержанию работоспособности сети и качества услуг нуждаются как в анализе сетевой топологии на различных уровнях, так и в аналитических моделях нормального и аномального поведения сетей.

По мнению многих современных исследователей в данной области [1–4], в общем случае природа сетевого трафика является случайной, причем статический количественный анализ имеет смысл либо на достаточно коротком промежутке времени, либо для довольно небольших сетевых объединений и не дает полного ответа на вопрос о характере функционирования сети.

Наибольший интерес и значимость имеет задача исследования поведения сетевого трафика с целью разработки системы анализа состояния сети для решения одной из важнейших задач сетевого администрирования, которой является мониторинг маршрутизаторов и другого оборудования магистральных сетей с целью выявления аномального поведения системы или сбоев в работе сети. Аномалии в поведении трафика определяют характер сбоя в сети и могут представлять собой, например, необоснованный рост или падение интенсивности трафика, изменения в стационарном характере трафика, когда наблюдается чрезмерное повышение интенсивности использования отдельных частей сети, и т.д.

Выявление и распознавание аномального поведения сети очень часто основывается на так называемых *ad hoc*, или специальных, субъективных, появившихся в процессе долговременной работы в области управления сетью, методах. За последние несколько лет был разработан ряд как коммерческих, так и свободно распространяемых программных продуктов, предназначенных для сбора статистики и анализа поведения сети [5–7]. Однако область использования данных продуктов ограничивается необходимостью заблаговременно определять различные параметры, такие, как, например, пороговые значения параметров конкретных сетевых соединений или статическое описание нормальных режимов работы сети, а также необходимостью постоянного наблюдения за системой. Очевидно, что чем точнее произведена настройка параметров, тем выше эффективность данных программных средств, что непосредственно зависит от опыта работы администратора с данной сетью и знания особенностей ее поведения. Поэтому необходимо развитие объективных методов анализа состояния сети, и одна из основных проблем при этом состоит в нестационарной природе трафика, которая выражается в появлении сетевых аномалий, т.е. отклонений от нормального режима функционирования.

Аномалии сетевого трафика

Сетевые аномалии можно разделить на следующие основные виды: операционные аномалии, перегрузки сети в рабочем режиме, запрещенные воздействия на сеть.

Операционные аномалии

Данный вид аномалий возникает при выходе из строя сетевых устройств, изменении конфигурации сети, а также при перегрузке в сети и работе на предельных значениях пропускной способности канала.

Аномалии данной категории при графическом отображении трафика визуально отличаются крутым, практически мгновенным изменением в интенсивности трафика, после которого наблюдаются практически постоянные значения интенсивности, отличающиеся от обычных значений в течение наблюдаемого промежутка времени.

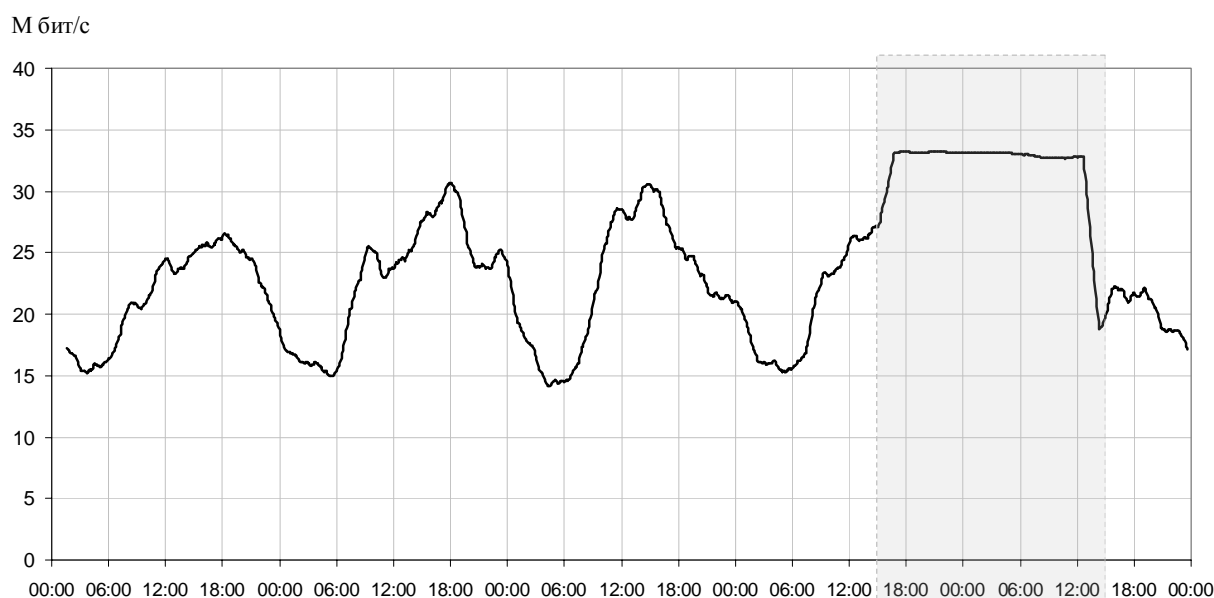


Рис. 1. Пример операционной аномалии входящего трафика магистрального канала распределенной сети

На рис. 1 представлен пример операционной аномалии. На отмеченном участке наблюдаются практически постоянные значения интенсивности входящего трафика магистрального сетевого канала, близкие к его максимальной пропускной способности. Такое поведение сетевого трафика существенным образом отличается от обычного и позволяет предположить наличие сбоя в работе сетевого устройства.

Перегрузки сети в рабочем режиме

В ряде случаев одновременная повышенная потребность в информационных услугах для многих пользователей вызывает перегрузку сети.

Данный вид аномалии обычно проявляется в очень быстром росте отдельной составляющей суммарного трафика, например, FTP, или трафика на определенном маршруте, который затем постепенно спадает до нормального уровня. Перегрузка канала может повлечь за собой потерю пакетов и попытки повторной передачи; сервера не способны обработать растущее количество запросов, что часто приводит к их сбою и необходимости перезагрузки.

Отдельные исследования данного вида аномалий [8] часто концентрируются на попытке оптимизировать работу конкретных сервисов, сталкивающихся с подобными проблемами.

Запрещенные воздействия на сеть

В отличие от предыдущих, данный вид аномалий является наиболее опасным и наименее заметным, так как типичные сценарии запрещенных воздействий на сеть далеко не всегда оказывают существенное видимое влияние на суммарную интенсивность трафика. Наиболее частыми и опасными являются так называемые атаки типа «отказ в обслуживании», или DoS (denial of service) в зарубежных источниках [9]. Суть атаки DoS заключается в том, что атакующий, используя связность сети Интернет, делает попытки вывести из строя определенные сервисы атакуемого сервера, чаще всего, используя при этом так называемый flooding (множественные запросы).

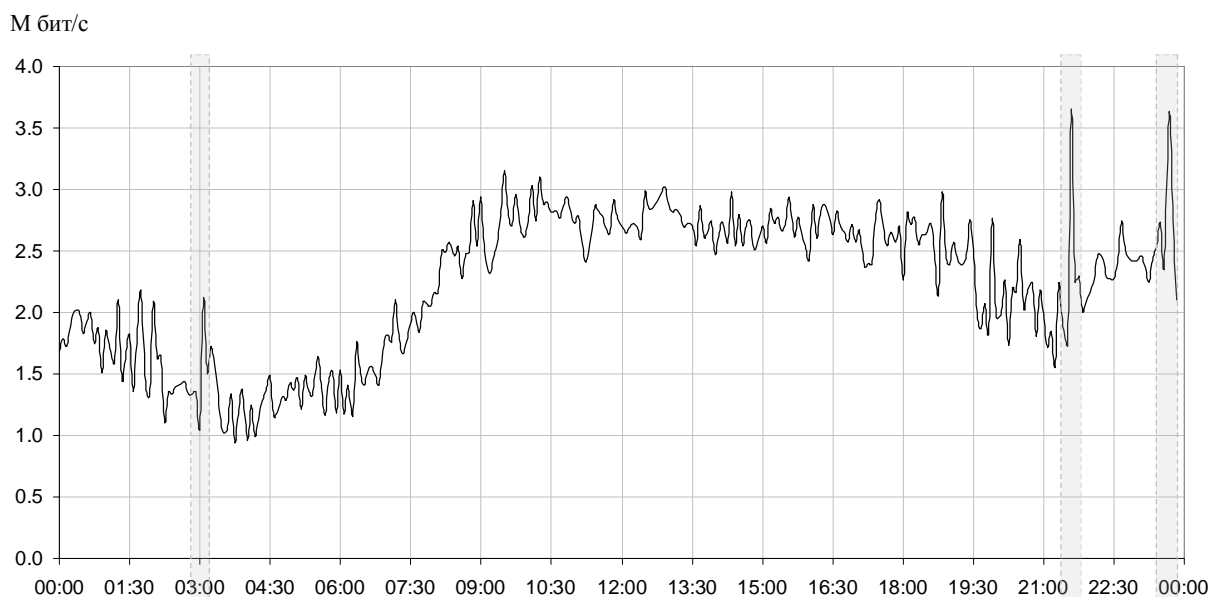


Рис. 2. Пример потенциальных аномалий запрещенных воздействий на сеть в виде отклонений интенсивности входящего WWW трафика международного канала распределенной сети

На рис. 2 представлен пример потенциальных аномалий запрещенных воздействий на сеть. На отмеченных участках наблюдаются кратковременные скачкообразные изменения интенсивности входящего трафика с портом назначения 80, соответствующим трафику WWW серверов. Такое поведение сетевого трафика существенно отличается от стационарного и позволяет предположить наличие потенциальной аномалии запрещенного воздействия на данном сегменте сети.

Подобного рода воздействия на сети наносят существенный экономический ущерб, ставя высокоприоритетной задачей своевременного обнаружения аномалий.

Методы диагностики сетевых аномалий

Рассматривая подходы к анализу и диагностике состояния сети, можно отметить два принципиально отличающихся вида методов диагностики состояния сети.

Первый вид – наиболее часто встречающийся на данный момент – это так называемые ad hoc методы. Принципиальной чертой этих методов является использование некоторых субъективных критериев оценки, исходя из опыта сетевого администратора. Тем не менее, такой подход часто оказывается полезным для определения направлений исследований с учетом особенностей функционирования конкретной сети.

Второй вид методов диагностики состояния IP сети заключается в математическом моделировании процессов, определяющих поведение трафика. При этом основным подходом является использование теории случайных процессов [10–13].

Простейшую модель трафика можно представить как процесс, описывающий поступление дискретных единиц трафика, например, на интерфейс маршрутизатора. При этом выбор характеристики трафика зависит от архитектуры рассматриваемой сети. Например, такой характеристикой может быть IP пакет. Процесс описывается либо с точки зрения частоты поступления пакетов, либо с точки зрения промежутка времени между отдельными появлениями трафика, а именно:

$$\{N(t)\}_{t=0}^{\infty}, N(t) = \max\{n : T_n \leq t\},$$

где $\{N(t)\}_{t=0}^{\infty}$ – непрерывный во времени, неотрицательный целочисленный случайный процесс, а $N(t)$ – количество поступлений пакетов в интервале времени $(0, t]$;

$$\{A_n\}_{n=1}^{\infty}, A_n = T_n - T_{n-1}, \quad (1)$$

где $\{A_n\}_{n=1}^{\infty}$ – вещественная случайная последовательность, а A_n – временной интервал между поступлением n -ого пакета и предыдущего.

В большинстве известных моделей сетевых очередей использовалось предположение о том, что сетевой трафик представляет собой обновляющийся процесс, в котором значения A_n предполагаются независимыми, одинаково распределенными, в общем случае – с любым статистическим распределением. Основным недостатком такого предположения является отсутствие учета корреляционных зависимостей сетевого трафика, наличие которых подтверждено многочисленными исследованиями [10–13].

Во многих случаях в качестве модели трафика используется пуассоновский процесс, при этом $\{A_n\}$ распределены экспоненциально, $P\{A_n \leq t\} = 1 - \exp(-\lambda t)$, при различных значениях параметра интенсивности λ . В случае дискретного времени говорят о процессе Бернулли.

Составной трафик представляет собой групповые появления IP пакетов на интерфейсе сетевого устройства. При этом вводится случайная последовательность $\{B_n\}_{n=1}^{\infty}$, где B_n – случайное количество пакетов в группе. На высоком уровне абстракции B_n может представлять собой какие-либо характеристики n -ой группы.

Очевидно, что пуассоновский процесс превращается в составной нестационарный процесс, если λ зависит от времени и задан закон распределения B_n , не зависящий от A_n .

Достаточно широкое применение модели пуассоновских процессов и их модификаций при описании IP трафика обуславливается рядом простых свойств: суперпозиция пуассоновских процессов будет также являться пуассоновским процессом с интенсивностью, равной сумме интенсивности составляющих; пуассоновский процесс является процессом без памяти; при ряде ограничений мультиплексированные потоки трафика аппроксимируются пуассоновским процессом. Тем не менее, несмотря на ряд преимуществ, данная группа теоретических моделей имеет упомянутые выше недостатки, а именно, отсутствие учета корреляционных зависимостей сетевого трафика, что в общем случае не может гарантировать адекватность модели магистральных сетевых соединений с высокой вероятностью кратковременных интенсивных изменений сетевого трафика.

При моделировании сетевого трафика, характерного для определенного информационного сервиса, например, VBR видео, используются авторегрессивные (AR) модели. Основной особенностью данного вида моделей является однозначное определение зависимости последующих появлений дискретных единиц трафика от предыдущих, например, с помощью детерминированной функции. Наиболее простой является модель линейной авторегрессии порядка p :

$$X_n = a_0 + \sum_{r=1}^p a_r X_{n-r}, n > 0, \quad (2)$$

где (X_{-p+1}, \dots, X_0) – случайный вектор предыдущих значений процесса, $a_r, 0 \leq r \leq p$ – вещественные константы. Рекурсивная форма (2) очевидным образом определяет закон получения каждого последующего значения случайной последовательности $\{X_n\}_{n=0}^{\infty}$ в зависимости от предыдущих значений.

Модель скользящего среднего (СС) порядка q определяется как

$$X_n = \sum_{r=0}^q b_r \varepsilon_{n-r}, n > 0, \quad (3)$$

где b_r , $0 \leq r \leq q$ – вещественные константы, а ε_n – случайные величины с нулевым средним. Комбинация (2) и (3) представляет собой наиболее гибкую модель, называемую моделью авторегрессии – скользящего среднего (АРСС) порядка (p, q) :

$$X_n = a_0 + \sum_{r=1}^p a_r X_{n-r} + \sum_{r=0}^q b_r \varepsilon_{n-r}. \quad (4)$$

Недостатком АРСС моделей является отсутствие объективного критерия выбора порядка модели (p, q) , который определяется на основе априорных сведений о процессе. Если сведения неадекватны, это приводит к значительным погрешностям.

В отличие от описанных выше простейших моделей, а также от моделей, основанных на обновляющих процессах и АР, марковские модели, предложенные автором в [14, 15], обладают важными свойствами, позволяющими учитывать корреляционные зависимости сетевого трафика. Иначе говоря, марковские модели вносят определенные априорные зависимости в случайную последовательность $\{A_n\}$, позволяющие наиболее адекватно учитывать кратковременные интенсивные изменения сетевого трафика, а также применять их при динамическом моделировании трафика магистральных каналов информационных сетей.

Мониторинг и диагностика сети на основе марковских моделей

Методика диагностики состояния сети на основе марковских моделей подробно изложена в работах автора [14, 15]. Кратко суммируя основные идеи, необходимо отметить, что предложенный автором подход основывается на сравнении наблюдаемых характеристик сети с так называемым нормальным профилем.

Определение

Нормальным профилем или шаблоном работы сети назовем такую совокупность характеристик рассматриваемой сети в течение определенного промежутка времени, которая соответствует работе всей системы без каких-либо существенных отклонений от нормы.

Нормальный профиль можно в дальнейшем применять при исследовании реального состояния сети, определяя тем или иным образом различия между предсказанным значением шаблона и проведенными измерениями.

Чтобы определить характер функционирования сети, необходимо обеспечить объективные критерии оценки работы сети. Для этого наблюдаемые значения случайной интенсивности сетевого трафика необходимо каким-либо образом сравнивать с нормальным профилем. При этом используется динамическая процедура так называемого «обучения» модели или, другими словами, эмпирического определения нормального функционирования сети.

Применяя алгоритм «обучения» системы, можно построить модель определения состояния сети на основе «обученной» математической модели системы и реальных данных о трафике, проходящем через маршрутизатор.

Для этого необходимо определить механизм сравнения теоретического шаблона нормальной работы сети с данными, поступающими с маршрутизатора. Рассмотрим так называемое временное «окно» в прошлом или интервал наблюдения размером N временных отсчетов. Накладывая окно выбранного размера на непрерывный поток трафика через маршрутизатор, получим отображение работы сети в интервале времени $[t - N; t]$, где t обозначает текущий момент времени. Положим, что система последовательно случайным образом меняла свое состояние в течение данного промежутка времени, т.е. принимала состояния $\theta_{t-N}, \theta_{t-(N-1)}, \dots, \theta_t$ соответственно в моменты времени $t - N < t - (N - 1) < t - (N - 2) < \dots < t$. Тогда ключевым критерием оценки работы сис-

темы будет являться вероятностью того, что данная последовательность состояний $\theta_{t-N}, \theta_{t-(N-1)}, \dots, \theta_t$ является отображением нормальной работы системы или, что то же, вероятность того, что марковская модель нормального профиля работы системы поддерживает данную последовательность состояний $\theta_{t-N}, \theta_{t-(N-1)}, \dots, \theta_t$ [14, 15].

Указанная вероятность определяется, исходя из предлагаемой математической модели [14, 15], следующим образом:

$$P(\theta_{t-N}, \dots, \theta_t) = P(\theta_{t-N}) \prod_{\mu=t-(N-1)}^t \pi_{k_\mu k_{\mu-1}}, \quad (5)$$

где $P(\theta_{t-N})$ является вероятностью начального состояния системы, π – матрицей одношагового перехода из одного состояния в другое.

Очевидно, что чем выше значение вероятности, вычисленное в (5), тем вероятнее, что последовательность $\theta_{t-N}, \theta_{t-(N-1)}, \dots, \theta_t$ является отображением нормальной работы сети. Аномалии в сети, в свою очередь, должны отрицательно сказаться на полученном значении, тем самым делая вероятность поддержки моделью последовательности очень малой.

Апробирование марковской модели при исследовании IP сети

В работе была детально исследована предложенная модель с целью диагностики состояния и выявления аномального функционирования сети.

Для получения необходимой информации о функционировании рассматриваемой магистральной сети была собрана информация о потоках IP трафика узлового маршрутизатора международного магистрального канала сети RUNNet.

В данной работе использовались данные о трафике, отличающиеся от данных, обычно используемых при подобных исследованиях [3, 4].

Во-первых, данные собирались в течение месяца, 7 дней в неделю и 24 часа в сутки и составляют около 4300 Гбайт, прошедших через маршрутизатор IP пакетов со средней интенсивностью около 16 Мбит в секунду.

Во-вторых, использование в качестве источника данных узлового маршрутизатора глобального объединения сетей, такого, как RUNNet, является очень редкой возможностью и позволяет исследовать поведение сети в целом, а не отдельного соединения.

В-третьих, экспериментальные данные были получены не на основе сбора пакетной информации, а на применении современной технологии коммутации потоков Net-Flow [16], а также с использованием SNMP статистики работы маршрутизатора [3, 4].

В результате эксперимента был получен большой объем данных, из которых было необходимо выбрать последовательность, отвечающую нормальной работе сети, и с ее помощью уже конструировать модель системы.

В результате длительного наблюдения за трафиком на рассматриваемом узле был сделан вывод, что характер нормального профиля трафика носит устойчивую точную периодичность. Более того, при нормальной работе сети трафик ведет себя стационарным образом, инвариантным относительно сдвига на сутки. Таким образом, было экспериментально доказано, что нормальная работа сети характеризуется последовательностью отсчетов трафика, представленной на рис. 3.

Ниже приводятся результаты эксперимента, а именно, тестовые последовательности отсчетов случайной интенсивности трафика с маршрутизатора RUNNet-NORDUnet и вероятности их поддержки разработанной моделью.

Для начала рассмотрим длительную реализацию случайного процесса, представляющую собой последовательность отсчетов интенсивности исходящего трафика RUNNet-NORDUnet вида рис. 4.

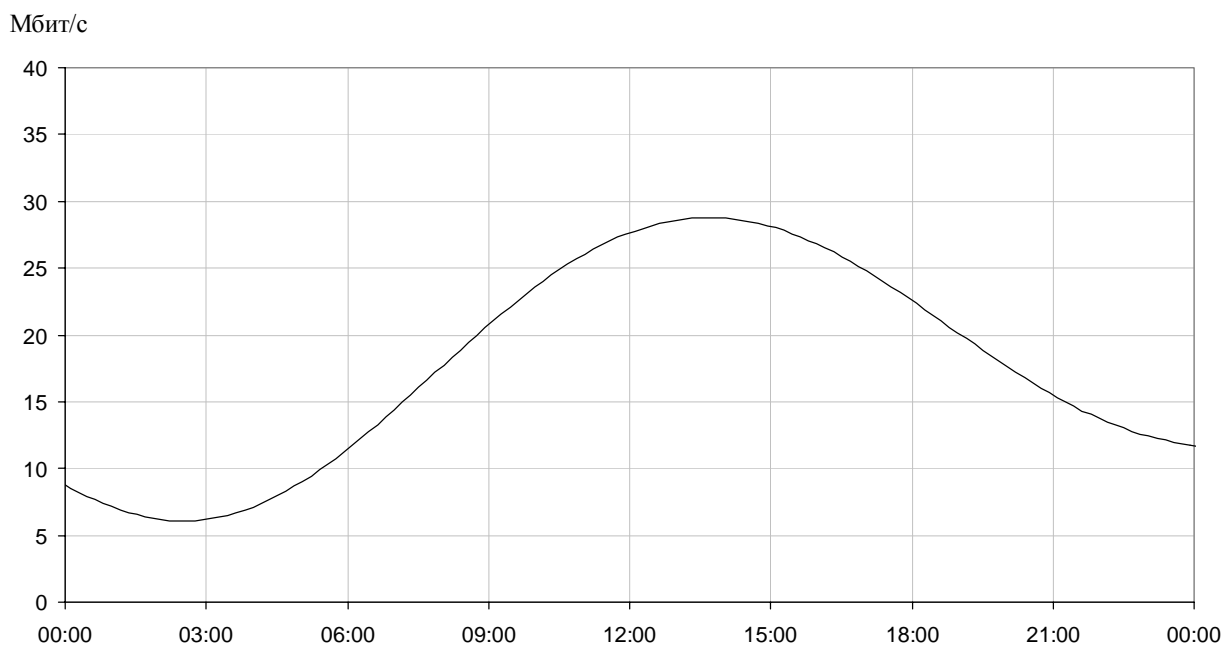


Рис. 3. Нормальный профиль работы сети в виде средней интенсивности входящего трафика международного магистрального канала RUNNet

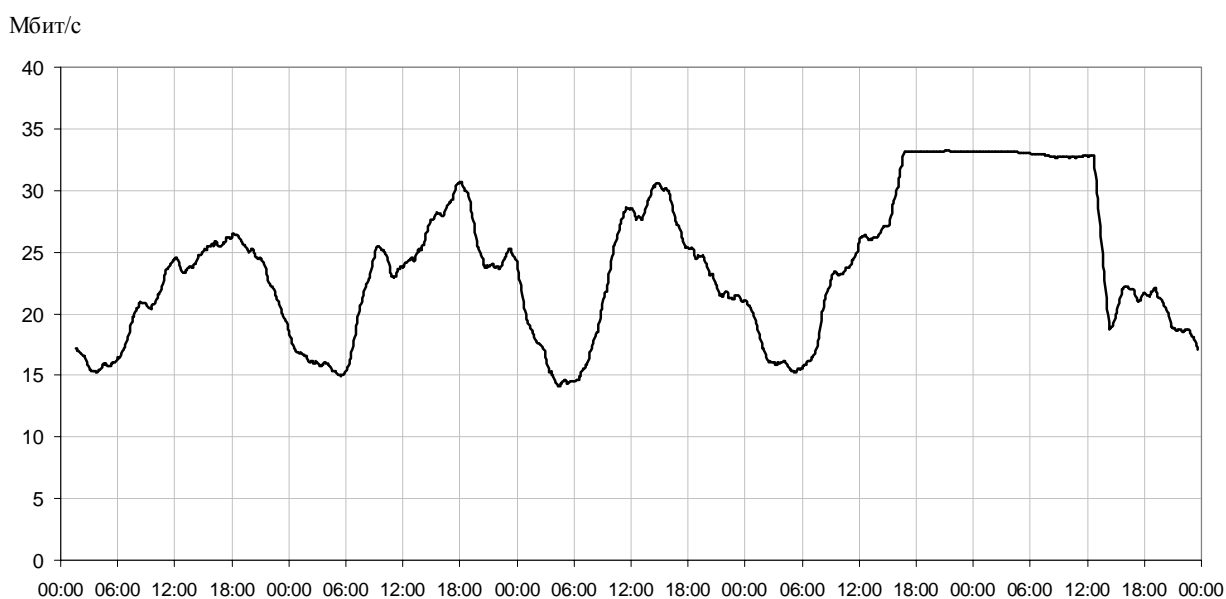


Рис. 4. Экспериментальная последовательность отсчетов интенсивность входящего трафика международного магистрального канала RUNNet

Очевидно, первые три суточных периода представляют отображение нормальной работы сети, а последний период является показателем каких-то проблем, поскольку в течение суток наблюдалась перегрузка канала. Стоит оговориться, что выбор данного примера является случайным, так как, в принципе, достаточно сложно получить подобное отображение поведения сети на реальном канале, поскольку оно встречается крайне редко, и только благодаря экспериментальному подходу, при котором исследуется большое количество данных, было возможно получение подобной тестовой последовательности.

График вероятностей для суммарной последовательности представлен на рис. 5.

На рис. 5. $\log P_s$ вводится для удобства отображения результатов работы модели. P_s – вероятность поддержки моделью системы, определенная соотношением (5).

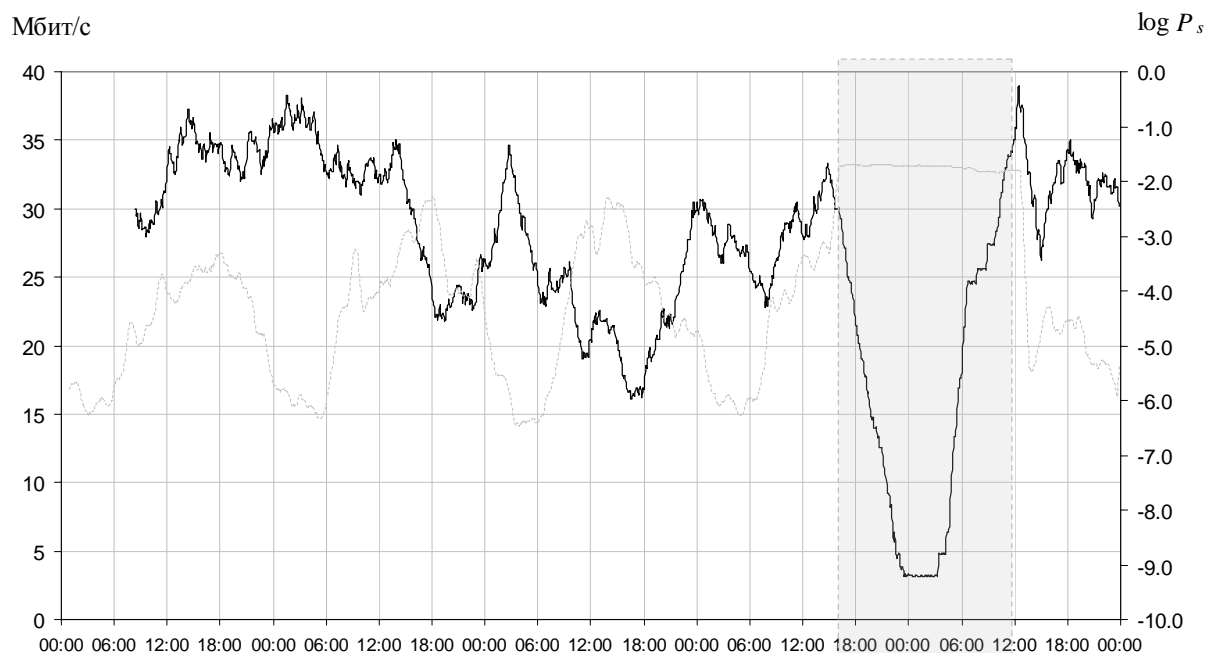


Рис. 5. Вероятность поддержки моделью последовательности отсчетов интенсивности входящего трафика международного магистрального канала RUNNet

На рис. 5. выделена операционная аномалия. Таким образом, можно сделать вывод о том, что вероятности поддержки моделью нормальной последовательности намного выше вероятностей поддержки последовательности, вызванной сбоями функционированием сети. Более того, на данном экспериментальном примере видно, что вероятности поддержки нормальной последовательности отличаются от аномальных так, что достаточно легко выбрать так называемую сигнальную или пороговую вероятность, при достижении которой можно говорить о сбое в сети. Были проведены подробные исследования данной модели, которые также доказывают высказанное предположение.

Заключение

Проведенные экспериментальные исследования узлового маршрутизатора глобального объединения сетей RUNNet позволяют сделать вывод о том, что в результате изучения предложенной теоретической модели экспериментально подтверждена ее адекватность при исследовании больших объемов представительных выборок данных. При этом чувствительность метода обеспечила идентификацию локальных отклонений интенсивности трафика, что позволило экспериментально выявить потенциальные операционные аномалии, а также потенциальные аномалии запрещенного воздействия на сеть.

Литература

1. P. Bradford, D. Plonka. Characteristics of Network Traffic Flow Anomalies / Proceedings of ACM SIGCOMM Internet Measurement Workshop. 2001.
2. P. J. Lizcano et al. MEHARI: a system for analyzing the use of the Internet services. // Computer Networks. 1999. V. 31.
3. S. Uhlig, O. Bonaventure. Implications of Interdomain Traffic Characteristics on Traffic Engineering / In-fonet group, University of Namur, Belgium. 2001.

4. S. Uhlig, O. Bonaventure. Analysis of Interdomain Traffic / Infonet group, University of Namur, Belgium. 2001.
5. Cooperative Association for Internet Data Analysis (CAIDA). Cflowd: traffic flow analysis tool / Technical documentation. 1998.
6. D. Plonka. Flowscan: A network traffic flow reporting and visualization tool / Proceedings of the USENIX Fourteenth System Administration Conference LISA XIV. 2000.
7. The Architecture of CoralReef: An Internet traffic monitoring software suite.
8. J. Jung, B. Krishnamurthy, M. Rabinovich. Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites. // In Proceedings of the World Wide Web Conference. 2002.
9. A. Hussain, J. Heidemann, and C. Papadopoulos. A Framework for Classifying Denial of Service Attacks. / ACM SIGCOMM. 2003.
10. А. Я. Городецкий, В. С. Заборовский. Информатика. Фрактальные процессы в компьютерных сетях: Учеб. пособие. СПб.: Изд-во СПбГТУ, 2000.
11. H. Hlavac, G. Kotsis, C. Steinkellner. Traffic Source Modeling / Institute of Applied Computer Science and Information Systems, University of Vienna. 1999.
12. L. A. Kulkarni, S. Q. Li. Measurement-Based Traffic Modeling: Capturing important statistics // Journal of Stochastic Modeling. 1998. V. 14.
13. A. Sang, S. Q. Li. A Predictability Analysis of Network Traffic // Proceedings of IEEE INFOCOM. 2000.
14. В.Н. Васильев, Ю.В. Гугель, И.П. Гуров, М.П. Шалаев. Анализ характеристик информационного трафика в компьютерных сетях на основе моделей Марковских процессов. // Известия Вузов. Приборостроение. 2003. Т. 46, №8, стр. 19-24.
15. М.П. Шалаев. Моделирование трафика в компьютерных сетях на основе Марковских процессов. // Вестник конференции молодых ученых СПбГУИТМО. Сборник научных трудов 2004. – СПб: СПбГУИТМО, 2004.
16. NetFlow services and applications / Technical documentation. Cisco Systems. 1999.

ТЕХНИЧЕСКИЙ АНАЛИЗ МЕТРИК ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНЫХ САПР

А.А. Зарафьянц

Научный руководитель – д.т.н., профессор Ю.А. Гатчин

В работе исследуется возможность технического анализа метрик процесса для поддержки принятия управленческих решений при разработке программных САПР.

Введение

Метрики программных систем обеспечивают визуализацию проектных процессов. По ним менеджер формирует прогнозы развития и принимает решения о необходимости вмешаться для корректировки хода работ. На измеряемые значения влияют множество факторов, не всегда представляется возможным и рациональным проводить многофакторный анализ для прогнозирования дальнейшего изменения величины метрики.

Существуют различные многофакторные системы, генерирующие числовые метрики. Так, например, аналитики мировых рынков используют прогнозы на основе инструментов технического анализа – построения индикаторов. Задача технического анализа состоит не в обнаружении причин возникновения трендов, а в выявлении самих этих трендов на фоне обычных флуктуаций с помощью построения статистических индикаторов.

В данной статье рассматривается, каким образом инструменты технического анализа могут быть использованы при анализе метрик, связанных с процессом разработки программных систем; формулируются основные условия, накладываемые на метрики для успешного применения методов технического анализа, и приводится пример использования модели из двух индикаторов.

Технический анализ метрик программных систем

Успех создания современных программных САПР, таких как системы проектирования интегральных микросхем, зависит от эффективности усилий команды разработчиков и руководства на всех этапах жизненного цикла проекта. Сложность - неотъемлемая черта современных программных систем [1], борьба с которой является принципиальной задачей системы контроля качества программного обеспечения.

Наиболее успешные сложные программные проекты, согласно опыту [2], обеспечивались серьезной системой контроля качества, основанной на правильно настроенном сборе и анализе различных метрик проекта. Под метриками проекта понимается набор изменяемых во времени численных характеристик. Львиная доля применяемых в настоящее время метрик была изначально разработана для контроля оборонных заказов [3].

К настоящему времени существует десятки различных методик сбора метрик, подробно классифицированных в [4], измеряющих характеристики процессов контроля, качества, управления конфигураций, а также размера и внутренней сложности программного обеспечения.

Условия, предъявляемые к метрикам для использования методов технического анализа, аналогичны условиям, предъявляемым к котировкам. Технический анализ оперирует равновесной ценой, изменяется под воздействием внешних факторов. Величина котировки относительно стабильна при неизменных факторах, изменяется без разрывов. Часто анализируется не сама величина, а ее агрегаты за период, в случае метрик программного обеспечения агрегаты могут браться за день или неделю.

Для построения индикаторов необходима развертка метрики во времени. Необходимо также иметь достаточное число измерений, хотя бы десяток за характерное время

этапа жизненного цикла – наиболее распространенные индикаторы усредняют величины по различному количеству периодов, от единиц до десятков. Таким образом, например, метрики, связанные с вехами проекта, не подходят для технического анализа из-за недостаточного количества данных.

Для технического анализа могут быть выбраны следующие метрики: метрики качества, которые можно агрегировать по неделям, метрики учета изменений в коде, метрики роста объема, сложности программного обеспечения, метрики, связанные с количеством найденных и устраненных дефектов (с группировкой по уровням значимости). Индикаторы, построенные на таких метриках, как метрики объемов коммуникаций по проектам, количество несоответствий стандартам, будут характеризовать состояние дел в организации в целом. И, наконец, на наш взгляд, перспективными кандидатами на применение методов технического анализа являются метрики бюджетных характеристик.

Основы применения методов технического анализа на базе статистических индикаторов котировок подробно рассмотрены в [5–7]. Методы технического анализа могут применяться в поддержке принятия решений, наряду с классическими методами теории принятия решений [9].

Генерация сигналов по набору индикаторов для средней приведенной длительности выполнения задач

На верхнем графике рис. 1 продемонстрирован пример одной из метрик проектов [8]. Исследуется средняя относительная вариация фактической длительности выполнения задач (разработки модулей) по отношению к плановой. Авторы [8] предлагали воспользоваться эмпирически выведенным пороговым критерием для принятия решения о необходимости вмешательства менеджмента, когда вариация составляет более 10 %.

В качестве альтернативного критерия мы предлагаем воспользоваться следующими инструментами технического анализа: конвергенцией–дивергенцией скользящего среднего (MACD) и стохастическим индикатором, показанным в средней и нижней части рис. 1.

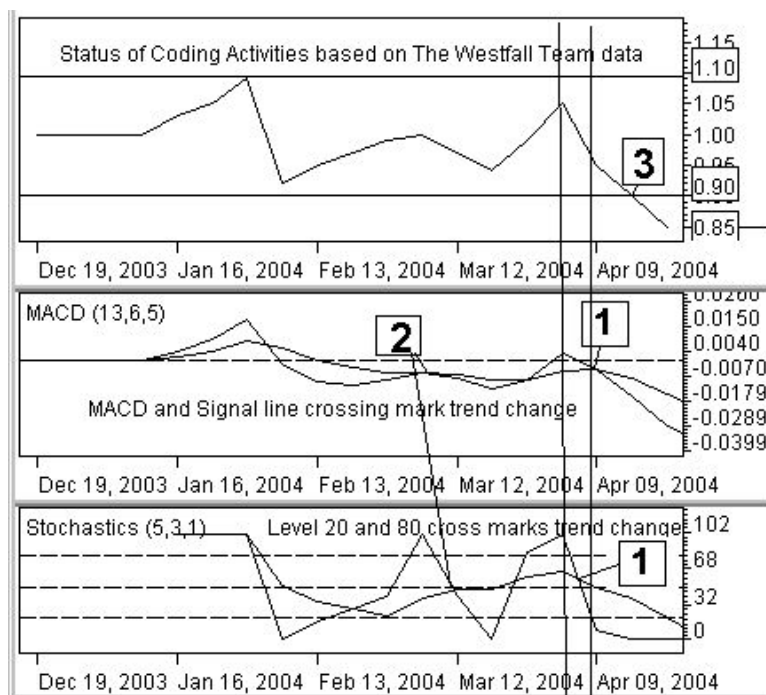


Рис. 1. Пример одной из метрик проектов

Сигналы, генерируемые индикаторами, говорят о возможном изменении трендов. Авторы [8] предлагают пересечение графиком порога в точке 3 в качестве критерия для формирования сигнала менеджеру о необходимости вмешаться и скорректировать процесс. Предлагаемый нами критерий, использующий комбинации двух индикаторов, обнаружит опасное изменение тренда уже в коридоре 1, при срабатывании сигналов по обоим индикаторам, и оповестит менеджмент о надвигающейся опасности на две недели раньше.

Технические срабатывания возможны и в других точках, но использование комбинации индикаторов позволяет избегать появления лишнего сигнала в точке 2.

Заключение

Для практического применения технического анализа при разработке программных систем необходимо снижение издержек на построение индикаторов, поэтому представляется рациональным автоматизировать сбор метрик и построение индикаторов и встраивать системы сигнализации об изменениях трендов метрик на основе построения индикаторов в системы автоматизации жизненного цикла и документооборота проекта. Тем не менее, графический анализ может эффективно проводиться менеджерами и инвесторами, имеющими практический опыт технического анализа.

При соблюдении изложенных в статье ограничений по выбору и методам сбора метрик метод технического анализа метрик способен дать ощутимые результаты для контроля качества программных систем, оказывать руководителям поддержку в принятии корректирующих решений.

Литература

1. Metrics and Models in Software Quality Engineering (2nd Edition) by Stephen H. Kan (Adisson-Wessley, 2002).
2. The mythical man-month by P. Brooks. (Adisson-Wessley, 1995).
3. Software Measurement for DoD Systems: Recommendations for Initial Core Measures / By Carleton, Anita D., et al., CMU/SEI-92-TR-19.
4. Software Metrics Classification by Software measurement LABORATORY.
<http://ivs.cs.uni-magdeburg.de/sw-eng/us/metclas/index.shtml>
5. Intermarket Technical Analysis by John Murphy (John Wiley & Sons, 1991).
6. Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications by John J. Murphy (Prentice Hall Art, 1999).
7. A Complete Guide to Technical Trading Tactics : How to Profit Using Pivot Points, Candlesticks & Other Indicators by John L. Person (Wiley Trading, 2004).
8. Metrics & Measurements Processes by Linda Westfall (Applications in Software Measurement (ASM) Conference, 2003) http://www.westfallteam.com/software_metrics,_measurement_&_analytical_methods.htm
9. Поддержка принятия решений при разработке компьютерных программ. А.А Зараяфьянц. Труды международных научно-технических конференций «Интеллектуальные системы (IEEE AIS'04)» и «Интеллектуальные САПР» (CAD-2004) в 3х томах. Т.1. М. Изд. Физико-математической литературы, 2004, ISBN 5-9221-0531-0.
10. Measuring Customer Satisfaction: Survey Design, Use, and Statistical Analysis Methods, 2nd Edition; Bob E. Hayes; ASQ Quality Press, Milwaukee, Wisconsin; 1998.
11. Measuring the Software Process, Statistical Process Control for Software Process Improvement; William Florac & Anita Carleton; Addison-Wesley, Reading, MA; 1999.
12. Metrics & Models in Software Quality Engineering; Stephen Kan; Addison-Wesley, Reading, MA; 1995.
13. Practical Implementation of Software Metrics; Paul Goodman; McGraw Hill, London; 1993; ISBN 0-07-707665-6.
14. Practical Software Metrics for Project Management and Process Improvement; Robert Grady; PTR Prentice Hall, Englewood Cliffs, New Jersey; 1992.
15. Foundations of Software Measurement by Martin Shepperd Prentice-Hall, 1996) ISBN: 0133361993.
16. Applying Software Metrics, IEEE Computer Society Press, 1997.
17. Software Engineering Measurement by John C. Munson, Auerbach, 2003.
18. Software Measurement Process standard for Software Engineering (ISO/IEC 15939, 2002).

МЕТОДЫ ТЕСТИРОВАНИЯ РЕШЕНИЙ ЗАДАЧ НА СОРЕВНОВАНИЯХ ПО ПРОГРАММИРОВАНИЮ

Г.А. Корнеев, А.С. Станкевич

Научный руководитель – д.т.н., профессор В.Г. Парфенов

В работе рассматривается классификация задач с точки зрения автоматического тестирования и предлагается новый метод автоматического тестирования программ на соревнованиях по программированию.

Введение

В настоящее время широкое распространение получили различные соревнования по программированию, например, этапы Командного студенческого чемпионата мира по программированию (АСМ ICPC), Всероссийские командные олимпиады школьников по программированию. Кроме того, указанные задачи встречаются в современных технологиях дистанционного обучения, в частности, в проекте Интернет–школы программирования (Internet Programming School, IPS), реализуемом на кафедре компьютерных технологий СПбГУ ИТМО [4].

В последнее время, все большую популярность приобретают соревнования реального времени, вытесняющие классические соревнования с отложенной проверкой. Но соревнования такого рода требуют значительно более высокого уровня организации проверки решений участников, так как время, выделяемое на тестирование решения, измеряется минутами. Напротив, в соревнованиях с отложенной проверкой время тестирования решения практически не ограничено.

В данной статье описывается новый подход к тестированию задач, основанный на игровой стратегии. Это означает, что вместо традиционного тестирования на заранее определенном наборе тестов наша система «играет» с проверяемой программой и по результатам данной «игры» определяет общий результат. Такой метод позволяет адекватно оценивать задачи, для которых оптимальное решение практически недостижимо в силу высоких требований к вычислительным ресурсам. Кроме того, он позволяет тестировать управляющие программы, когда автоматическая тестирующая система эмулирует управляемый объект и оценивает качество управления.

Описанный в данной статье метод тестирования был апробирован при проведении реальных соревнований (четверть- и полуфинала Командного студенческого чемпионата мира по программированию), а также используется в тестирующем ядре интернет-школы программирования [4] и системе автоматического проведения интернет-соревнований [3]. Более подробно об этих проектах можно узнать в работе [1]. Краткое описание системы автоматического тестирования, основанной на описываемых методах, может быть также найдено в работе [2].

Автоматическое тестирование

Задача процедуры тестирования программного решения состоит в определении, решает ли тестируемая программа требуемую задачу.

В последнее время широкое применение находят методы, основанные на анализе исходного кода программ. Но для любого достаточно сложного языка программирования (эквивалентного машине Тьюринга) существуют программы, для которых невозможно эффективно определить, решают ли они заданную задачу, или, хотя бы, закончит ли программа выполнение за конечное время.

Так как точное определение корректности программы невозможно, то широкое применение находят различные эмпирические методы тестирования. При этом в зависимости от структуры задачи используются различные методы тестирования.

Назовем метод тестирования *косвенным*, если для определения корректности ответа, получаемый на наборе входных данных, производится проверка принадлежности его некоторому, заранее вычисленному множеству.

Косвенные методы обычно применяются при тестировании задач с единственным решением, а также задач, для которых известны эффективные методы получения всех возможных правильных ответов.

Недостаток косвенных методов состоит в том, что для проверки правильности результата мы должны уметь решать исходную задачу или даже более сложную (в случае, когда решение исходной задачи не единственно).

Назовем метод тестирования *прямым*, если для проверки правильности результата нет необходимости иметь решение исходной задачи.

Классификация задач по отношению к автоматическому тестированию

Рассмотрим следующую классификацию задач.

- 1) По количеству правильных ответов:
 - А) задачи с единственным ответом;
 - В) задачи с конечным множеством правильных ответов;
 - С) задачи с бесконечным множеством правильных ответов.
- 2) По существованию эффективного метода решения:
 - А) для задачи известен эффективный метод получения всех ответов;
 - В) для задачи известен эффективный метод получения некоторого ответа;
 - С) для задачи не известны эффективные методы решения.
- 3) По существованию эффективного метода проверки решения, не включающего решение исходной задачи:
 - А) для задачи известен эффективный метод проверки, не требующий решения исходной задачи;
 - В) такой метод не известен.

Будем обозначать типы задач трехбуквенными сокращениями, где каждая буква является значением соответствующего параметра. На пример, задача типа ВСА имеет конечное множество решений, для которого не известны эффективные методы нахождения хотя бы одного элемента, но существует эффективный метод проверки принадлежности данному множеству.

Очевидно, что не все возможные 18 типов задач имеют смысл. Рассмотрим примеры задач для каждого типа.

ААА. Задача извлечения арифметического квадратного корня из целого числа, являющегося полным квадратом. Для данной задачи существует метод решения, линейный по количеству цифр в исходном числе. Так как ответ единственен, данным методом мы получаем все возможные решения. Посредством возведения полученного решения в квадрат и сравнения с исходным числом производится эффективная проверка правильности результата, не основанная на решении исходной задачи.

ААВ. Задача перемножения двух чисел. Очевидно, задача имеет единственное решение, которое можно получить эффективным методом. При этом единственным разумным методом проверки результата является перемножение исходных чисел.

АВА и АВВ. Так как правильно решение единственно и мы можем эффективно получить некоторое правильное решение, то мы имеем эффективный метод получения всех возможных решений. Таким образом, не существует задач таких типов.

АСА. Задача вычисления дискретного логарифма, т.е. нахождение такого K , что $B = A^K \bmod N$. Если N – простое число, то результат единственен. Для его проверки нужно просто возвести число A в K -ю степень по модулю N .

АСВ. Задача разложения числа на простые множители. Одним из этапов проверки решения является определение простоты числа, которая не может быть произведена эффективно.

ВАА. Представление натурального числа N в виде суммы двух различных натуральных слагаемых. Мощность множества решений равна $\lfloor N/2 \rfloor - 1$. Проверка производится простым сложением полученных слагаемых.

ВАВ. Нахождение минимального пути в графе.

ВВА. Нахождение гамильтонова пути в плотном графе.

ВВВ. Нахождение минимального пути во взвешенном графе.

ВСА. Задача разложения составного числа на пару натуральных множителей, не равных единице. Задача имеет конечное множество правильных ответов, т.е. его мощность не превосходит некоторого числа. Но эффективного метода нахождения хотя бы одного решения неизвестно. При этом, если ответ найден, проверить его не составляет труда – необходимо просто перемножить полученные множители и сравнить результат с исходным числом.

ВСВ. Задача о назначениях.

САА и САВ. Так как все элементы бесконечного множества не могут быть получены, то зада данных типов не существуют.

СВА. Задача нахождения Пифагоровой тройки (натуральных чисел A , B и C , таких, что $A^2 + B^2 = C^2$), все числа которой больше заданного.

ССА. Поиск простого числа, больше заданного.

ССВ. Численное решение «Задачи трех тел».

По отношению к тестированию задачи разбиваются на три класса:

- типы ААА, ААВ, ВАА, ВАВ, АСВ допускают прямое тестирование;
- типы ААА, АСА, ВАА, ВВА, ВСА, СВА, ССА, допускающие косвенное тестирование;
- типы ВВВ, ВСВ, СВВ, ССВ – задачи, не допускающие ни прямых, ни косвенных методов тестирования. Для задач этого класса не существует эффективного метода проверки корректности ответа.

Однако путем наложения дополнительных условий на ответ, задачи типов ВВВ и ССВ зачастую удается преобразовать в задачи с единственным решением, что переводит их в класс задач, допускающих прямое тестирование.

Тестирование на фиксированном наборе тестов

При использовании данного метода тестирования программа считается корректной, если она выдает правильный результат на некотором конечном подмножестве множества всевозможных входных данных.

Данный метод имеет как прямой, так и косвенный вариант, в зависимости от способа проверки ответа.

В прямом варианте полученный ответ проверяется с помощью соответствующей процедуры, а в косвенном ответ сравнивается со всеми правильными ответами для данного набора входных данных (обычно применяется в случае единственности ответа).

Тестирование на наборе случайных тестов

Программа считается корректной, если она выдает правильные ответы на заданном количестве случайных наборов входных данных.

Данный метод обычно используется как косвенный в случаях, когда известен эффективный метод проверки правильности решения, но не нахождения решения как такового.

Тестирование посредством эмуляции

Программе на вход подается некоторая исходная ситуация, после чего она изменяется в соответствии с полученным ответом и вновь подается на вход тестируемой программе. Если после ряда запусков программе удастся добиться требуемого результата, она считается корректной.

Такой метод обычно используется для программ управления или программ для игр (в математическом смысле этого слова), в которых неизвестна оптимальная стратегия. В первом случае тестирующая программа эмулирует управляемый объект, во втором «играет против» тестируемой программы.

Все приведенные методы тестирования могут быть реализованы в виде «диалога» тестирующей и тестируемой программы, что позволяет реализовать единую тестирующую систему, включающую их. Данный подход и был заложен как базовый в автоматическую тестирующую систему PCMS2.

Игровая стратегия тестирования

На данный момент подавляющее большинство систем автоматического тестирования использует потестовый подход. В рамках этого подхода тестируемая программа запускается на заранее определенном наборе тестов, после чего выданный ею результат проверяется. Существенным недостатком такого подхода является невозможность адаптации системы тестов под конкретное решение для определения его сильных и слабых сторон.

При использовании данной стратегии проверка решения рассматривается как игра двух противников – тестирующей системы и тестируемой программы. Первый ход осуществляет тестирующая система (проверяющая программа). Она подготавливает входные данные для запуска тестируемой программы, а также устанавливает ограничения на количество ресурсов, которые будут доступны ей. Далее ход переходит к тестируемой программе. Она запускается на входных данных, подготовленных тестирующей системой, и выдает некоторый результат. При этом осуществляется контроль за объемом использованных ресурсов. Ход вновь получает тестирующая система. Она анализирует результат, полученный тестируемой программой. На основе результатов анализа принимается решение о продолжении тестирования. Если тестирование продолжается, то проверяющая система подготавливает новый набор входных данных, при этом может быть использована информация, полученная от тестируемой программы на предыдущих шагах. После этого ход снова передается тестируемой программе, и так далее.

Игровая стратегия тестирования позволяет повысить гибкость проверки решений и существенно расширяет круг задач, доступных для тестирования.

Очевидно, потестовый подход является подмножеством предложенной стратегии тестирования. При этом информация, полученная при запуске программы участника на одном тесте, никак не используется для проверки других.

То, что в свой ход автоматическая система тестирования может производить произвольные действия (в том числе, и запуск внешних программ), позволяет без дополнительных механизмов осуществлять вероятностное тестирование. При этом тестирующая система создает случайный набор входных данных, удовлетворяющих условию задачи, и передает ход тестируемой программе. Для проверки полученных результатов возможны следующие подходы.

- Тестирующая программа непосредственно проверяет правильность полученного ответа. Применяется при тестировании задач классов ??А.
- Запускается правильное решение (решение жюри). Результаты, выданные тестируемой программой и правильным решением, сравниваются. Применяется для тестирования задач с единственным правильным решением.

- Запускается генератор множества всех правильных решений. Результат, выданный тестируемой программой, проверяется на принадлежность сгенерированному множеству. Применяется для тестирования задач класса ?A?, при небольшой верхней границе мощности множества правильных ответов.
- Запускается решение жюри, и полученный результат сравнивается по оптимальности с результатом, выданным тестируемой программой. Данный подход используется для тестирования оптимизационных задач, для которых точный ответ получить не представляется возможным (например, задачи коммивояжера для плотных графов с большим числом вершин).

Использование последнего из предложенных подходов для тестирования на заранее заданном наборе тестов позволяет осуществлять детерминированное тестирование задач, точное решение для которых неизвестно.

Дополнительной возможностью игровой стратегии тестирования является сравнительное тестирование игровых задач. При этом проверяющая программа действительно играет с тестируемой. Вначале задается исходная позиция в игре, после чего одна из сторон делает первый ход и передает ход другой стороне, которая, в свою очередь, делает свой ход, и т.д. При этом проверяющая программа проверяет корректность производимых ходов обеих сторон и оценивает результаты игры.

Игровая стратегия тестирования позволяет также осуществлять проверку программ управления. При этом проверяющая система в свой ход проверяет корректность управляющих воздействий, предложенных тестируемой программой, и рассчитывает отклик управляемой системы на произведенные воздействия, после чего подает полученный результат на вход тестируемой программе, и т.д. Тестируемая программа признается корректной при достижении требуемого состояния управляемой системы.

Таким образом, в рамках игровой стратегии может быть осуществлен двухуровневый подход к тестированию. На верхнем уровне используется заранее заданный набор тестов. На нижнем уровне производится несколько запусков тестируемой программы с использованием схемы сравнительного тестирования или тестирования программ управления. На верхнем же уровне оценивается результат произведенной последовательности запусков в целом (результат игры, состояние управляемой системы, и т.п.).

Описанные в данном разделе новые методы тестирования не могут быть реализованы в рамках классической потестовой стратегии.

Литература

1. Казаков М.А. Разработка и внедрение системы поддерживающей новые технологии обучения программированию // Телекоммуникации и информатизация образования. 2002 №6. С. 81–100.
2. Корнеев Г.А., Елизаров Р.А. Автоматическое тестирование решений на соревнованиях по программированию // Телекоммуникации и информатизация образования. 2003. №1. С. 61–73.
3. Архив соревнований с системой автоматического проведения online-соревнований // <http://neerc.ifmo.ru/online>
4. Интернет-школа программирования // <http://ips.ifmo.ru>

МЕТОД ПРЕОБРАЗОВАНИЯ ПРОГРАММ В СИСТЕМУ ВЗАИМОДЕЙСТВУЮЩИХ АВТОМАТОВ

Г.А. Корнеев

Научный руководитель – д.т.н., профессор А.А. Шалыто

В работе предлагается формальный метод преобразования программы на алгоритмическом языке высокого уровня в систему взаимодействующих автоматов, обеспечивающую возможность трассировки исходной программы как вперед, так и назад.

Введение

С 1991 г. в России развивается SWITCH-технология, которая базируется на автоматном программировании [1]. При развитии этой технологии встает вопрос о формальных методах построения конечных автоматов или их систем по программам на императивных процедурных языках программирования. В работе [2] был предложен метод преобразования программ, не содержащих рекурсии. В дальнейшем это метод был развит, что позволило преобразовывать рекурсивные программы [3].

Отметим, что в области аппаратного обеспечения эта задача рассматривалась уже в 70-х годах, и ее решение для одной процедуры приведено в работе [4].

Во всех указанных работах строился один автомат, который позволял осуществлять трассировку только в прямом направлении, в то же время возможность трассировки назад в некоторых приложениях является весьма важной [5]. Кроме того, рассматривались только программы, состоящие из одной процедуры.

В работе [6], при участии автора, был предложен метод преобразования программы из произвольного числа рекурсивных процедур в систему взаимодействующих автоматов. Получаемая система автоматов обеспечивает трассировку исходной программы как в прямом, так и в обратном направлении.

В настоящей статье предлагается основанный на работе [6] формальный метод построения такой системы автоматов по программе. Для системы автоматов, построенной в результате применения предлагаемого метода, доказываемся, что трассировка в обоих направлениях осуществляется корректно.

Язык преобразуемой программы

Будем преобразовывать программы, написанные на языке, порождаемом следующей грамматикой:

1	<i>Программа</i>	::=	<i>Процедура</i> <i>Программа</i>
2			<i>Процедура</i>
3	<i>Процедура</i>	::=	<i>Операторы</i>
4	<i>Операторы</i>	::=	<i>Операторы</i> <i>Оператор</i>
5			
6	<i>Оператор</i>	::=	<i>Оператор</i> <i>Присваивания</i>
7			<i>Оператор</i> <i>Ветвления</i>
8			<i>Оператор</i> <i>Цикла</i>
9			<i>Вызов</i> <i>Процедуры</i>
10	<i>Оператор</i> <i>Присваивания</i>	::=	<i>Переменная</i> = <i>Выражение</i>
11	<i>Оператор</i> <i>Ветвления</i>	::=	<i>Выражение</i> <i>Операторы</i> ₁ <i>Операторы</i> ₂
12	<i>Оператор</i> <i>Цикла</i>	::=	<i>Выражение</i> <i>Операторы</i>
13	<i>Вызов</i> <i>Процедуры</i>	::=	<i>Процедура</i>

С одной стороны, данный язык является достаточно простым для формального описания процесса преобразования, а с другой – содержит операторы присваивания,

ветвления, цикла и блочный оператор, и как следствие, является достаточно богатым для описания произвольной программы [7].

Будем предполагать, что при вычислении выражений нет побочных эффектов (значения переменных изменяются только операторами присваивания).

Автоматы и их свойства

Опишем формальный метод преобразования программы в систему взаимодействующих автоматов. Построенная система будет содержать по два автомата для каждой процедуры: *прямой* (для трассировки вперед) и *обратный* (для трассировки назад). Автоматы, построенные по одной процедуре, имеют общее множество состояний, но разные функции переходов [6].

Введем понятие *фрагмент автомата* – набор состояний и переходов. При этом у некоторых переходов может быть не определено начальное или конечное состояние. Такие переходы называются *входами* и *выходами* фрагмента соответственно. Заметим, что один переход может быть входом и выходом одновременно. В дальнейшем будем рассматривать фрагменты автоматов с одним входом и одним выходом.

Рассмотрим дерево синтаксического вывода преобразуемой программы. Для каждой его вершины построим фрагменты прямого и обратного автоматов, соответствующих поддереву с корнем в данной вершине. Построение будем осуществлять в порядке выхода из вершин при обходе дерева в глубину. При этом фрагменты для всех потомков вершины будут построены до построения фрагмента для этой вершины.

Для построенной системы автоматов докажем следующие свойства.

Адекватность – система автоматов выполняет те же действия, что и исходная программа.

Обратимость – при выполнении действий обратного автомата будут восстановлены исходные значения всех переменных при условии, что в точке входа они были такими же, как и в точке выхода при прямом проходе.

Полнота – для каждого состояния, не являющегося конечным, при любых значениях переменных условие на одном из переходов должно быть истинно.

Непротиворечивость – условия на переходах из одного состояния не могут быть истинными одновременно.

Отсутствие недостижимых состояний – любое состояние может быть достигнуто по переходам из начального состояния. При этом состояние называется достижимым, если оно начальное или в него существует переход из достижимого состояния. Таким образом, проверяется только принципиальная возможность достижения состояний, а условия на переходах фактически игнорируются.

Доказательство *адекватности* и *обратимости* позволяет утверждать, что трассировка в обоих направлениях осуществляется корректно. *Полнота* и *непротиворечивость* обеспечивают детерминированность построенных автоматов. *Отсутствие недостижимых состояний* обычно является естественным требованием.

Указанные свойства будем доказывать посредством структурной индукции на дереве вывода преобразуемой программы. Таким образом, предполагая, что эти свойства выполняются для фрагментов, соответствующих потомкам вершины, будем доказывать их для фрагмента, соответствующего вершине.

Для фрагментов автоматов будем рассматривать достижимость состояний от входа фрагмента (напомним, что рассматриваются фрагменты с одним входом и одним выходом). При этом достижимости выхода будет соответствовать достижимости состояния, из которого он выходит.

Для описания состояний автоматов будем использовать нотацию, описываемую следующей грамматикой:

<i>Состояние</i>	::=	<i>ПереходыВ</i> < <i>Действия</i> > <i>ПереходыИз</i>
<i>Действия</i>	::=	<i>Действие</i> ; <i>Действия</i> <i>Действие</i>
<i>ПереходыВ</i>	::=	<i>ПереходВ</i> <i>ПереходВ</i> ; <i>ПереходыВ</i>
<i>ПереходВ</i>	::=	<i>ИмяФрагмента</i> { , <i>Действие</i> }
<i>ПереходыИз</i>	::=	<i>ПереходИз</i> <i>ПереходИз</i> ; <i>ПереходыИз</i>
<i>ПереходИз</i>	::=	<i>Условие</i> , <i>ИмяФрагмента</i>
<i>Действие</i>	::=	<i>Переменная</i> = <i>Выражение</i> <i>push</i> (<i>Выражение</i>) <i>Переменная</i> = <i>pop</i> ()

Заметим, что переход во фрагмент автомата не порождает новый переход, а лишь «закрывает» вход соответствующего фрагмента на описываемое состояние. Это же верно и для перехода из фрагмента.

Выстраиваемая система автоматов использует общий стек, над которым производятся следующие операции:

- *push*(*expr*) – поместить значение выражения *expr* на вершину стека;
- *pop*() – прочитать значение на вершине стека и удалить его.

Если для перехода в состояние указан фрагмент *Вход*, то это переход является входом фрагмента. Аналогично, если для перехода из состояния указан фрагмент *Выход*, то он является выходом фрагмента.

Действие на переходе указывается при определении исходящего перехода. Доказательство свойства обратимости будем осуществлять на границах переходов, когда действия, указанные на переходах уже выполнены.

Начнем описание процесса с отдельных операторов (продукции 10–13), затем перейдем к последовательностям операторов (продукции 4–9), а от них – к процедурам (продукции 1–3).

Преобразование отдельных операторов

Оператор присваивания

Фрагменты прямого и обратного автоматов для оператора присваивания состоят из одного состояния каждый.

Преобразуемая продукция:

10 *ОператорПрисваивания* ::= *Переменная* = *Выражение*

Состояние прямого автомата в используемой нотации имеет вид:

Вход < *push*(*Переменная*) ; *Переменная* = *Выражение* > *true*, *Выход*

При этом состояние обратного автомата имеет вид:

Вход < *Переменная* = *pop*() > *true*, *Выход*

Здесь *true* обозначает истину. Таким образом, оба перехода являются безусловными.

Докажем выполнение рассматриваемых свойств для оператора присваивания. Здесь и далее начало и конец доказательства помечаются значками ► и ◀ соответственно.

Адекватность. ► При выполнении действий в состоянии значение *переменной* сохраняется в стеке, и ей присваивается значение выражения. Таким образом, на выходе из фрагмента, как и после выполнения оператора присваивания, значение *переменной* равно значению выражения. При этом вершина стека содержит значение *переменной* до выполнения присваивания. ◀

Обратимость. ► На выходе из фрагмента обратного автомата значение *переменной* равно значению в вершине стека при входе во фрагмент. По индуктивному предположению на входе фрагмента вершина стека содержит значение, сохраненное при прямом проходе. Таким образом, на выходе переменная имеет значение, которое она имела при входе во фрагмент прямого автомата при прямом проходе, а стек имеет то же содержимое, что и при входе во фрагмент прямого автомата. ◀

Полнота и непротиворечивость. ► Из каждого состояния осуществляется безусловный переход. ◀

Отсутствие недостижимых состояний. ► Вход ведет непосредственно в добавленное состояние. ◀

Оператор ветвления

При преобразовании оператора ветвления добавляются два состояния: входное и выходное [б].

Преобразуемая продукция:

11 *ОператорВетвления* ::= *Выражение* *Операторы*₁ *Операторы*₂

Для прямого автомата добавляются состояния следующего вида:

1: *Вход* < > *Условие*, *Да*; ¬*Условие*, *Нет*

2: *Да*, push(true); *Нет*, push(false) < > true, *Выход*

Для обратного автомата:

1': *Да*; *Нет* < > true, *Выход*

2': *Вход* < > pop(), *Да*'; ¬pop(), *Нет*'

Номера, указанные для состояний, используются исключительно в доказательствах свойств и не переносятся в построенный фрагмент.

Построенные фрагменты ссылаются на фрагменты для операторов, выполняемых при истинности условия – *Да* (построен для ребенка *Операторы*₁) и его ложности – *Нет* (построен для ребенка *Операторы*₂). Фрагменты *Да*' и *Нет*' – соответствующие фрагменты обратных автоматов.

Докажем выполнение рассматриваемых свойств.

Адекватность. ► Если *Условие* истинно, то будет произведен переход во фрагмент *Да*, и на вершине стека будет значение true. В противном случае будет произведен переход во фрагмент *Нет*, а в вершине стека будет значение false. По индуктивному предположению действия, выполняемые фрагментами *Да* и *Нет*, соответствуют *Операторы*₁ и *Операторы*₂. ◀

Обратимость. ► При обратном проходе вход осуществляется в состояние 2'. Если на вершине стека значение находится значение true, то оно снимается со стека и осуществляется переход во фрагмент *Да*'. В противном случае после снятия значения со стека происходит переход во фрагмент *Нет*'.

По индуктивному предположению на входе в вершине стека содержится значение, помещенное туда при прямом проходе. Таким образом, фрагмент обратного автомата выбирается верно. По тому же индуктивному предположению после выхода из фрагментов *Да*' или *Нет*' значения всех переменных будут восстановлены. ◀

Полнота и непротиворечивость. ► Переход из состояний 2 (1') является безусловными. Переходы из состояний 1 (2') помечены взаимно обратными условиями. ◀

Отсутствие недостижимых состояний. ► Вход ведет непосредственно в состояние 1 (2'), из которого выходят переходы во фрагменты *Да* и *Нет* (*Да*' и *Нет*'). По индуктивному предположению от входов этих фрагментов достижимы их выходы. Следовательно, состояние 2 (1') также достижимо. ◀

Оператор цикла

При преобразовании цикла с предусловием к фрагменту добавляется новое состояние.

Преобразуемая продукция:

12 *ОператорЦикла* ::= *Выражение Операторы*

Состояние, добавляемое к прямому автомату:

Вход, `push(false)`; *Операторы*, `push(true)` <
> *Условие*, *Операторы*; \neg *Условие*, *Выход*

Состояние, добавляемое к обратному автомату:

Вход; *Операторы'* <`pop()`, *Операторы'*; \neg `pop()`, *Выход*

Здесь *Операторы* и *Операторы'* — фрагменты прямого и обратного автоматов для тела цикла.

Докажем выполнение рассматриваемых свойств.

Адекватность. ► При входе во фрагмент в вершину стека помещается значение `false`. После этого, пока *Условие* истинно, осуществляется переход во фрагмент *Операторы*, на выходе из которого каждый раз в вершину стека помещается значение `true`. По индуктивному предположению фрагмент *Операторы* адекватен. Таким образом, *Условие* при очередной итерации будет иметь то же значение, что и при исполнении исходной программы. При этом стек будет содержать столько элементов со значением `true`, сколько было итераций цикла, и один элемент со значением `false`. ◀

Обратимость. ► При обратном проходе, пока на вершине стека лежит значение `true`, оно снимается и осуществляется переход во фрагмент *Операторы'*. При этом по индуктивному предположению при каждом попадании в добавленное состояние вершина стека будет содержать значение `true` или `false`, помещенное туда при прямом проходе. Следовательно, во фрагмент *Операторы'* будет осуществлено столько переходов, сколько их было выполнено во фрагмент *Операторы* при прямом проходе.

Во фрагменте обратного автомата на каждом переходе из добавленного состояния значение, записанное при прямом проходе, удаляется из вершины стека. После этого в стеке находятся те же значения, что и были при переходе в добавленное состояние на прямом проходе. ◀

Полнота и непротиворечивость. ► Переходы из добавленных состояний помечены взаимно обратными условиями. ◀

Отсутствие недостижимых состояний. ► Добавленные состояния достижимы непосредственно от входа. При этом один из переходов ведет во фрагмент *Операторы* (*Операторы'*). Следовательно, по индуктивному предположению все состояния этого фрагмента достижимы. ◀

Оператор вызова процедуры

Оператор вызова процедуры преобразуется во фрагмент из одного состояния. При этом предполагается, что любая процедура может быть рекурсивной.

Преобразуемая продукция:

13 *ВызовПроцедуры* ::= *Процедура*

Состояние, добавляемое к прямому автомату:

Вход, `child = Процедура(start)`; *Состояние*
<`child.stepForward()`
 \neg `child.isAtEnd()`, *Состояние*; `child.isAtEnd()`, *Выход*

Состояние, добавляемое к обратному автомату:

Вход, `child = Процедура(end)`; *Состояние*
<`child.stepBackward()`
 \neg `child.isAtStart()`, *Состояние*; `child.isAtStart()`, *Выход*

Здесь *Состояние* – добавляемое состояние; *Процедура*(`start`) – создание экземпляра прямого автомата для процедуры в начальном состоянии; *Процедура*(`end`) – создание экземпляра обратного автомата для процедуры в конечном состоянии; `child.stepForward()` – прямой шаг экземпляра автомата; `child.stepBackward()` – обратный шаг экземпляра автомата.

– обратный шаг экземпляра автомата; `child.isAtStart()` – проверка, что автомат находится в начальном состоянии; `child.isAtEnd()` – проверка, что автомат находится в конечном состоянии.

Докажем выполнение рассматриваемых свойств при дополнительном предположении, что создаваемый экземпляр автомата адекватен и обратим. Это предположение будет доказано в разделе «Завершение доказательства».

Адекватность. ► После входа во фрагмент, пока созданный экземпляр автомата не придет в конечное состояние, производятся шаги вперед созданного экземпляра автомата. По дополнительному предположению это осуществляется корректно. ◀

Обратимость. ► При обратном проходе осуществляются обратные шаги экземпляра автомата до тех пор, пока тот не придет в исходное состояние. По дополнительному предположению после каждого такого шага переменные восстанавливают свои значения. Следовательно, при выходе они примут исходные значения. ◀

Полнота и непротиворечивость. ► Переходы из добавленных состояний помечены взаимно обратными условиями. ◀

Отсутствие недостижимых состояний. ► Вход ведет непосредственно в добавленное состояние. ◀

Преобразование последовательностей операторов

Продукция

5 *Операторы* ::=

определяет пустой оператор, результатом преобразования которого будет отдельный переход (фрагмент без состояний), являющийся одновременно входом и выходом. Для такого фрагмента выполняются все рассматриваемые свойства:

Адекватность и Обратимость. ► Действия не осуществляются. ◀

Полнота и Непротиворечивость. ► Фрагмент не содержит состояний. ◀

Отсутствие недостижимых состояний. ► Вход одновременно является выходом. ◀

В свою очередь, продукция

4 *Операторы* ::= *Операторы* *Оператор*

определяет последовательность операторов. Для преобразования вершины дерева, порожденной этой продукцией, необходимо объединить фрагменты, соответствующие *Операторам* ($\Phi 1$ и $\Phi 1'$) и *Оператору* ($\Phi 2$ и $\Phi 2'$). При этом добавлять новые состояния не требуется, а следует «замкнуть» выход $\Phi 1$ ($\Phi 2'$) на вход $\Phi 2$ ($\Phi 1'$).

Докажем, что при этом все рассматриваемые свойства также выполняются.

Адекватность. ► Выход из фрагмента $\Phi 1$ является входом во фрагмент $\Phi 2$. Таким образом, операторы, преобразованием которых были получены фрагменты $\Phi 1$ и $\Phi 2$, будут выполняться последовательно, и по индуктивному предположению будет получен корректный результат. ◀

Обратимость. ► По индуктивному предположению, после выхода из фрагмента $\Phi 2'$ значения переменных будут восстановлены в промежуточные значения. Следовательно, после выхода из фрагмента $\Phi 1'$ они будут восстановлены в исходные значения. ◀

Полнота и непротиворечивость. ► Состояния не добавляются. ◀

Отсутствие недостижимых состояний. ► По индуктивному предположению выходы фрагментов $\Phi 1$ и $\Phi 2'$ достижимы. Следовательно, достижимы входы фрагментов $\Phi 2$ и $\Phi 1'$. Поэтому достижимы все состояния построенного фрагмента, а также его выход. ◀

Продукции 6–9 введены для удобства записи, и их преобразование совпадает с результатом преобразования левой части соответствующей продукции.

Преобразование процедур

Рассмотрим преобразования вершин дерева, соответствующих продукциям 1-3:

- | | | | |
|---|------------------|-----|----------------------------|
| 1 | <i>Программа</i> | ::= | <i>Процедура Программа</i> |
| 2 | | | <i>Процедура</i> |
| 3 | <i>Процедура</i> | ::= | <i>Операторы</i> |

Продукции 1 и 2 определяют программу как последовательность процедур. Таким образом, результат преобразования программы – множество пар автоматов, получаемых при преобразовании процедур.

Продукция 3 определяет процедуру как последовательность операторов. Как указано в предыдущем разделе, последовательность операторов может быть преобразована во фрагмент автомата. Для построения автомата по процедуре к соответствующим фрагментам требуется добавить по два состояния – начальное и конечное.

Для прямого автомата:

начальное состояние: $\langle \rangle$ true, *Операторы*

конечное состояние: *Операторы* $\langle \rangle$

Для обратного автомата:

начальное состояние: $\langle \rangle$ true, *Операторы'*

конечное состояние: *Операторы'* $\langle \rangle$

Здесь *Операторы* и *Операторы'* – фрагменты прямого и обратного автоматов, соответствующие телу процедуры.

Докажем, что рассматриваемые свойства выполняются для построенного автомата.

Адекватность. ► По индуктивному предположению фрагмент *Операторы* выполняет действия, описанные в теле процедуры. ◀

Обратимость. ► По индуктивному предположению фрагмент *Операторы'* правильно обращает действия, описанные в теле процедуры. ◀

Полнота и непротиворечивость. ► В построенных автоматах из начальных состояний выходят безусловные переходы, а из конечного состояния переходы не выйдут. ◀

Отсутствие недостижимых состояний. ► Из начального состояния непосредственно достижим вход фрагмента *Операторы* (*Операторы'*). Следовательно, по индуктивному предположению достижим и выход этого фрагмента, который ведет в конечное состояние. Таким образом, все состояния построенного автомата достижимы. ◀

Завершение доказательства

Так как для каждой из вершин выполняется индуктивное предположение (при условии, что оно выполняется для всех детей вершины, порожденной этой продукцией), то осталось доказать базу индукции. В дереве вывода листьями могут быть только вершины, соответствующие операторам присваивания и оператору вызова процедуры. Для первого из них было приведено непосредственное доказательство выполнения рассматриваемых свойств.

Доказательства полноты, непротиворечивости и отсутствия недостижимых состояний для оператора вызова процедуры были приведены в явном виде. Следовательно, эти свойства выполняются для всех автоматов.

Доказательства *адекватности* и *обратимости* основывались на предположении истинности этих фактов для вызываемой процедуры в целом. Таким образом, для них доказательства следует дополнить.

► Рассмотрим дерево вызовов при трассировке программы посредством построенных автоматов. По построению, листья этого дерева не содержат вызовов других процедур. Заметим, что если экземпляр автомата не вызывал другие автоматы, то для

него *адекватность* и *обратимость* уже доказана по индукции (так как для этого не требуется доказывать базу для вызовов). Таким образом, у дерева можно «оборвать листья». При этом по доказанному выше все вызовы, соответствующие «оборванным» листьям, являются *адекватными* и *обратимыми*.

Заметим, что, если дерево вызовов конечно, то, последовательно «обрывая листья», можно оставить только корень, для которого также подходит приведенное доказательство. Поэтому в любой момент все выполненные действия являются *адекватными* и *обратимыми*. Таким образом, построенная система автоматов также является *адекватной* и *обратимой*. ◀

Отметим, что при преобразовании каждой вершины дерева во фрагменты автоматов добавляется не более двух состояний. Таким образом, суммарное количество состояний в полученной системе автоматов линейно по количеству вершины в дереве, а, следовательно, и по числу операторов в исходной программе.

Заключение

В работе рассмотрен формальный метод построения системы взаимодействующих автоматов по программе.

Для предложенного метода приведено доказательство *адекватности*, *обратимости*, *полноты*, *непротиворечивости* и *достижимости* всех состояний построенной системы автоматов. Полученная система содержит по два автомата (прямой и обратный) для каждой процедуры. При этом количество состояний в автоматах линейно по количеству операторов в соответствующей процедуре.

Предложенный метод является математической основой создания пакета *Vizi*, предназначенного для создания визуализаторов алгоритмов, используемых при обучении основам программирования и дискретной математики.

Литература

1. Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.
2. Шалыто А.А., Туккель Н.И. Преобразование итеративных алгоритмов в автоматные. // Программирование. 2002. № 5. С.12–26.
3. Туккель Н.И., Шалыто А.А., Шамгунов Н.Н. Реализация рекурсивных алгоритмов на основе автоматного подхода // Телекоммуникации и информатизация образования. 2002. № 5.
4. Баранов С. И. Синтез микропрограммных автоматов (граф-схемы и автоматы). Л.: Энергия, 1979.
5. Казаков М.А., Столяр С.Е. Визуализаторы алгоритмов как элемент технологии преподавания дискретной математики и программирования // Международная научно-методическая конференция «Телематика-2000». СПб.: 2000. С.189–191.
6. Казаков М.А., Корнеев Г.А., Шалыто А.А. Метод построения логики работы визуализатора алгоритмов на основе конечных автоматов // Телекоммуникации и информатизация образования. 2003. №6. С. 27–58.
7. Грис Д. Наука программирования. М.: Мир, 1984.

ТЕХНОЛОГИЯ РАЗРАБОТКИ ВИЗУАЛИЗАТОРОВ АЛГОРИТМОВ

Г.А. Корнеев

Научный руководитель – д.т.н., профессор А.А. Шалыто

Предлагается новая технология разработки визуализаторов, основанная на применении конечных автоматов и поддержанная разработанным автором пакетом *Vizi*.

Введение

При изучении алгоритмов обработки информации, представляемой различными структурами данных [1, 2], важную роль играют визуализаторы алгоритмов, позволяющие в наглядной форме динамически отображать детали их работы. Это открывает возможность использования новой технологии обучения дискретной математике и программирования [3, 4].

Визуализатор – это программа, в процессе работы которой на экране компьютера динамически демонстрируется применение алгоритма к выбранному набору данных. Визуализаторы позволяют изучать работу алгоритмов в пошаговом режиме, аналогичном режиму трассировки программ. Они при необходимости допускают трассировку укрупненными шагами, игнорируя рутинную часть вычислительного процесса, что существенно, например, для переборных алгоритмов.

В настоящей работе рассматривается процесс построения визуализаторов и их частей.

Процесс построения визуализатора алгоритма достаточно сложен, поэтому разобьем его на отдельные составляющие.

В работе рассматриваются основные части визуализатора и особенности их разработки с использованием пакета *Vizi*. Затем описывается порядок разработки визуализатора с использованием пакета *Vizi*. В заключении статьи проводится сравнительный анализ методов построения визуализаторов на основе пакета *Vizi* и без него.

Основные части визуализатора

Визуализатор является сложным программным продуктом. Поэтому выделим в нем основные части, которые можно рассматривать самостоятельно. Тогда процесс построения визуализатора сведется к их построению и интеграции.

Выделение основных частей визуализатора

Целесообразно выделить следующие основные части визуализатора:

- логика визуализатора;
- визуальное представление;
- набор комментариев;
- элементы управления;
- интерфейс визуализатора;
- проектная документация.

Опишем указанные части подробнее.

Логика визуализатора – часть визуализатора, осуществляющая передвижение по алгоритму и предоставляющая данные другим частям визуализатора для отображения их пользователю. Логика визуализатора также определяет *интересные* состояния – те состояния, в которых информация отображается пользователю.

Например, при визуализации алгоритма поиска максимального потока в сети вряд ли стоит выделять в поиске кратчайшего пути между вершинами отдельные состояния. Действия таких «подалгоритмов» обычно отображаются как одна операция. С другой

стороны, в визуализаторе алгоритма Дейкстры поиска кратчайшего пути будет выделено множество интересных состояний.

Логика визуализатора должна обеспечивать возможность движения по алгоритму как вперед, так и назад на неограниченное количество шагов. При этом состояние визуализатора на любом шаге должно оставаться неизменным, вне зависимости от последовательности действий, совершенных пользователем.

Визуальное представление – часть визуализатора, определяющая, что и как будет отображаться пользователю в интересных состояниях. Обычно визуальное представление задается набором изображений (схем изображений), которые отображаются пользователю в процессе визуализации.

Визуальное представление предназначено для облегчения понимания визуализируемого алгоритма. Например, в визуализаторе алгоритма на графах граф может отображаться различными способами: графически (кружками и стрелками), матрицей смежности или списками ребер.

Набор комментариев – часть визуализатора, определяющая, какие сообщения будут выводиться пользователю в интересных состояниях. Комментарий может включать в себя данные, предоставленные логикой визуализатора, и описывать визуальное представление.

Комментарии поясняют текущее действие алгоритма и помогают пользователю глубже и быстрее понять смысл производимых действий, а, следовательно, и сам алгоритм. Хороший набор комментариев позволяет понять алгоритм и без визуального представления.

Элементы управления – часть визуализатора, посредством которой пользователь управляет визуализатором. При этом определяются набор действий, которые пользователь может осуществлять с визуализатором, и связь этих действий с логикой визуализатора. Примерами таких действий могут служить передвижение по алгоритму вперед и назад (при этом могут быть использованы маленькие и большие шаги), изменение количества элементов (вершин в графе, размерность массива), генерация случайных исходных данных.

Интерфейс визуализатора – часть, указывающая, каким образом остальные части визуализатора отображаются на экране (графический интерфейс), а также определяющая способы взаимодействия пользователя с элементами управления (функциональный интерфейс). В нашем случае интерфейс визуализатора должен определять отображение визуального представления и набора комментариев.

Проектная документация отображает все стадии разработки визуализатора и описывает получившийся продукт. Она позволяет легко разобраться во внутренней структуре визуализатора и помогает находить и устранять ошибки.

Разработка основных частей визуализатора

Рассмотрим разработку основных частей визуализатора по отдельности и покажем, как пакет *Vizi* может быть использован для упрощения их разработки.

Заметим, что, несмотря на то, что строится визуализатор алгоритма, часто говорят о визуализируемой программе, а не о визуализируемом алгоритме. Эти понятия существенно различаются.

- *Визуализируемый алгоритм* – алгоритм, который поясняет визуализатор.
- *Визуализируемая программа* – конкретная реализация алгоритма, на основе которой строится визуализатор.

Логика визуализатора разрабатывается для каждого визуализатора в отдельности. При этом наличие логики визуализатора для другого (даже схожего) алгоритма не облегчает разработку логики для нового алгоритма.

В соответствии с назначением логика визуализатора может быть представлена как совокупность двух частей:

- *программа визуализации,*
- *модель данных.*

Программа визуализации служит для передвижения по алгоритму. В соответствии с выдвинутыми требованиями она должна быть обратимой – допускать движение как вперед, так и назад. Программа визуализации также определяет интересные состояния и должны сигнализировать другим частям об их наступлении.

Модель данных – структура данных, в которой хранятся значения всех переменных программы. Она служит для сохранения состояния алгоритма, а также передачи информации от программы визуализации к другим частями визуализатора.

Таким образом, текущее состояние модели данных и точка выполнения программы визуализации (отсчитываемая по динамическим и текстуальным индексам [5]) полностью задают состояние визуализатора.

Пакет *Vizi* облегчает создание обеих частей логики визуализатора. Для этого визуализируемая программа записывается в виде XML-описания, которое полностью повторяет структуру исходной программы.

Программа визуализации строится на основе системы взаимосвязанных конечных автоматов, которая генерируется по XML-описанию визуализируемой программы. При этом для каждой процедуры строятся прямой и обратный автоматы, позволяющие осуществлять шаги вперед и назад.

Модель данных также автоматически генерируется пакетом *Vizi* по XML-описанию визуализируемой программы и является структурой (классом), в которую вынесены переменные, используемые визуализируемой программой.

Визуальное представление шагов алгоритма должно быть выбрано так, чтобы наиболее наглядным образом отобразить состояние визуализируемого алгоритма.

В большинстве случаев не имеет смысла разрабатывать визуальное представление с нуля. Например, во многих случаях требуется отображать одинаковые структуры данных, такие как таблицы, массивы и списки. В этих случаях следует воспользоваться уже разработанными элементами для их отображения, входящими в пакет *Vizi*.

Действия, осуществляемые для создания изображений в интересных состояниях, могут быть указаны в XML-описании визуализируемой программы. Это облегчает программирование.

Визуальное представление разрабатывается после того, как будут определены интересные состояния алгоритма. При этом разработанное визуальное представление может не быть самодостаточным, а требовать дополнительных комментариев. Поэтому и введена возможность отображения комментариев.

Набор комментариев. В отличие от визуального представления и элементов управления, комментарии необходимо разрабатывать «с нуля», так как для каждого алгоритма они уникальны. По комментариям в совокупности с визуальным представлением должно быть понятно, как работает алгоритм и зачем выполняются те или иные действия.

Оперирование с конкретными числами (значениями) во многом помогает разобратся в алгоритме. Поэтому пакет *Vizi* предоставляет средства для включения в комментарии конкретных значений переменных во время исполнения.

Элементы управления. В большинстве визуализаторов используются сходные элементы управления. В частности, все визуализаторы должны содержать кнопки переходов по алгоритму вперед и назад.

В пакете *Vizi* представлены часто используемые элементы управления, интегрированные с автоматным представлением программы визуализатора.

Интерфейс визуализатора. При выполнении нескольких визуализаторов следует придерживаться единого интерфейса. Таким образом, пользователям, освоившим один визуализатор, не составит труда разобраться в остальных.

В пакете *Vizi* реализован единый интерфейс визуализаторов, состоящий из трех областей, расположенных одна над другой:

- область визуального представления;
- область комментариев;
- область элементов управления.

Такой интерфейс делает работу с визуализатором простой и позволяет получать скриншоты визуального представления с комментариями, например для последующей публикации.

Проектная документация является неотъемлемой частью визуализатора и должна разрабатываться одновременно с ним. Более подробную информацию о проектной документации к визуализаторам, созданным на основе пакета *Vizi*, можно найти в работе [7].

Таким образом, пакет *Vizi* позволяет упростить программирование всех основных частей визуализатора, кроме создания проектной документации, что существенно ускорит создание визуализаторов.

Порядок разработки визуализатора

В настоящем разделе рассматривается порядок разработки визуализаторов с использованием пакета *Vizi*.

Пакет *Vizi* позволяет упростить и ускорить процесс разработки визуализатора. Опишем соответствующий порядок разработки визуализатора.

- Анализ литературы.
- Создание визуализируемой программы.
 - Реализация алгоритма.
 - Отладка реализации.
- Разработка концепции визуализатора
 - Выделение интересных состояний.
 - Разработка концепции визуального представления.
 - Разработка набора комментариев.
 - Разработка элементов управления.
- Построение XML-описания визуализируемой программы.
 - Выделение модели данных.
 - Упрощение конструкции программы.
 - Запись XML-описания.
 - Отладка XML-описания.
- Добавление комментариев к XML-описанию.
- Реализация визуального представления.
- Реализация элементов управления.
- Интеграция визуализатора.
 - Генерация кода по XML-описанию.
 - Совместная отладка логики и визуального представления.
- Оформление проектной документации.

На первом этапе производится анализ литературы. При этом рассматриваются существующие модификации алгоритма и одна из них выбирается для визуализации.

На следующем этапе реализуется выбранная модификация алгоритма. Одновременно полученная реализация отлаживается. При этом выделяются шаги алгоритма, требующие особого внимания. Полученная информация используется на следующем этапе. При реализации алгоритма на втором этапе следует избегать применения сложных структурных конструкций, что впоследствии облегчит упрощение программы для записи XML-описания.

На третьем этапе разрабатывается общая концепция визуализатора. Для этого сначала выделяются те шаги алгоритма, которые представляют наибольший интерес, и соответствующие интересные состояния. Затем разрабатывается общая концепция визуального представления и ее конкретизация для каждого выделенного шага. Одновременно с разработкой визуального представления для каждого шага пишутся комментарии, поясняющие действия, выполняемые алгоритмом. После этого разрабатываются элементы управления визуализатором, в частности, определяется, какие параметры и в каких пределах сможет регулировать пользователь.

На четвертом этапе осуществляется построение XML-описания визуализируемой программы.

Для этого сначала выделяется модель данных. При использовании системы визуализации *Vizi* этот шаг существенно упрощается, так как возможность использования локальных переменных и параметров процедур предусмотрена в XML-описании. Таким образом, этот шаг сводится к простому преобразованию заголовков процедур в XML-формат.

После выделения модели данных производится упрощение структуры. При этом программа должна использовать ограниченный набор операторов. После упрощения получается программа, более удобная для преобразования в систему автоматов, так как она не содержит громоздких синтаксических конструкций.

Упрощенная программа может быть непосредственно записана в XML-формате, так как в нем предусмотрены элементы для кодирования всех видов операторов. Программа, записанная в XML-формате, может быть автоматически преобразована пакетом *Vizi* в систему взаимосвязанных автоматов. Полученная программа отлаживается с использованием средств, предоставляемых пакетом *Vizi*.

На пятом этапе к XML-описанию визуализатора добавляются комментарии, разработанные на третьем этапе. Для этого у каждого элемента, соответствующего оператору исходной программы, заполняются поля комментариев и их параметров.

На шестом этапе реализуется концепция визуального представления шагов визуализатора. Этот этап может быть упрощен за счет использования визуальных элементов, входящих в пакет *Vizi*. Для связи визуального представления и XML-описания применяются элементы `draw`.

Реализация элементов управления на седьмом этапе существенно упрощается за счет использования элементов управления, входящих в пакет *Vizi*.

На восьмом этапе осуществляется интеграция визуализатора с использованием средств, предоставляемых пакетом *Vizi*. При этом по XML-описанию автоматически генерируется программа визуализации, модель данных, система отображения комментариев, а также осуществляется интеграция с реализацией визуального представления и стандартными элементами управления. После отладки получается готовый визуализатор.

На заключительном этапе оформляется проектная документация. Заметим, что проектная документация должна вестись все время, пока выполняется проект. На последнем этапе производятся только оформление проектной документации и обобщение полученных результатов.

Заключение

Использование пакета *Vizi* позволяет уменьшить как техническую, так и интеллектуальную сложность разработки визуализатора.

Техническая сложность уменьшается за счет разработки основных частей визуализатора с использованием компонент, входящих в пакет *Vizi*.

Интеллектуальная сложность разработки уменьшается за счет того, что разработанный пакет автоматически строит обращенную версию программы. Ранее построение обращенной версии программы было связано с большими трудностями, и на этом этапе допускалось много ошибок.

Литература

1. Кнут Д. Искусство программирования. Том 1. Основные алгоритмы. М.: Вильямс, 2000.
2. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. М.: МЦНМО, 1999.
3. Казаков М.А., Столяр С.Е. Визуализаторы алгоритмов как элемент технологии преподавания дискретной математики и программирования // Международная научно-методическая конференция «Телематика-2000». СПб, 2000. С.189-191.
4. Корнеев Г.А., Парфенов В.Г., Столяр С.Е., Васильев В.Н. Визуализаторы алгоритмов как основной инструмент технологии преподавания дискретной математики и программирования / Труды международной научно-методической конференции «Телематика-2001». СПб.: СПбГИТМО (ТУ), 2001. С.119-10.
5. Дал У., Дейкстра Э., Хоор К. Структурное программирование. М.: Мир, 1975.
6. Лингер Р., Миллс Х., Уитт Б. Теория и практика структурного программирования. М.: Мир, 1982.
7. Корнеев Г.А., Шалыто А.А. Требования в визуализаторам алгоритмов, выполняемых на базе технологии *Vizi*. http://ctddev.ifmo.ru/vizi/Requirements-4_00.pdf.

МЕТОД ВЕРИФИКАЦИИ ПОДПРОГРАММ, НАПИСАННЫХ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ

К.В. Петров

Научный руководитель – д.т.н., профессор О.Ф. Немолочнов

В статье рассматривается метод верификации подпрограмм, написанных на языках высокого уровня (ЯВУ). Для получения предусловий, необходимых для проверки корректности, при этом используется не исходный текст подпрограммы, а ассемблерный код, восстановленный из объектного модуля.

Введение

В данной статье рассматривается один из возможных методов верификации подпрограмм, написанных на языках высокого уровня (ЯВУ). При таком подходе к верификации можно избежать многих трудностей, связанных с аксиоматизацией языков высокого уровня и получением сложных правил вывода для их разнообразных конструкций. Хотя в данной области достигнуты немалые успехи (это касается и правил вывода для Паскаля, а также для С), многие проблемы остаются актуальными [1]. Предлагаемый в данной статье подход (с использованием ассемблерного кода подпрограммы, полученного из объектного модуля) тоже не лишен значительных трудностей в реализации. Дальнейшее изучение должно показать, являются ли эти сложности настолько серьезными, что данный метод не может иметь практической ценности, или они вполне преодолимы. Кроме того, возможно, что в некоторых случаях этот метод будет уступать традиционному, а в некоторых окажется в преимущественном положении. В настоящей статье приводится только самый обобщенный алгоритм подобной верификации.

Статический анализ и верификация

Статический анализ (называемый также статическим тестированием) включает в себя все виды тестов, проверяющих качество исходных текстов программ. Динамическое тестирование, в свою очередь, включает в себя все виды тестов, применяемых во время выполнения программы. Разница между этими подходами заключается, таким образом, в состоянии программы, которое в данном контексте изменяется компилятором и компоновщиком [5].

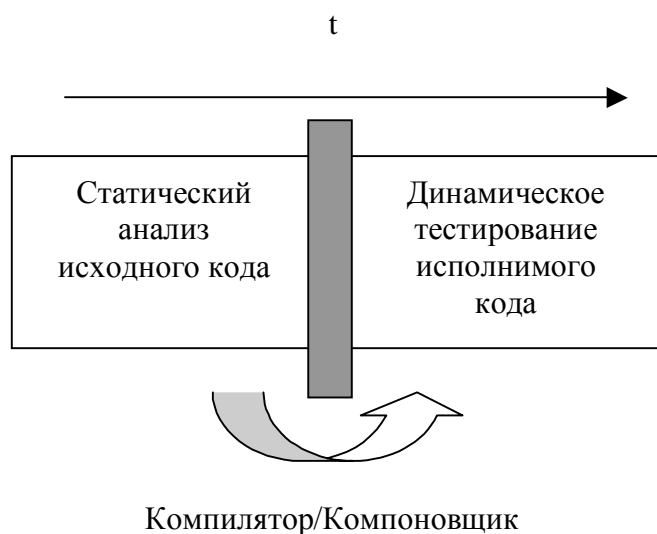


Рис. 1. Статический анализ и динамическое тестирование

Во время компиляции внутреннее (статическое) качество программы становится скрытым. Независимо от количества усилий и материальных средств, потраченных на

динамическое тестирование, внутреннее качество программы остается неизвестным. Когда обнаруженные ошибки исправляются, можно утверждать, что внутреннее качество программы повышается, но до какого уровня – неизвестно. Цикл «редактирование – компиляция – отладка» будет продолжаться до тех пор, пока количество обнаруживаемых ошибок не достигнет приемлемого уровня. Тем не менее, внутреннее качество программы останется неизвестным [2, 5].

Таким образом, традиционный цикл неэффективен в смысле оценки внутреннего качества программного обеспечения, а программы неизвестного качества не могут гарантировать требуемую надежность. Так возникла необходимость развития методов определения внутреннего качества программ, т.е. методов статического анализа, производимого до компиляции исходного текста [2].

Одним из наиболее строгих и трудоемких методов статического анализа является верификация исходного текста, т.е. аналитическое доказательство его корректности. Под корректностью здесь подразумевается соответствие программы ее спецификации (так называемая частичная корректность) [4].

Верификация эффективно доказывает корректность кода, основываясь на низкоуровневой математической спецификации, которая должна быть представлена в некоторой удобной форме. Тщательность анализа требует большого количества времени даже в случае существенной автоматизации. В идеальном случае исследуются все пути исполнения программы, что невозможно при динамическом тестировании [6].

Метод Дейкстры – Гриса. Предусловия и постусловия

Для доказательства правильности программы, сегмента программы или оператора формулируются условия корректности рассматриваемого сегмента. Делается это следующим образом: если конкретное условие истинно непосредственно перед выполнением сегмента программы, тогда после выполнения этого сегмента также истинным будет некоторое условие (в общем случае другое). Такие условия принято называть предусловиями и постусловиями; предусловия и постусловия используют значения переменных из программы [1].



Рис. 2. Предусловия и постусловия

Обычно анализ фрагмента программы начинается с постусловия для этого фрагмента и производится в обратной последовательности, от конца к началу. Если при достижении начала было получено указанное в программе предусловие, правильность фрагмента считается доказанной [1, 4]. Приведем два наиболее простых (и часто применимых) правила получения предусловия из постусловия.

Получение предусловия для оператора присваивания. Чтобы получить предусловие для заданного постусловия P по отношению к заданному оператору присваивания $x:=E$,

подставим выражение E вместо переменной с именем x всюду, где это имя встречается в постуловии P.

$$\{P^X_E\} \quad x:=E \quad \{P\}.$$

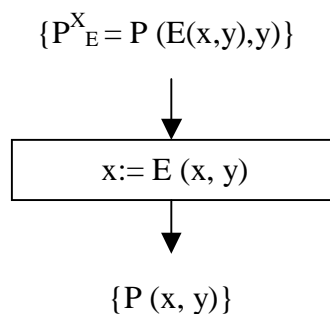


Рис.3. Правило вывода для оператора присваивания

Получение предусловия для последовательности операторов. Если
 и $\{V\} \quad S1 \quad \{P1\}$
 то $\{P1\} \quad S2 \quad \{P\}$
 то $\{V\} \quad (S1;S2) \quad \{P\}$.

Правило вывода S2 очевидным образом обобщается на последовательность операторов произвольной длины [1].

Приведенные выше правила вывода легко применимы, прежде всего, для анализа программы, написанной на языке ассемблера. Именно верификации ассемблерных программ была посвящена выпускная квалификационная работа, написанная автором данной статьи. Если имеется исходный текст верифицируемого модуля, написанный на ассемблере, легко представить команды обработки данных в виде операторов присваивания, например, команду

add op1, op2

как

op1 = op1 + op2.

Конечно, не все ассемблерные команды представимы в виде только одного оператора присваивания, здесь есть много нюансов; но все же верификация программы на языке низкого уровня выглядит существенно более простой задачей, чем верификация программы, написанной на ЯВУ [3].

Применение описанного метода верификации к подпрограммам на ЯВУ

При верификации программ, написанных на ЯВУ, можно столкнуться с целым рядом трудностей, связанных с аксиоматизацией языка. Правила доказательства корректности имеют различную степень сложности для разных конструкций ЯВУ; от языка к языку эта сложность также может варьироваться. Одним из наиболее изученных ЯВУ в этом смысле является Паскаль; уже в 1975 г. была построена автоматическая система верификации для подмножества этого языка, основанная на аксиомах и правилах вывода [3]. Однако, несмотря на успехи в этой области, некоторые проблемы аксиоматизации до сих пор не решены.

Как можно обойти эти трудности при проверке корректности какого-либо фрагмента исходного текста на ЯВУ, например, отдельной подпрограммы? В этом случае представляется вполне резонным следующий порядок действий:

1) Подпрограмма транслируется на язык ассемблера. Этого можно достичь, например, путем дизассемблирования уже оттранслированного модуля, содержащего

анализируемую подпрограмму. В данном случае состояние программы изменяется дважды (ср. рис. 1). Строго говоря, можно анализировать и собственно объектный файл. Однако для большей наглядности метода будем считать, что каким-либо образом мы получаем фрагмент программы на ассемблере, функционально совершенно аналогичный исследуемой подпрограмме.

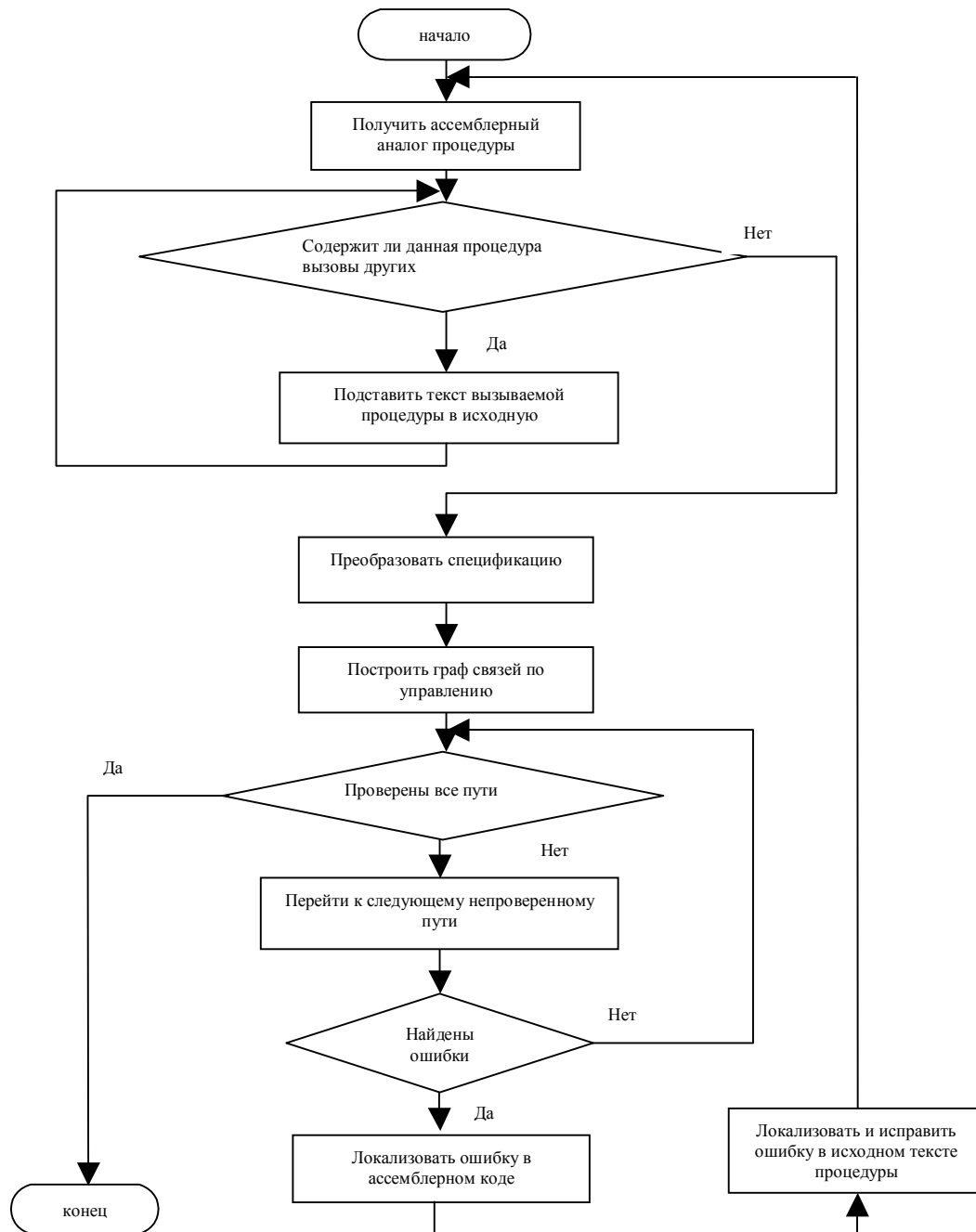


Рис. 4. Обобщенный алгоритм верификации подпрограммы на ЯВУ

2) В случае, если подпрограмма содержит вызовы других подпрограмм (что, как правило, и бывает), то на место вызовов должны быть помещены тексты этих процедур. Этот рекурсивный процесс, конечно, сильно увеличит объем анализируемого кода.

3) На следующем этапе нужно преобразовать спецификацию, относящуюся к исходной подпрограмме, таким образом, чтобы получить спецификацию для фрагмента на ассемблере. Это нетривиальная задача, ведь, например, имена переменных, исполь-

зуемых в исходной подпрограмме, будут в общем случае отсутствовать в ее ассемблерном варианте, и необходимо будет установить соответствие.

4) Для упрощения анализа должен быть получен граф связей по управлению исследуемой процедуры. Это также отдельная, но вполне решаемая задача.

5) Теперь можно получить предусловия для всех путей выполнения подпрограммы. После сравнения результатов с заданными в спецификации можно говорить о корректности или некорректности подпрограммы. В случае полного соответствия все ясно, однако если анализ выявил какие-либо расхождения, необходимо сначала локализовать ошибку в ассемблерном коде, а затем в исходном тексте подпрограммы на ЯВУ. Это тоже довольно сложно.

Итак, предложен алгоритм верификации подпрограммы, написанной на языке программирования высокого уровня (рис. 4). Автор данной статьи использовал в своей выпускной квалификационной работе подобный метод, но с целым рядом упрощений.

1) Анализировался дизассемблер, однако не кода на ЯВУ, а простой ассемблерной программы. Таким образом, дизассемблирование было применено только для общности, чтобы показать, что исходная подпрограмма может быть написана и на другом языке. Очевидно, что проблем адаптации спецификации в данном случае не возникало.

2) Предполагалось, что каким-либо образом уже получен граф связей по управлению исследуемой подпрограммы.

3) Множество применяемых команд ассемблера было сильно ограничено. Кроме того, были введены серьезные ограничения и на разрядность операндов.

Заключение

В данной статье был предложен метод верификации подпрограмм, написанных на ЯВУ. Для получения предусловий, необходимых для проверки корректности, при этом использовался не исходный текст подпрограммы, а ассемблерный код, восстановленный из объектного модуля. Это позволяет обойти многие трудности, возникающие при верификации непосредственно исходного кода на ЯВУ, так как анализировать ассемблерный код гораздо проще. Однако при этом возникает большое количество новых проблем, некоторые из которых кажутся весьма сложными. Дальнейшее изучение данного вопроса должно показать, оправдана ли эта замена одних трудностей другими. Впрочем, здесь многое будет зависеть от конкретного языка программирования высокого уровня и прочих обстоятельств. Тем не менее, на взгляд автора статьи, следует продолжать исследование затронутых проблем, и на этом пути могут быть получены новые интересные результаты.

Литература

1. Бейбер Р. Программное обеспечение без ошибок. М.: Радио и связь, 1996.
2. Дейкстра Э. Дисциплина программирования. М.: Мир, 1978.
3. Пустоваров В. И. Ассемблер. Киев: Ирида, 2000.
4. German A. Software Static Code Analysis Lessons Learned. <http://www.stsc.hill.af.mil/crosstalk/2003/11/0311German.html>
5. Giesen D. A Philosophy and Practical Implementation of Static Analyzer Tools. http://www.dse.nl/~thelosen/artikelen/static_analysis.pdf
6. Peng W. W., Wallace D. R. Software Error Analysis. NIST Special Publication 500-209. <http://hissa.nist.gov/HHRFdata/Artifacts/ITLdoc/209/error.htm>

РЕАЛИЗАЦИЯ МУЛЬТИМЕТОДОВ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C++

Д.Г. Шопырин

Научный руководитель – д.т.н., профессор А.А. Шалыто

Язык программирования C++ не предоставляет стандартных механизмов реализации мультиметодов. Однако известны разнообразные способы эмуляции этой возможности. Данная работа содержит краткое описание одного из таких подходов.

Введение

Мультиметод является общим случаем виртуального метода, или, другими словами, обычный виртуальный метод можно назвать унарным мультиметодом. Вызов виртуального метода зависит от конкретного типа объекта. Вызов же мультиметода зависит от конкретного типа нескольких участвующих в вызове объектов.

Мультиметоды имеют большое прикладное значение. Классическим примером является задача пересечения геометрических фигур. Требуется реализовать функцию, проверяющую, пересекаются ли две геометрические фигуры (в общем случае, две или более фигур). Параметрами данной функции являются два указателя на базовый тип геометрической фигуры (например, Shape). Другими словами, конкретные типы геометрических фигур (*окружность*, *треугольник* и т.д.) неизвестны на момент компиляции программы. Пример использования подобной функции приведен ниже.

```
Shape* one = new Circle(/*parameters*/);  
Shape* two = new Triangle(/*parameters*/);  
bool intersects = CheckIntersection(one, two);
```

Существует два основных подхода к решению данной проблемы. Первый состоит в разработке максимально обобщенной функции пересечения геометрических фигур, что требует написания алгоритмически очень сложного кода.

Другой подход состоит в написании множества *простых* функций, пересекающих заранее известные типы геометрических фигур, например, прямоугольник и треугольник, прямоугольник и окружность и т.д. Очевидно, что написание таких *точечных* функций не сопряжено с решением сложных алгоритмических задач. Проблема состоит в механизме *диспетчеризации* вызовов этих *точечных* функций, или *специализаций*. Мультиметоды как раз и предоставляют необходимый для решения подобных задач механизм диспетчеризации.

Некоторые языки программирования, например Dylan [1], поддерживают мультиметоды напрямую. Однако наиболее распространенные *промышленные* объектно-ориентированные языки программирования, такие как C++, Java и C#, не поддерживают мультиметоды.

Существует два основных подхода к обеспечению поддержки мультиметодов:

- подход, основанный на *нестандартном* расширении компилятора (используется, например, в работе [3]);
- подход, основанный только на стандартных средствах используемого языка программирования (используется, например, в работе [2]).

Ниже приводится краткое описание способа обеспечения поддержки мультиметодов в языке C++ с использованием только стандартных средств этого языка.

Отложенная диспетчеризация

Предложенный способ реализации мультиметодов основан на так называемой *отложенной диспетчеризации* [4]. Отложенная диспетчеризация является расширением широко известного паттерна проектирования Visitor, или *Посетитель* [6]. Каждый

посещаемый объект должен реализовать метод `Visit()`, принимающий в качестве параметра ссылку на специальный объект – *диспетчер*. *Диспетчер* должен иметь набор методов `Dispatch()`, принимающих ссылки на конкретные типы посещаемых объектов.

В случае использования классического паттерна проектирования *Посетитель* возникает циклическая зависимость между *Посетителем* и посещаемыми объектами. Чтобы объявить интерфейс *Посетителя*, необходимо знать список посещаемых объектов. В свою очередь, чтобы объявить посещаемый объект, необходимо знать интерфейс посетителя. *Отложенный Диспетчер* позволяет частично разорвать эту связь и дает возможность объявить и реализовать посещаемые объекты до того, как будет известен интерфейс *Посетителя*.

Интерфейс *Отложенного Диспетчера* и его конкретные реализации генерируются на основе списка типов посещаемых объектов [2, 5]. *Отложенный Диспетчер* используется для определения конкретного типа посещаемого объекта. Каждый конкретный диспетчер принимает в виде шаблонного параметра шаблона [7] тип получателя информации о типе посещенного объекта, что позволяет вкладывать диспетчеры один в другой. Данный механизм позволяет определить тип n объектов, участвующих в вызове n -арного мультиметода. В качестве последнего, $n+1$ -го уровня, указывается класс, позволяющий на основе уже известной информации о типах всех участвующих в вызове мультиметода объектах, вызвать наиболее подходящую специализацию.

Механизм вызова мультиметода схематически изображен на рис. 1.

Реализация

Реализация предлагаемого подхода достаточно сложна. Однако подавляющая часть сложного кода реализована в виде повторно используемой библиотеки классов, что значительно упрощает использование мультиметодов в прикладных приложениях. Библиотека общедоступна по адресу <http://mmdd.sourceforge.net>.

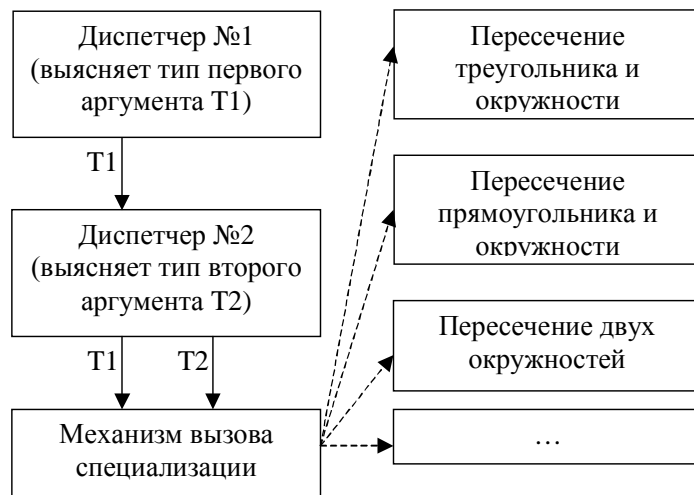


Рис. 1. Механизм вызова бинарного мультиметода

Для реализации мультиметода с использованием предлагаемой библиотеки достаточно следующего:

- все объекты в целевой иерархии должны быть *посещаемыми*, т.е. наследоваться от интерфейса `IVisitable` и перегружать виртуальный метод `Visit()`;
- библиотека должна быть сконфигурирована посредством *конфигурационного класса*, содержащего, в частности, список типов всех участвующих в вызове мультиметода объектов;

- должны быть предоставлены все необходимые специализации мультиметода;
- пользователем должна быть реализована свободная функция, вызывающая мультиметод.

Достоинства и недостатки предлагаемого подхода

Выделим достоинства предлагаемого подхода по сравнению с существующими:

- не используются никакие виды операций приведения типа;
- не используется механизм RTTI (Run-Time Type Identification);
- не используется препроцессор языка C;
- весь исходный код удовлетворяет строгим требованиям безопасности типов;
- поддерживается отдельная компиляция исходного кода;
- время вызова мультиметода константно и зависит только от арности вызываемого мультиметода;
- в процессе вызова мультиметода не выделяется динамическая память;
- не используются нестандартные библиотеки;
- используется только *стандартное* подмножество языка C++.

Основным недостатком предлагаемого подхода является экспоненциальный рост бинарного кода, генерируемого компилятором. Это происходит потому, что на этапе компиляции учитываются все возможные комбинации вызова мультиметода. Например, для вызова бинарного мультиметода при количестве различных типов, равном n , количество комбинаций будет равно n^2 . Однако этот отрицательный фактор смягчается возможностью отдельной компиляции. Весь *экспоненциальный* код сосредоточен в пределах одной единицы компиляции и не влияет на скорость компиляции остального кода.

Заключение

Предлагаемый подход позволяет эмулировать поддержку мультиметодов в языке программирования C++. При этом, предлагаемый подход использует только *хорошие приемы* программирования на этом языке. Например, не используются такие средства как препроцессор языка C и небезопасные приведения типа в стиле C. Более того, не используется механизм RTTI, что позволяет гарантировать строгую безопасность типов.

Предлагаемый подход обеспечивает высокую скорость вызова мультиметода вследствие того, что не выделяется динамическая память. Кроме того, время вызова предсказуемо, константно и зависит только от арности мультиметода.

Отдельная компиляция позволяет локализовать реализацию мультиметода, не затрудняя компиляцию остального кода системы. Использование только стандартных средств языка C++ делает предлагаемый подход доступным для пользователей различных реализаций компилятора языка C++.

Литература

1. Dylan Programming: An Object-Oriented and Dynamic Language / Feinberg N., Keene S., Mathews R., Withington T., Mathews R. Addison-Wesley, 1996. 412 с.
2. Александреску А. Современное проектирование на C++: Обобщенное программирование и прикладные шаблоны проектирования. М.: Вильямс, 2002. 336 с.
3. Baker J., Hsieh W. Maya: multiple-dispatch syntax extension in Java // Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation, 2002. P. 270–281.
4. Shopyrin D. MultiMethods via Deferred Dispatch <http://www.codeproject.com/cpp/mmvdd.asp>, 2004.
5. Shopyrin D. MultiMethods in C++: Finding a complete solution <http://www.codeproject.com/cpp/mmcppfcs.asp>, 2004.
6. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. СПб.: Питер, 2001. 368 с.
7. Vandevorde D., Josuttis N. C++ Templates: The Complete Guide .Addison-Wesley, 2003.

ОБОСНОВАНИЕ ВЫБОРА ЯЗЫКА МОДЕЛИРОВАНИЯ ДЛЯ ИМИТАЦИИ ПОВЕДЕНИЯ ОПЫТНОГО ПРОИЗВОДСТВА

И.В. Каунов

Научный руководитель – д.э.н., профессор О.В. Васюхин

Рассматриваются возможности специализированных языков моделирования для построения систем имитационного моделирования деятельности предприятия. Особое внимание уделяется языку GPSS.

Производственная и экономическая деятельность определяют суть процесса функционирования производственных подразделений, результатом которой должен стать выпуск продукции по заключенным с заказчиками договорам. В этих условиях ставится вопрос о методах планирования и управления производственными подразделениями. Опытное производство характеризуют методы ведения производственного планирования для единичного производства. Для таких предприятий характерен высокий уровень изменчивости производственных заданий в коротких плановых периодах, а также большое количество не определенных к моменту запуска производственных заказов факторов, влияющих на поведение внутренних процессов. Плановые задания определяются на основе сетевых, а также сводных графиков изготовления изделий. Особенностью планирования предприятий, выпускающих единичную продукцию, является групповой запуск деталей разного наименования, обладающих конструктивным сходством.

Важно отметить, что предприятия, занимающиеся выпуском мелкосерийной и единичной продукции на базе опытного производства, не ставят перед собой задачи наращивания производственных мощностей цехов за счет ввода дополнительного оборудования, поскольку оборот выпускаемой продукции не всегда стабилен, а ввод дополнительных мощностей ведет к увеличению расходов собственников предприятий. Стоит вопрос о максимально полном использовании производственного потенциала. Оценивая резервы использования оборудования с учетом действительных возможностей вовлечения его в сферу хозяйственного оборота, можно сказать, что уровень фактической загрузки оборудования может повышаться только до уровня соответствия производственной программы производственному потенциалу. Решение задачи складывается не столько из того, чтобы дополнительно создать недостающие резервы, сколько из правильного перераспределения производственных мощностей для установления требуемого соотношения. Поскольку вариантов распределения производственных мощностей может быть множество в зависимости от состава выпускаемой продукции, то и переменных, влияющих на ход оперативного планирования, также может быть много. Учесть их позволяют системы имитационного компьютерного моделирования.

Специализированные языки моделирования обычно делятся на три группы, соответствующие видам имитации: для непрерывных, дискретных и комбинированных процессов. Одним из примеров языка моделирования дискретных систем может служить язык SIMULA-67, но эффективность его применения в большей степени зависит от разработчика модели – специалиста-математика, нежели от специалиста, имеющего высокую подготовку в сфере производственного планирования. Хорошую поддержку в применении получил язык Pilgrim, разработанный отечественными программистами и включающий в свой набор большое количество операционных объектов. Этот язык хорошо применим для изучения и построения экономических моделей, в частности, для имитации процессов деятельности дискретных производственных систем. Обладая большими инструментальными возможностями и средствами для написания моделей, язык Pilgrim нашел должное применение в области теоретического исследования внутрипроизводственных процессов, но, с другой стороны, изобилие инструментальных

средств становится недостатком для адаптации языка к оперативным производственным процессам.

Еще одним языком имитационного моделирования является Simpas, который, как считают разработчики, превосходит язык GPSS, поскольку обладает более высоким быстродействием и гибкостью. Simpas позволяет использовать возможности Object Pascal, в то время как при разработке моделей на GPSS/PC исследователь ограничен возможностями языка. Существенным недостатком стало то, что Simpas предназначен для работы в операционной системе MS DOS, что приводит к существенным ограничениям не позволяющим осуществить разработку и эксплуатацию моделей сложных систем с высокой степенью детализации.

Языку моделирования GPSS практически удалось избежать всех вышеперечисленных недостатков. GPSS представляет собой универсальный аппарат для изучения комбинированных процессов, реализуемых с помощью набора математических инструментальных средств, позволяющих посредством сопоставления процессов-аналогов выявить оптимальный вариант функционирования изучаемого объекта. Так, сфера применения GPSS может охватывать различные аспекты деятельности при управлении экономическими объектами. GPSS позволяет не только описать посредством специальной имитирующей компьютерной программы поведение изучаемого объекта, но и сделать его более ясным и лаконичным, что выделяет GPSS из ряда других процедурных языков программирования.

Написанные при помощи GPSS модели получаются более адекватными исследуемой системе, чем их аналоги, описанные с использованием теории массового обслуживания. В объектах можно учесть большее количество факторов, тем самым убрать многие первоначальные допущения.

Учитывая рассмотренные выше особенности функционирования производственных систем, занимающихся выпуском единичной продукции, необходимо исходить из представления, что поддержание их работоспособности обеспечивается пропускной способностью производственных ресурсов. Именно система уровня GPSS, моделирующая с высокой точностью прогнозные поведение каждого из компонентов производственных процессов, реализует поддержание целостности системы.

Стоит задача дать оценку эффективности применения системы имитационного моделирования в области изучения производственных процессов предприятий, занимающихся выпуском единичной продукции.

Под пропускной способностью производства понимается динамический параметр мощности, выраженный как количество продукции, необходимой для изготовления в плановом отрезке времени, при сопоставлении со структурой располагаемых ресурсов. При описании производственных процессов посредством языка GPSS заказы, поступающие на станки, можно выразить через распределение входящего потока системы управления очередями. Для выбора параметров системы управления очередями вначале следует определить способ, с помощью которого ожидающие заказы организуются для дальнейшего обслуживания. Инструментальные средства GPSS позволяют в наглядном виде отобразить динамику их распределения в виде блок-диаграммы. На практике различают равномерное и произвольное распределение входящего потока производственных заданий. Равномерное распределение характеризуется строгой периодичностью, т.е. равными интервалами времени между подряд идущими входящими заданиями, что свойственно предприятиям с серийным выпуском продукции, но в нашем случае значительней распространено произвольное распределение входящего потока. Язык имитационного моделирования GPSS способен управлять как равными, так и произвольными потоками.

Преимущества использования языка GPSS как системы имитационного моделирования внутрипроизводственных процессов можно выявить по следующим критериям:

1. имитационная модель позволяет лучше понять реальную систему;
2. выбор метода использования мощностей при составлении производственной программы и последствия этого выбора можно промоделировать путем «сжатия» времени в течение нескольких минут;
3. имитационные модели носят более общий характер, чем математические модели, их можно использовать в тех случаях, когда для проведения стандартного математического анализа нет надлежащих условий;
4. моделирование можно использовать как средство обучения производственного персонала работе с реальной системой;
5. моделирование дает более реалистичное воспроизведение системы, чем математический анализ;
6. моделирование можно использовать для анализа переходных процессов производственных систем, в то время как посредством математических моделей эта задача решается слабо.

Но моделирование производственных систем посредством языка GPSS имеет свои недостатки:

1. нет возможности доказать, что работа описанной модели полностью соответствует работе реальной производственной системы, так как моделирование связано с многочисленными повторениями последовательностей, основанных на генерации случайных чисел, имитирующих наступление событий;
2. для создания имитации модели требуется значительное время построения системы на компьютере.

Эти недостатки можно сгладить лишь тем, что повышение возможностей компьютеров и шаги по унификации различных имитирующих систем значительно расширяют спектр объектов производственной деятельности, которые можно описать посредством имитационных моделей, в частности с применением языка GPSS, делая допущения относительно моделируемой системы практически безграничными.

Литература

1. Боев В.Д. Моделирование систем. Инструментальные средства GPSS World. СПб, 2004.
2. Сороханов А.Д. Моделирование производственных систем. СПб, 2003.
3. Фатхутдинов Р.А. Организация производства. М., 2000.
4. Чейз Р.Б., Эквилайн Н.Дж., Якобс Р.Ф. Производственный и операционный менеджмент. М., 2003.
5. Шрайбер Т. Дж. Моделирование по GPSS. М., 1980.

СРЕДА ИССЛЕДОВАНИЯ МЕТОДОВ ГЕНЕРАЦИИ ВХОДНЫХ ДАННЫХ ДЛЯ СТРУКТУРНОГО ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Ю.А. Яковлев

Научный руководитель – к.т.н., доцент Т.А. Павловская

Рассматриваются компоненты системы для экспериментального исследования методов генерации входных тестовых данных и взаимодействие этих компонентов. Описывается информация, которая может быть получена системой. Анализируются возможности использования этой информации для сравнения, модификации и усовершенствования методов.

Введение

Тестирование – процесс выполнения программы с целью обнаружения ошибок. Под ошибкой понимается несоответствие между действительным результатом работы программы и ожидаемым (правильным) результатом. Входные данные для выполнения и ожидаемые результаты образуют тестовый вариант (тест). В то время как исчерпывающее тестирование, предполагающее полный перебор всех возможных комбинаций исходных данных, невозможно на практике (из-за ограничений времени и ресурсов), целью проектирования тестов является выбор тестовых вариантов с высокой вероятностью обнаружения еще не найденной ошибки [2]. Структура процесса тестирования представлена на рис. 1. При тестировании по принципу «белого ящика» (структурном тестировании) тесты проектируются на основе анализа внутренней структуры программы (Вариант А на рис. 1). При использовании принципа «черного ящика» (функциональном тестировании) тесты проектируются, исходя из описания (спецификации) функций программы (Вариант Б на рис. 1). При обеих методологиях ожидаемые результаты могут быть получены из спецификации, на основе работы прототипов, моделей, аналогов или старых версий программы, а также вручную [1].

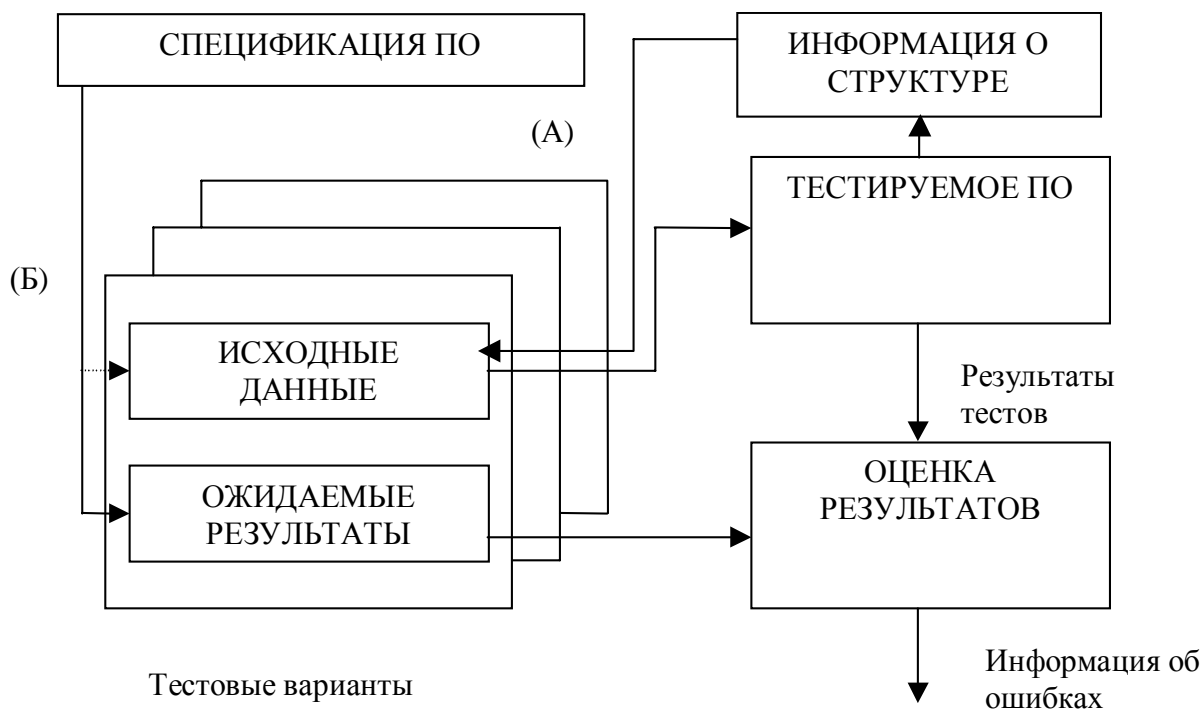


Рис. 1. Схема процесса тестирования

Тестирование – очень трудоемкий и дорогостоящий процесс, автоматизация которого весьма актуальна. В то время как автоматизированное контролируемое выполнение программы, оценка и запись результатов поддерживаются существующими коммерческими системами, задача автоматизированного получения входных тестовых данных зачастую остается нерешенной [6].

В рамках структурного тестирования задача генерации тестовых данных состоит в нахождении такого множества наборов входных переменных, чтобы при выполнении на этих наборах были покрыты все элементы структуры, заданные в соответствии с выбранным критерием тестирования. В частности, эта задача может быть сведена к нахождению входных данных, на которых будет пройден заданный путь или выполнен заданный оператор.

Система автоматизированной генерации тестовых данных обычно представляется состоящей из 3 компонентов [3]: модуля анализа программы, модуля выбора путей и собственно генератора тестовых данных (рис. 2).

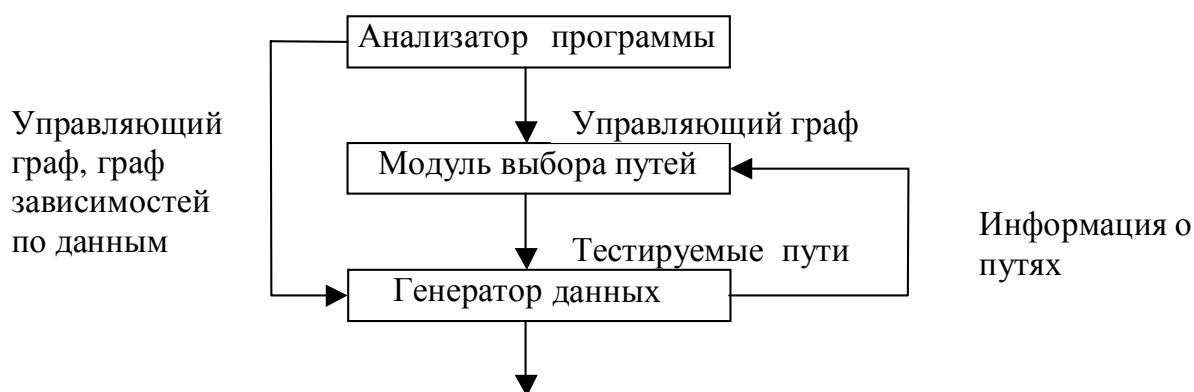


Рис. 2. Высокоуровневое представление системы автоматизированной генерации тестовых данных

В данной работе предлагается расширить подобную систему с целью исследования свойств алгоритмов генерации тестовых данных. Генерируется информация, характеризующая работу метода (алгоритма) для конкретных программ, а также реализуется предоставление исходной информации, необходимой для работы всех рассматриваемых алгоритмов генерации.

Теоретическая часть

Подходы к генерации тестовых данных для структурного тестирования подразделяются на случайные, путь-ориентированные и цель-ориентированные. В зависимости от реализации среди методов генерации выделяют статические, динамические и гибридные методы [4].

Динамические методы основываются на алгоритмах минимизации функций, таких как, например, градиентный метод поиска локального минимума, метод симуляции отжига или генетические алгоритмы поиска глобального минимума. Динамические методы ориентированы на реальное выполнение тестируемой программы. В результате выполнения на конкретных входных данных вычисляется значение функции, показывающее, насколько близок (далек) текущий путь от заданного пути.

Рассмотрим динамический путь-ориентированный алгоритм, предложенный Б. Корелом [5] и использующий градиентный метод поиска. Задается желаемый путь выполнения в управляющем графе (УГ) программы. С каждым из узлов ветвления на этом пути связывается дуговая функция, формируемая на основе соответствующего предид-

ката ветвления. Значение этой функции отрицательно, если выполнение проходит по требуемой ветви, и положительно в противном случае. Первоначально программа выполняется на произвольных входных данных. Если в каком-либо из узлов ветвления выполнение проходит по ветви, отличной от требуемой, то запускается процедура локального поиска. Цель этого поиска – последовательно варьируя исходные переменные, найти такое их значение, при котором текущая дуговая функция примет отрицательное значение, а участок пути, предшествующий текущей вершине, по-прежнему будет выполняться. В первую очередь варьируются те переменные, которые влияют на переменные текущей дуговой функции и менее всего влияют на переменные дуговых функций уже пройденных узлов.

В отличие от градиентного метода, обеспечивающего нахождение локального минимума целевой функции, генетические алгоритмы позволяют найти её глобальный минимум. Если в градиентном методе рассматривается один набор входных данных, переменные которого последовательно меняют свои значения, увеличиваясь или уменьшаясь на определенные величины приращений, то генетический алгоритм поддерживает целое множество наборов (популяцию), и над выбранными элементами этого множества выполняются операции кроссовера (скрещивания) и мутации, по аналогии с эволюционными процессами. Генетический алгоритм, как и алгоритм градиентного поиска, направляется значением целевой функции, которая характеризует каждый набор данных.

Реализация



Рис. 3. Структура среды исследования методов генерации тестовых данных

Рассмотрим функционирование среды для динамических методов генерации тестовых данных.

На основании конфигурационных данных выбирается критерий тестирования. Исходя из заданного критерия, с использованием УГ программы модуль выбора путей формирует набор путей для удовлетворения этого критерия. Для каждой итерации работы алгоритма генерации тестовых данных модулем выбора путей модулю реализации алгоритма передается конкретный требуемый путь. Например, он может быть задан списком вершин УГ или вектором булевских переменных, значения которых кодируют требуемые дуги узлов ветвления на пути.

Анализатор программы передает реализации алгоритма сведения о входных переменных программы и их типах. При этом анализатор выделяет конкретную подпрограмму, подлежащую тестированию (или использует всю программу), контролирует, удовлетворяет ли исходный текст программы ограничениям, присущим методу, выделяет входные переменные из множества всех переменных.

Инструментатор исходного текста, в зависимости от используемого метода, преобразует текст программы, формируя и вставляя дополнительные операторы, записывающие данные о выполнении программы. В частности, для рассматриваемых методов должны быть записаны по отношению к каждому набору входных переменных:

- пройденный путь,
- значения дуговых функций для каждого узла ветвления на пути исполнения.

Реализация метода выполняет присвоение входным переменным значений в соответствии с алгоритмом и вызывает среду выполнения инструментированной программы, передавая сформированные значения. Анализатор зависимостей по данным, в случае использования метода локального поиска, позволяет выбрать те переменные, которые варьируются в первую очередь.

Во время работы алгоритм позволяет собрать информацию, необходимую для анализа его эффективности и путей его совершенствования. Для рассматриваемых алгоритмов к такой информации относятся:

- количество выполнений программы, потребовавшееся для нахождения искомых входных данных или установления факта, что алгоритм не может найти входные данные для заданного пути;
- количество выполнений программы для каждого узла ветвления;
- количество нарушений предшествующего пути.

Помимо этого, для алгоритма Б. Корела имеет смысл протоколировать количество переменных, которые потребовалось перебрать, чтобы добиться отрицательного значения дуговой функции для каждого узла ветвления, а для генетического алгоритма – количество операций кроссовера и мутации, которые были произведены до успешного завершения. При этом вместе с полученными экспериментальными характеристиками следует рассматривать параметры конфигурации и свойства программы – для выявления причинно-следственных связей и возможности настройки алгоритма для конкретных классов программ. К ним можно отнести:

- сложность управляющего графа, число циклов и ветвлений;
- типы данных и количество входных переменных;
- степень присутствия в предикатах ветвления входных переменных.

Для метода Б. Корела к таким данным дополнительно можно отнести параметры варьирования входных переменных (величину приращения). Для генетического алгоритма к конфигурационным параметрам относятся также начальный размер популяции и вероятностные характеристики операций кроссовера и мутации.

Заключение

Рассмотрена структура и функции среды для исследования динамических методов генерации входных данных для структурного тестирования. Данная среда предоставляет возможность для сравнения, модификации и усовершенствования методов.

Литература

1. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. СПб: Питер, 2004.
2. Орлов С. Технологии разработки программного обеспечения. Учебное пособие. СПб.: Питер, 2003.
3. Edwardsson J. A survey on Automatic Test Data Generation. 2001. http://www.ida.liu.se/~joned/papers/class_atdg.pdf
4. Han, S.-H. Survey on Automatic Test Data Generation. 2002. http://salmosa.kaist.ac.kr/LAB/RESEARCH/SEMINAR/GENERAL/year2002/material/20020415_1.ppt .
5. Korel B. Automated Software Test Data Generation. //IEEE Transactions on Software Engineering. 1990. Vol 16. No. 8.
6. Ould M.A. Testing – a challenge to method and tool developers. //Software Engineering Journal. 1991. №3.

СОЗДАНИЕ СИСТЕМЫ АВТОМАТИЧЕСКОГО ЗАВЕРШЕНИЯ ВВОДА С ИСПОЛЬЗОВАНИЕМ ПАКЕТА *UNIMOD*

В.С. Гуров, М.А. Мазин

Научный руководитель – д.т.н., проф. А.А. Шалыто

В работе описана технология разработки системы автоматического завершения ввода, предназначенная для удобного редактирования программных текстов. Используя грамматику, задающую язык, предлагается строить конечный автомат типа Мили. Для разработки и отладки конечных автоматов используется программный пакет с открытым кодом *UniMod* (<http://unimod.sf.net>).

Введение

В работе [1] предложен метод проектирования событийных объектно-ориентированных программ с явным выделением состояний, названный «*SWITCH*-технологией». Особенность этого подхода состоит в том, что поведение в таких программах описывается с помощью графов переходов структурных конечных автоматов с нотацией, предложенной в работе [2]. *SWITCH*-технология для описания каждого автомата определяет два типа диаграмм (схемы связей и графы переходов). При наличии нескольких автоматов, кроме того, строится схема их взаимодействия. *SWITCH*-технология задает нотацию и операционную семантику используемых диаграмм.

Программный пакет с открытым кодом *UniMod* (<http://unimod.sf.net>), созданный авторами статьи, обеспечивает разработку и выполнение автоматически-ориентированных программ. Пакет, сохраняя автоматный подход, позволяет использовать *UML*-нотацию при построении диаграмм в рамках *SWITCH*-технологии. При этом схемы связей, определяющие интерфейс автоматов, строятся в нотации диаграмм классов языка *UML*, а графы переходов — в *UML*-нотации диаграмм состояний. В состав пакета *UniMod* входит встраиваемый модуль (*plug-in*) для платформы *Eclipse* (<http://www.eclipse.org>), позволяющий создавать и редактировать *UML*-диаграммы классов и состояний, которые соответствуют схеме связей и графу переходов.

На сегодняшний день интегрированные системы для разработки программ предоставляют удобные средства для работы с кодом, такие как, например:

- подсветка семантических и синтаксических ошибок;
- автоматическое завершение ввода и автоматическое исправление ошибок;
- форматирование и рефакторинг [3] кода;
- запуск и отладка программы внутри среды разработки.

В английском языке эти средства получили название "*code assist*".

В рамках создания очередной версии пакета *UniMod* перед авторами встала задача реализации системы автоматического завершения ввода при редактировании условий на переходах на *UML*-диаграмме состояний. В статье описана технология создания такой системы и ее автоматически-ориентированная реализация, выполненная с помощью пакета *UniMod*.

Постановка задачи

Автоматическим завершением ввода, применительно к редактированию программных текстов, традиционно называют технологию, позволяющую пользователю получить список строк, при добавлении которых в текст после позиции курсора программа будет синтаксически верна. Например, на рис. 1 показано, как среда разработки *Eclipse* предлагает варианты автоматического завершения ввода для текущей позиции курсора.

Сформулируем требования к проектируемой системе автоматического завершения ввода. Пусть задан язык L и на вход системы подана строка α .

1. Если поданная на вход строка α является префиксом предложения языка L ($\exists \omega : \alpha\omega \in L$), то система должна возвращать множество строк $C(\alpha) = \{\beta_i\}_{i=1..n}$, любая из которых может являться продолжением данной α ($\forall i \in [1..n] \exists \gamma : \alpha\beta_i\gamma \in L$);
2. Если поданная на вход строка α не является префиксом предложения на заданном языке ($\neg \exists \omega : \alpha\omega \in L$), то система должна с помощью дополнения строки недостающими символами или с помощью удаления лишних символов трансформировать строку в правильный префикс предложения языка. Количество дополнений и удалений должно быть как можно меньше.

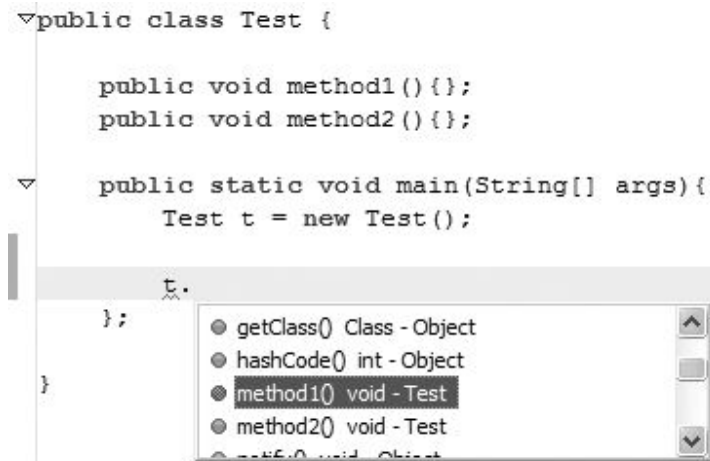


Рис. 1. Пример автоматического завершения ввода

Элементы теории построения компиляторов

Если исходный язык L задан формальной порождающей грамматикой, то очевидно, что для построения такой системы необходимо использовать методы проектирования компиляторов [5]. Существует множество инструментов для автоматического создания компиляторов по заданной грамматике. Достаточно большой список приведен, например, на сайте <http://www.kulichki.net/kit/tools/java.html>.

На рис. 2 приведена обобщенная структура компилятора. Лексический анализатор осуществляет чтение входной цепочки символов и их группировку в элементарные конструкции, называемые лексемами. Синтаксический анализатор осуществляет разбор исходной программы, используя поступающие лексемы. семантический анализ программы и построение промежуточного представления программы. Генератор кода преобразует промежуточное представление программы в объектный код. Генератор кода может быть заменен интерпретатором, при этом вместо генерации объектного кода будет выполняться интерпретация промежуточного представления программы.

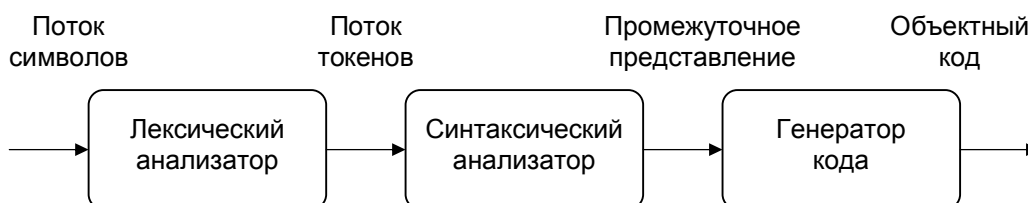


Рис. 2. Обобщенная структура компилятора

В проекте *UniMod* трансляция выражений на переходах выполняется с помощью, так называемого, «компилятора компиляторов» *ANTLR* [4]. Он по заданной $LL(k)$ грам-

матике строит код на языке *Java*, реализующий лексический анализатор и рекурсивный нисходящий синтаксический анализатор. Построенный синтаксический анализатор может быть использован и как распознаватель принадлежности выражения заданному грамматикой языку, и как транслятор выражений в абстрактное синтаксическое дерево. Данный анализатор не может быть использован для построения системы автоматического завершения ввода, так как в случае подачи ему на вход префикса для выражения на заданном языке вместо законченного выражения, он выдает ошибку.

Одним из возможных вариантов реализации требуемой системы может быть использование нерекурсивного нисходящего синтаксического анализатора явно использующего стек и управляемого таблицей разбора. Таблица разбора представляет собой двумерный массив $M[A, a]$, где A – нетерминал, а a – терминал (лексема) или символ конца потока $\$$. В ячейках таблицы записываются продукции грамматики, с помощью которых заменяются нетерминалы на вершине стека, пустые ячейки таблицы означают ошибки. Подробно работа такого анализатора описана в работе [5].

При подаче на вход описанному выше анализатору незавершенной строки α без символа конца потока, анализатор остановится, имея какой-то нетерминал на вершине стека. В этом случае множество терминалов $C(\alpha)$, ожидаемых вслед за обработанной строкой, может быть определено как $\{\beta : M[T, \beta] \neq \emptyset\}$, где T - нетерминал на вершине стека после остановки анализатора.

Для реализации восстановления после ошибок в «режиме паники», таблица разбора может быть дополнена синхронизирующими символами, которые вписываются в некоторые пустые ячейки. При получении неожиданного терминала, анализатор пропускает символы входного потока до тех пор, пока не будет обнаружен терминал, соответствующий синхронизирующему символу. Для восстановления на уровне фразы в некоторые пустые ячейки вписываются указатели на подпрограммы обработки ошибок, которые могут изменять, вставлять или удалять терминалы входного потока или элементы стека.

Описание предлагаемой технологии

В [7] показано как создать программу нерекурсивного нисходящего синтаксического анализатора, используя автоматически-ориентированный подход, при этом таблица разбора в работе [7] оставлена и выступает в роли объекта управления.

В настоящей работе предлагается технология создания системы автоматического завершения ввода, позволяющая исключить таблицу разбора нисходящего нерекурсивного синтаксического анализатора и использующая гибкий алгоритм восстановления после ошибок на уровне фразы.

Технология основывается на том, чтобы для заданной $LL(1)$ грамматики построить конечный автомат типа Мили, который будет являться синтаксическим анализатором. Автомат должен реагировать на события, которые поставляет ему лексический анализатор. Каждому событию соответствует терминал. В работе [5, 12] приведено описание подхода для создания нисходящего синтаксического анализатора на основе диаграмм переходов. При этом предлагается записывать по одной диаграмме для каждого правила вывода грамматики. Описываемая в настоящей статье технология предлагает сворачивать все диаграммы в одну, при необходимости удаляя рекурсию с помощью метода, описанного в работе [6]. Такой подход позволяет избавиться от упоминания нетерминалов на диаграммах переходов и, следовательно, разорвать семантическую связь с исходной грамматикой. Такой разрыв позволит описывать язык только с помощью диаграммы переходов и автоматически получать реализацию распознавателя для данного языка.

При подаче на вход системе, построенной описанным выше образом, незавершенной строки, автомат, реализующий синтаксический анализатор, останавливается в каком-то состоянии. События, заданные на переходах из состояния, в котором остановился автомат, определяют множество терминалов, которые могут следовать за последним терминалом, извлеченным из входной строки. После построения такого множества терминалов, каждый терминал обратно преобразуется в строку символов.

Построение диаграммы переходов синтаксического анализатора

Пусть $LL(1)$ грамматика для нашего примера задана следующим множеством правил вывода:

- 1) $S \rightarrow \text{else} \mid T S'$
- 2) $S' \rightarrow \text{or} T S' \mid \varepsilon$
- 3) $T \rightarrow L T'$
- 4) $T' \rightarrow \text{and} L T' \mid \varepsilon$
- 5) $L \rightarrow \text{not} L \mid P$
- 6) $P \rightarrow '(S)' \mid \text{int rel N} \mid \text{bool} \mid N P'$
- 7) $P' \rightarrow \text{rel int} \mid \varepsilon$
- 8) $N \rightarrow \text{id dot id}$

Терминал **id** соответствует идентификатору, терминал **int** — целочисленной константе, терминал **bool** — булевой константе, а терминал **rel** — бинарному отношению ('>', '<', '>=', '<=', '=', '≠'). Опишем формальный процесс построения автомата типа Мили для данной грамматики.

На рис. 3 приведены диаграммы переходов для каждого нетерминала заданной грамматики, построенные с помощью метода, описанного в работе [5]. Единственным отличием указанных диаграмм, от диаграмм предлагаемых указанной работе, является наличие выделенных начального и конечного состояний, отображаемых закрашенным кругом и закрашенным кругом внутри окружности соответственно. Из начального состояния существует всегда только один переход, в конечное состояние также ведет только один переход.

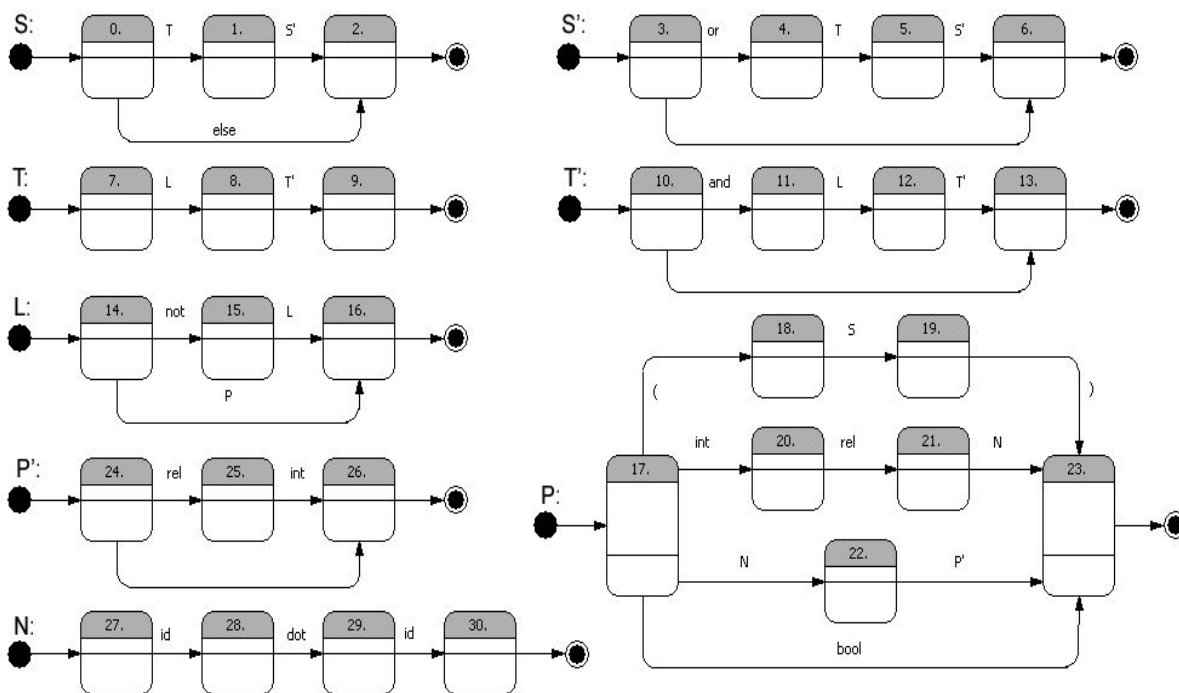


Рис. 3. Диаграммы переходов для каждого нетерминала грамматики

Состояния соответствуют позициям [8] в правилах вывода, метки на переходах — терминалам и нетерминалам, отделяющим позиции друг от друга. Если нетерминал выводит ε -правило, то из состояния, соответствующего начальной позиции, существует непомеченный переход в состояние, соответствующее конечной позиции. Непомеченные переходы также называются немотивированными.

Далее, множество диаграмм, представленных на рис. 3, можно преобразовать в одну диаграмму состояний, на которой все переходы будут помечены только терминалами. Процесс такого преобразования предполагает выполнение следующих шагов:

1. удаление правой рекурсии;
2. удаление немотивированных переходов;
3. подстановка диаграмм переходов друг в друга;
4. удаление срединной рекурсии;

Опишем каждый шаг подробно.

Удаление правой рекурсии

Наличие праворекурсивного правила вывода, означает, что на диаграмме, соответствующей некоторому терминалу N , есть переход, помеченный тем же нетерминалом N , ведущий в состояние, соответствующее конечной позиции.

Например, наличие праворекурсивного правила (2) влечет наличие перехода из состояния 5 в состояние 6 на рис. 3. Для устранения правой рекурсии этот переход должен быть заменен немотивированным переходом в состояние, соответствующее начальной позиции – в состояние 3 (рис. 4).

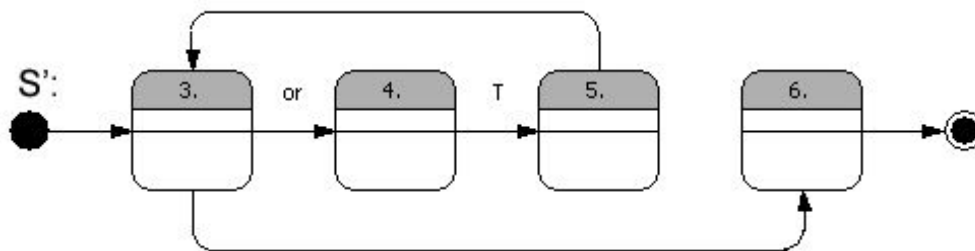


Рис. 4. Удаление правой рекурсии на диаграмме состояний для нетерминала S'

Удаление немотивированных переходов

Наличие немотивированного перехода из состояния S_1 в состояние S_2 , означает, что за позицией, соответствующей состоянию S_1 , могут следовать те же терминалы и нетерминалы, что и за позицией, соответствующей состоянию S_2 .

Для устранения немотивированного перехода выполняются следующие операции:

1. создать сложное состояние $S_{1,2}$ [9];
2. поместить состояния S_1 и S_2 внутрь состояния $S_{1,2}$;
3. все переходы из состояния S_2 заменить аналогичными переходами из состояния $S_{1,2}$.

На рис. 5 показано удаление немотивированного перехода из состояния 5 в состояние 3, присутствующего на рис. 4.

Будем говорить, что исходящий переход из состояния S_1 совпадает с исходящим переходом из состояния S_2 , если он помечен тем же символом и ведет в тоже состояние.

После выделения группового состояния $S_{1,2}$ множества переходов, исходящих из состояний S_1 и S_2 могут совпасть. В этом случае из каждой пары совпадающих переходов следует оставить только один и его начало переместить в состояние $S_{1,2}$. Все переходы, входящие в состояния S_1 и S_2 , перенаправить в состояние $S_{1,2}$. Состояния S_1 и S_2 ликвидировать.

Отметим, что данный алгоритм аналогичен вычеркиванию одинаковых записей из таблицы, задающей функцию переходов автомата [10].

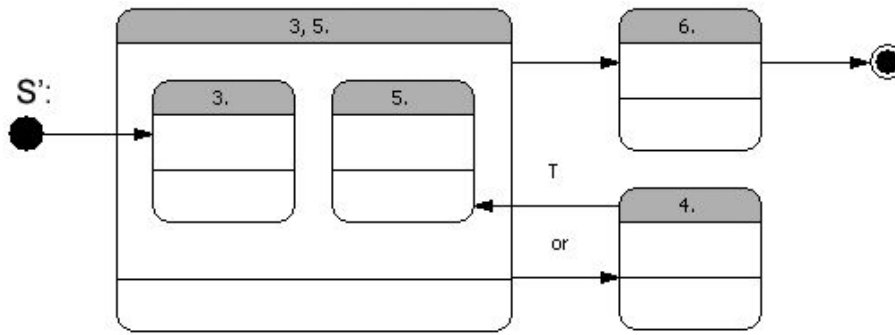


Рис. 5. Удаление немотивированного перехода из состояния 5 в состояние 3

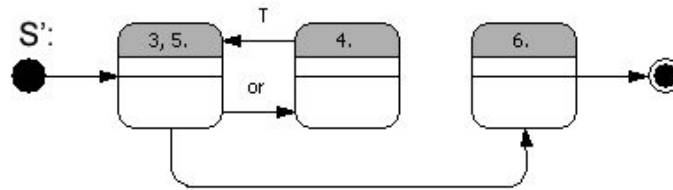


Рис. 6. Упрощение диаграммы состояний, показанной на рис. 5

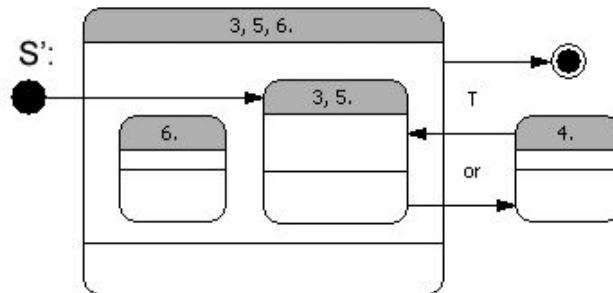


Рис. 7. Удаление немотивированного перехода из состояния 3,5 в состояние 6

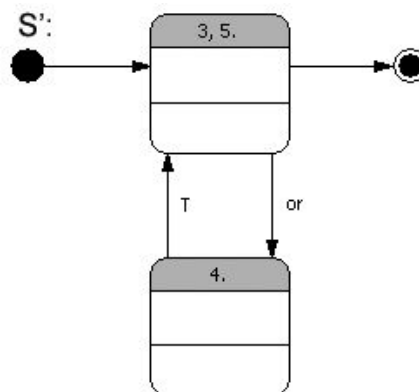


Рис. 8. Преобразованная диаграмма состояний для нетерминала S'

На рис. 5 описанный алгоритм можно применить для состояний 3 и 5. Результирующая диаграмма показана на рис. 6. На рис. 7 показано удаление немотивированного перехода из состояния 3,5 в состояние 6. Состояние 6 на рис. 7 не имеет входящих переходов и, следовательно, не достижимо, поэтому оно может быть удалено. После удаления состояния 6 в состоянии 3,5,6 будет вложено единственное состояние 3,5. Начала переходов, исходящих из состояния 3,5,6 следует перенести в состояние 3,5, а само состояние 3,5,6 удалить. На рис. 8 приведена диаграмма состояний для нетерминала S' после всех описанных модификаций.

Подстановка диаграмм переходов друг в друга

Диаграммы переходов могут быть упрощены подстановкой одних диаграмм в другие. Предположим, что на диаграмме для нетерминала N_1 существует переход из состояния S_1 в состояние S_2 , помеченный нетерминалом N_2 . Заменяем такой переход на немотивированный из состояния S_1 в состояние, следующее за начальным на диаграмме переходов для нетерминала N_2 . Добавим переход из состояния, предшествующего конечному, на диаграмме переходов для нетерминала N_2 в состояние S_2 . Отметим, что указанную подстановку необходимо выполнять, только если $N_1 \neq N_2$, так как в противном случае имеет место срединная рекурсия, удаление которой будет описано ниже.

После выполнения такой подстановки, возникшие немотивированные переходы следует устранить описанным ранее способом. При этом сначала устраняется немотивированный переход, из состояния S_1 , а затем переход в состояние S_2 .

На рис. 9 продемонстрирована подстановка диаграммы переходов для нетерминала L в диаграмму переходов для нетерминала T . На рис. 10 приведена диаграмма переходов после устранения немотивированных переходов.

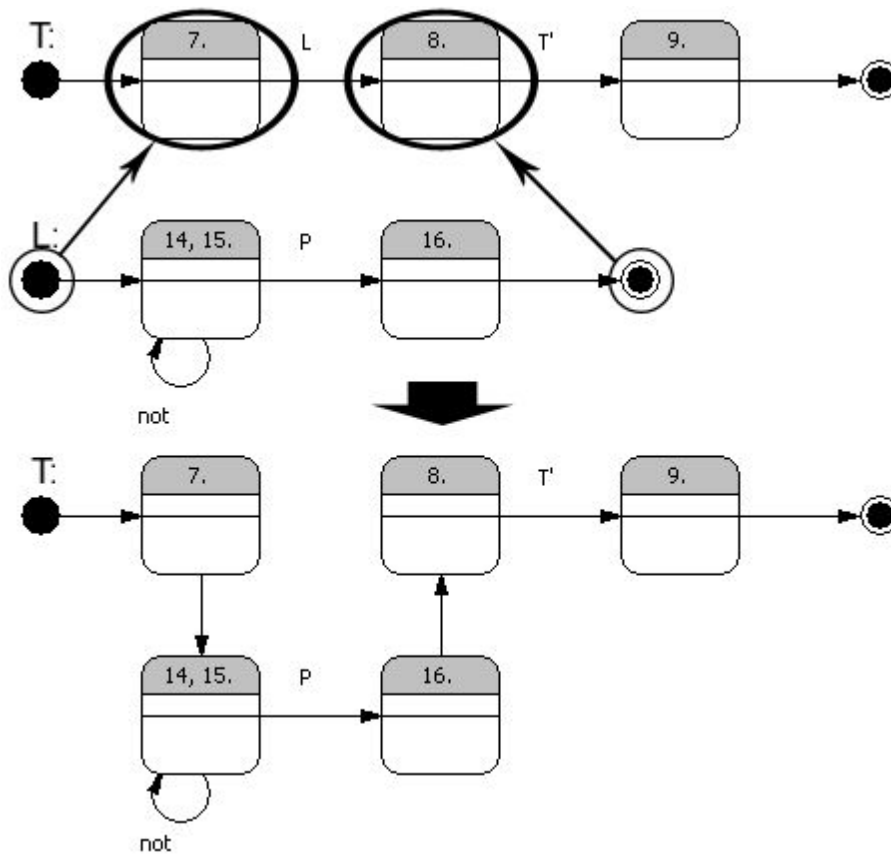


Рис. 9. Подстановка диаграмм переходов друг в друга

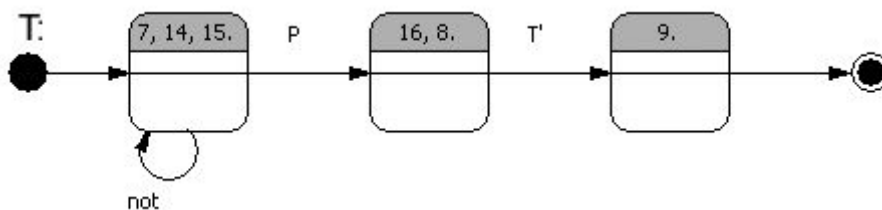


Рис. 10. Устранение немотивированных переходов после подстановки

Если некоторый нетерминал N_1 неоднократно присутствует на диаграмме для нетерминала N_2 , то каждый переход, помеченный N_1 , необходимо заменить соответст-

вующей диаграммой переходов. В результате количество однотипных подграфов на диаграмме N_2 чрезмерно возрастает.

Предлагается преобразовать диаграмму N_2 таким образом, чтобы нетерминал N_1 встречался на ней минимальное число раз. Для этого используется следующий метод: если несколько переходов помеченных нетерминалом N_1 ведут в одно и то же состояние S , то можно выделить составное состояние, и заменить все эти переходы единственным переходом, помеченным нетерминалом N_1 , исходящим из группового состояния и входящим в состояние S .

На рис. 11 данный метод применен для диаграммы переходов нетерминала S .

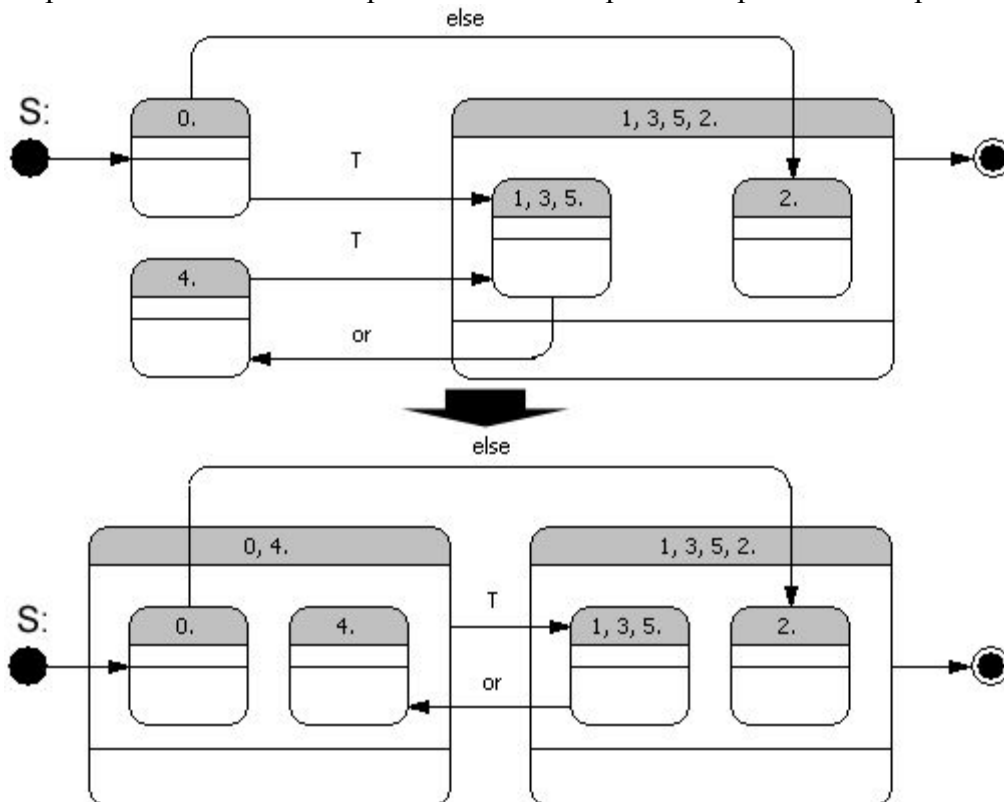


Рис. 11. Удаление двух переходов помеченных нетерминалом T на диаграмме переходов для нетерминала S

Удаление срединной рекурсии

В результате выполнения предыдущих шагов исходное множество диаграмм переходов преобразуется во множество диаграмм, возможно, имеющих срединную рекурсию. Для описанной выше грамматики исходное множество диаграмм, показанное на рис. 3, преобразуется в одну диаграмму, приведенную на рис. 12. На этой диаграмме присутствует срединная рекурсия – переход из состояния 18 в состояние 19 .

Для исключения переходов, образующих срединную рекурсию, предлагается использовать метод, предложенный в работе [6]:

- пусть на диаграмме переходов для нетерминала N существует рекурсивный переход из состояний S_1 в состояние S_2 , помеченный нетерминалом N ;
- заменим такой переход двумя немотивированными переходами. При этом первый из них должен выходить из состояния S_1 , а входить в состояние, следующее за начальным состоянием на диаграмме. Второй переход должен выходить из состояния, предшествующего конечному, а входить в состояние S_2 ;
- на первом переходе должно выполняться действие по добавлению в стек метки M_{S_2} , соответствующей исходному целевому состоянию S_2 , а на втором переходе — дей-

ствии по извлечению метки M_{S2} из стека, при условии, что метка M_{S2} находится на вершине стека.

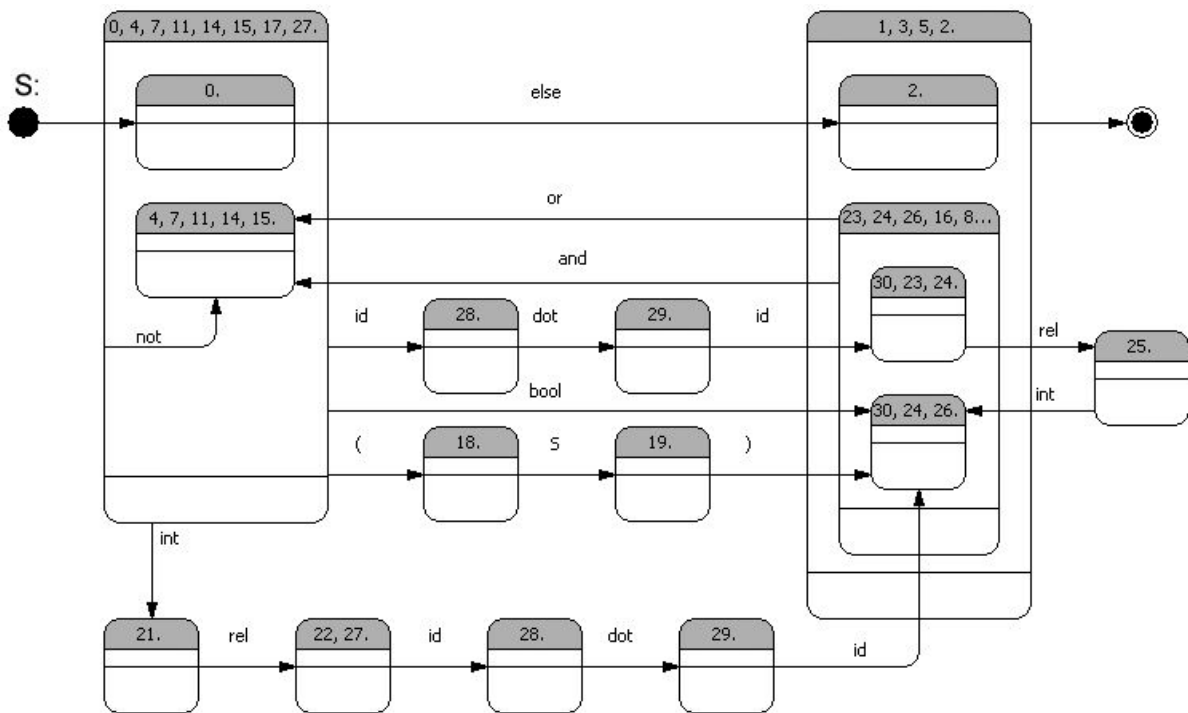


Рис. 12. Диаграмма состояний со срединной рекурсией

На рис. 13 показана диаграмма, приведенная на рис. 12, с удаленной срединной рекурсией.

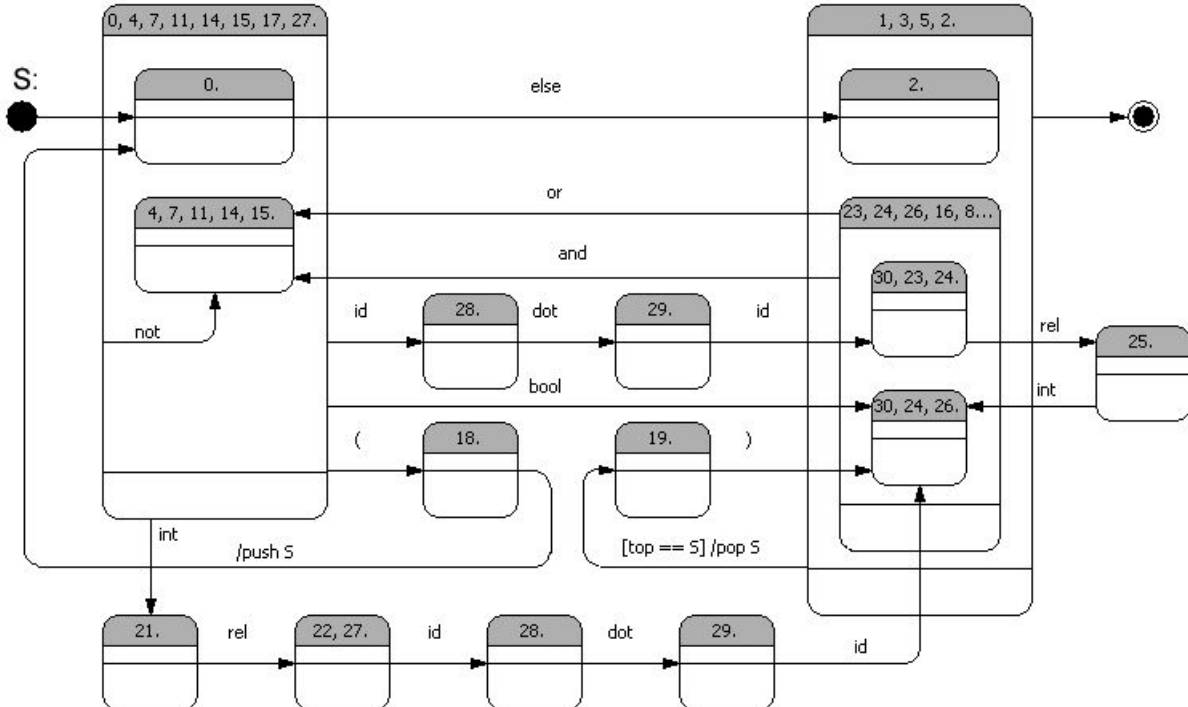


Рис. 13. Диаграмма состояний с удаленной срединной рекурсией

Отметим, что если преобразованное множество диаграмм состоит более чем из одной диаграммы, то после удаления срединной рекурсии в каждой из них, необходимо продолжить процесс преобразования с шага 3 – подстановка диаграмм переходов друг в друга.

Модель разрабатываемой системы

Описанные выше шаги приводят к построению диаграммы переходов для автомата типа Мили. Поставим в соответствие каждому терминалу событие, поставляемое лексическим анализатором, и создадим схему связей автомата [1] в виде *UML*-диаграммы классов [9].

На рис. 14 приведена такая схема. При этом на ней слева показан объект *p1*, соответствующий лексическому анализатору, в центре – объект *A*, соответствующий автомату Мили, а справа – объект управления *o1*, соответствующий стеку.

На рис. 15 показана *UML*-диаграмма состояний автомата, построенная на основе диаграммы, приведенной на рис. 13, с помощью замены терминалов событиями и замены действий на переходах ссылками на методы объекта управления *o1*. Состояния 18 и 19 на рис. 15 отсутствуют из-за удаления немотивированных переходов.

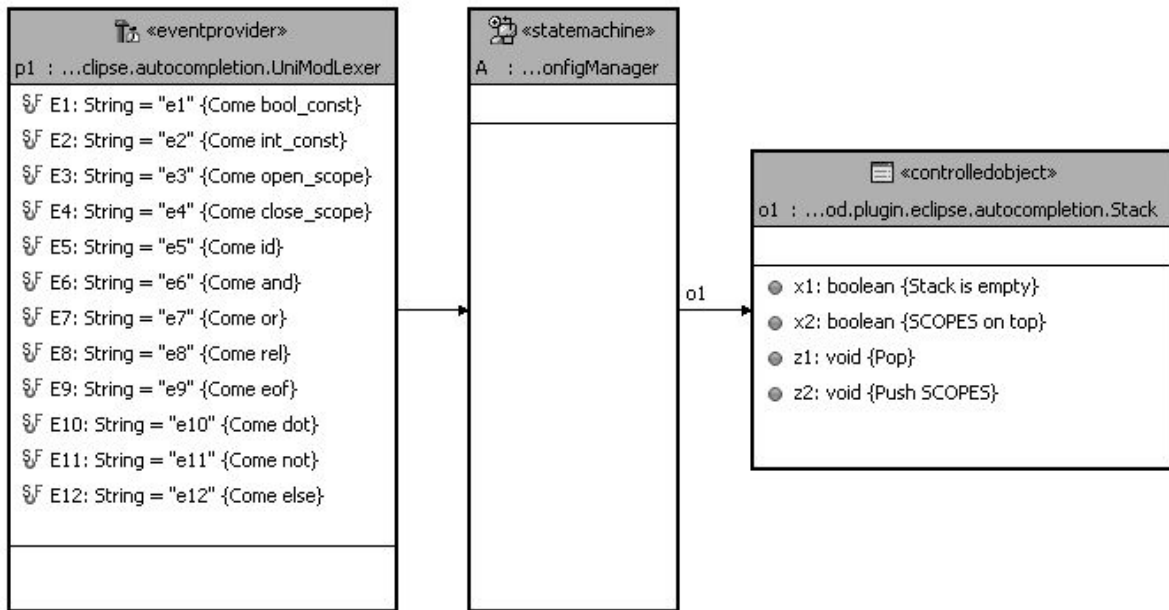


Рис.. 14. Схема связей автомата

Полученная модель системы состоит из двух *UML*-диаграмм (рис. 14, 15) и описывает распознаватель для языка, заданного приведенной выше грамматикой. Отметим, что информация о приоритете операций была потеряна в ходе преобразований, и, поэтому, модель может быть использована для распознавания принадлежности выражений языку, но не для трансляции выражений.

Выражение, принадлежащее языку, поданное на вход распознавателю, приводит автомат *A* в финальное состояние. При подаче на вход выражения являющегося префиксом какого-либо выражения принадлежащего языку, автомат остановиться в каком-то состоянии, множество исходящих переходов из которого определяет множество возможных следующих терминалов.

Если выражение языку не принадлежит и не является префиксом какого-либо выражения принадлежащего языку, то автомат *A* остановится в состоянии, в котором было получено событие, для которого не существовало исходящих переходов при текущих значениях входных переменных. В этом случае множество возможных следующих терминалов можно определить только для последнего правильно обработанного терминала.

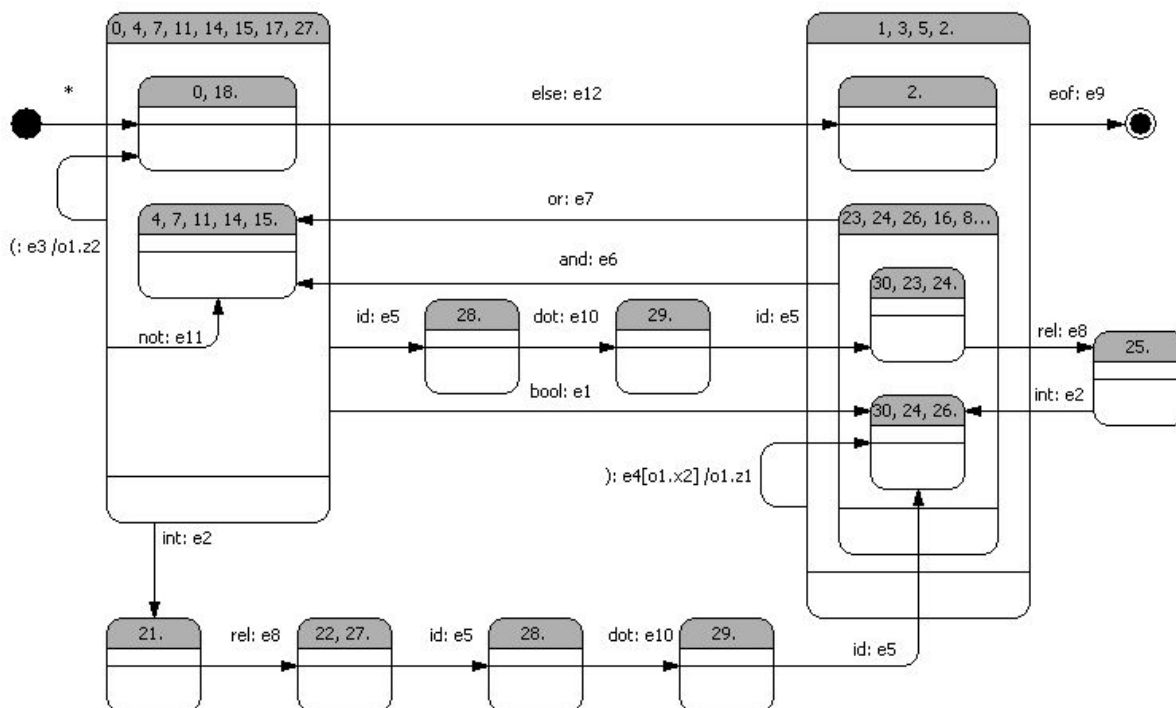


Рис. 15. Диаграмма состояний автомата

Восстановление после ошибок

Перейдем к реализации второго, предъявляемого к системе, требования – обработка ошибочных строк. Для его реализации автомат A необходимо модифицировать таким образом, чтобы он корректно восстанавливался в случае подачи на вход выражения, не являющегося префиксом какого-либо выражения принадлежащего языку. При этом автомат должен всегда останавливаться в состоянии, в которое существует переход по событию, соответствующему последнему терминалу, извлеченному из поданного на вход выражения.

Существует несколько возможных вариантов реализации восстановления автомата после ошибки [5]. Например, можно для каждого состояния добавить такой исходящий переход, ведущий в конечное «ошибочное» состояние, что он будет срабатывать в случае отсутствия какого-либо другого исходящего перехода для пришедшего события и текущих значений входных переменных. Это приведет к тому, что при появлении в процессе распознавания первого же ошибочного терминала, автомата завершит работу в «ошибочном» состоянии. Однако из «ошибочного» состояния нет исходящих переходов, и, следовательно, множество возможных последующих терминалов будет пустым.

Авторы предлагают использовать альтернативный алгоритм обработки ошибочных ситуаций, основанный на локально оптимальной коррекции входного потока терминалов от лексического анализатора. Такой подход также называют восстановлением на уровне фразы.

Пусть из состояния S нет исходящего перехода для пришедшего события e , соответствующего некоторому терминалу. Тогда коррекция потока может осуществляться двумя способами: дополнением потока недостающими терминалами; пропуском лишних терминалов в потоке.

Для того, чтобы автомат A , находясь в состоянии S , пропустил в потоке терминал, соответствующий пришедшему событию e , необходимо добавить в автомат петлю в состоянии S по событию e . Тогда, находясь в состоянии S и получив событие e , автомат останется в состоянии S , то есть проигнорирует пришедшее событие, и, как следствие, пропустит терминал в потоке.

Для того чтобы автомат A , находясь в состоянии S , дополнил поток недостающими терминалами, следует выполнить указанные ниже операции:

1. найти достижимое из S состояние S_h , такое, что в нем существует исходящий переход по событию e . Если из состояния S достижимы несколько таких состояний, то выберем ближайшее из них;
2. если из ближайшего найденного состояния S_h есть переход в некоторое состояние S_i по событию e при условии c , то необходимо добавить в автомат переход из состояния S в состояние S_i по событию e при условии c . Отметим, что отсутствие условия трактуется как тождественная истина.

Последовательность терминалов, соответствующих событиям, которыми помечен кратчайший путь из состояния S в состояние S_h , можно использовать для вставки в поток перед терминалом, соответствующим пришедшему событию e .

Если лексический анализатор позволяет заглядывать на произвольное количество терминалов вперед, то можно применять оба способа коррекции одновременно, выбирая оптимальный способ в процессе разбора.

Для выбора оптимального способа авторы предлагают использовать следующее правило:

1. при получении ошибочного терминала в текущем состоянии вычислим количество терминалов, которыми нужно дополнить поток;
2. вычислим количество терминалов, которое нужно пропустить в потоке, до следующего обрабатываемого в текущем состоянии терминала;
3. выполним коррекцию, требуемое количество терминалов для которой минимально.

Для реализации этого способа коррекции к автомату A распознавателя в качестве объекта управления добавим лексический анализатор (рис. 16). Лексический анализатор предоставляет автомату распознавателя целочисленную входную переменную $o2.x1$. Ее значение равно числу терминалов, которые необходимо пропустить в потоке, до следующего терминала, обрабатываемого в текущем состоянии. Если входной поток вообще не содержит терминалов, обрабатываемых в текущем состоянии, то значение переменной $o2.x1$ больше любого наперед заданного целого числа.

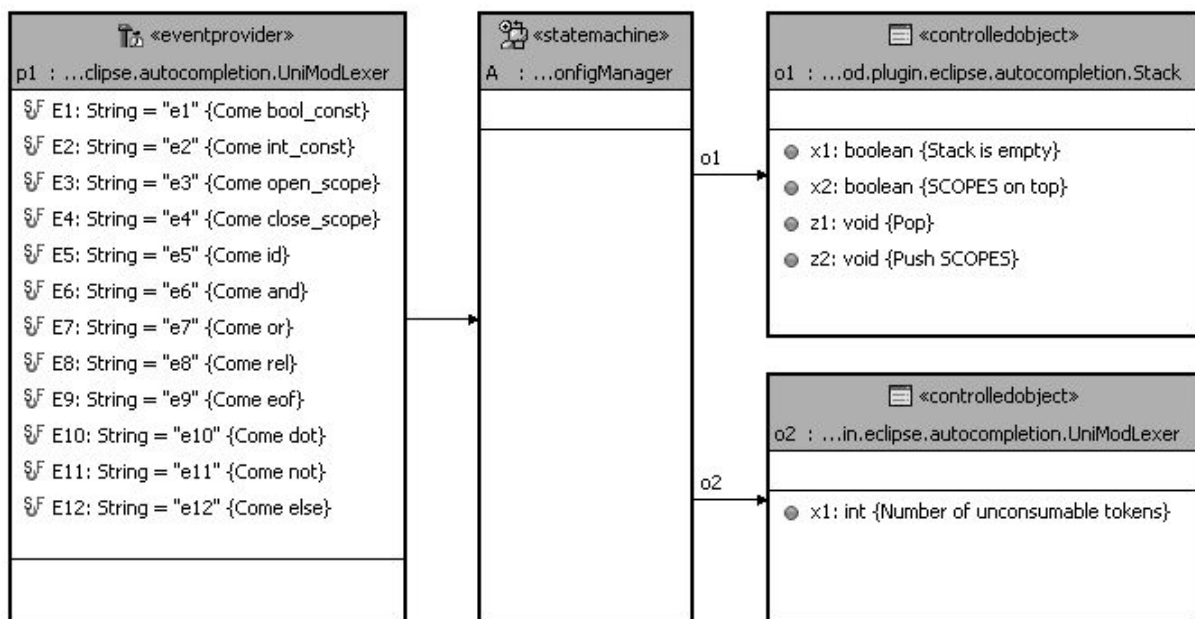


Рис. 16. Схема связей модели распознавателя с лексическим анализатором в качестве объекта управления

В автомат A добавляются переходы, реализующие и добавление и пропуск терминалов в потоке. Переходы, реализующие дополнение потока, помечаются следующим ус-

ловием: длина пути из состояния S в состояние S_h меньше или равна значению входной переменной $o2.x1$. Переходы, реализующие пропуск терминалов, помечаются отрицанием того же условия. Если для состояния S не существует состояния S_h , то переход, удаляющий лексему, выполняется безусловно.

Например, в состоянии 21 нет переходов по событию $e10$ – в этом состоянии появление во входном потоке терминала **dot** (точка) не ожидается. Для того чтобы обработать ошибочное появление этого терминала, необходимо добавить два перехода, исходящих из состояния 21 (рис. 17). Ближайшее состояние, в котором обрабатывается событие $e10$ — состояние 28. Длина пути из состояния 21 в состояние 28 равна двум. Поэтому условие на петле в состоянии 21 по событию $e10$ имеет вид $o1.x1 < 2$. Таким образом, в случае, если сразу за терминалом **dot** в потоке следует терминал **rel** (отношение), то терминал **dot** игнорируется. Если следует какой-нибудь другой терминал, то целесообразно сразу перейти в состояние 29, то есть добавить в поток отсутствующие терминалы **rel** и **id** (идентификатор).

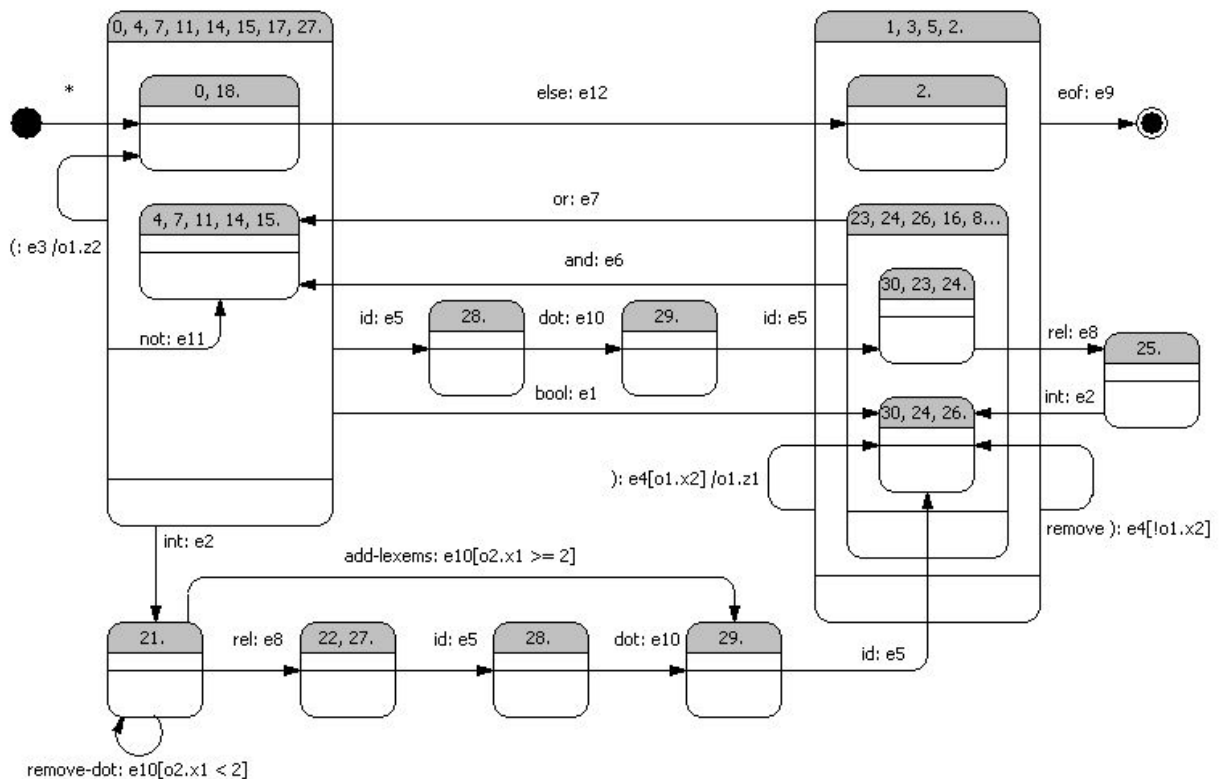


Рис. 17. Добавление переходов, корректирующих поток, в состоянии 21

Описанные выше преобразования могут быть выполнены автоматически для любой диаграммы переходов, так как ближайшее состояние, в котором обрабатывается неожиданный терминал для данного состояния, может быть вычислено, используя, например, алгоритм Флойда-Уоршала [11].

Получение множества строк для автоматического завершения ввода

Предлагаемый алгоритм коррекции входного потока позволяет вычислить множество вариантов завершения как для выражений, являющихся префиксами принадлежащих языку выражений, так и для ошибочных выражений.

После того, как автомат распознавателя, дополненный корректирующими переходами, обработает все терминалы, извлеченные из поданного на вход выражения, он окажется в некотором состоянии S . Для построения множества вариантов завершения следует определить множество переходов, исходящих из S , условия на которых при те-

кущих значениях входных переменных истинны. Терминалы, соответствующие событиям, которыми помечены эти переходы, должны быть преобразованы обратно во множество строк. Например, терминал **id** должен быть преобразован во множество имен переменных, а терминал **and** – в строку «&&». Полученное множество строк и будет множеством вариантов завершения.

Пример работы системы

Приведем пример построения множества вариантов завершения.

Пусть на вход распознавателю подана строка:

```
! o1.x1 &&
```

Лексический анализатор преобразует ее в поток терминалов:

```
not id dot id and,
```

которому соответствует последовательность событий:

```
e11, e5, e10, e5, e6
```

В процессе обработки этих событий автомат изменяет состояния в следующем порядке:

```
(0,18) →(4,7,11,14,15) →(28) →(29) →(30,23,24) →(4,7,11,14,15)
```

Состояние (4,7,11,14,15), в котором остановился автомат, содержит исходящие переходы для событий:

```
e1, e2, e3, e5, e11.
```

Этим событиям соответствуют терминалы:

```
bool, int, '(', id, not,
```

которые преобразуются в строки:

```
"true", "false", "(", "o1", "o2", "o22", "!".
```

Эти строки и формируют множество вариантов завершения для строки, поданной на вход распознавателю.

На рис. 18 показан фрагмент среды разработки с встроенной системой автоматического завершения ввода, описанной в настоящей статье.

Property	Value
Event	e9
Guard	! o1.x1 &&
Name	
Output	<div style="border: 1px solid black; padding: 2px;"> (! o1 o2 o22 true false </div>

Рис. 18. Пример автоматического завершения ввода

Заключение

В работе [5] отмечено, что нерекурсивные нисходящие синтаксические анализаторы можно строить, используя диаграммы переходов, записанные для каждого нетерминала исходной грамматики. Настоящая работа предлагает подход для построения всего одной диаграммы переходов для исходной грамматики. На базе построенной диаграммы реализуется система автоматического завершения ввода. Также отметим, что, в известной авторам литературе, описание формального метода построения подобных систем отсутствует. Данная работа устраняет указанный пробел.

Реализация системы автоматического завершения ввода для следующей версии проекта *UniMod* выполнена с помощью предыдущей версии проекта, вследствие чего

часть проектной документация была получена «автоматически», так как диаграммы созданные с помощью *UniMod* редактора являются автоматными программами и могут быть включены в проектную документацию без изменений. Разработка последующих версий средств разработки с помощью предыдущих является общепринятой практикой и позволяет говорить о зрелости программного продукта.

Литература

1. Шалыто А.А., Туккель Н.И. Танки и автоматы // ВУТЕ/Россия. 2003. № 2, с. 69-73. <http://is.ifmo.ru/> (раздел «Статьи»).
2. Шалыто А.А., Туккель Н.И. SWITCH-технология — автоматный подход к созданию программного обеспечения "реактивных" систем // Программирование. 2001. № 5, с. 45-62. <http://is.ifmo.ru/> (раздел «Статьи»).
3. Фаулер М. Рефакторинг. Улучшение существующего кода. М.: Символ-Плюс, 2003. — 623 с.
4. Parr T.J., Quong R.W. ANTRL: A Predicated-LL(k) Parser Generator // Software — Practice And Experience. 1995, №25(7). p. 789-810.
5. Ахо А., Сети Р., Ульман Д. Компиляторы: принципы, технологии и инструменты. Вильямс. 2001. — 768 с.
6. Шалыто А.А., Туккель Н. И., Шамгунов Н.Н. Реализация рекурсивных алгоритмов на основе автоматного подхода. // Телекоммуникации и информатизация образования, 2002, № 5.
7. Шалыто А.А., Штучкин А.А., Совместное использование теории построения компиляторов и SWITCH-технологии (на примере построения калькулятора). <http://is.ifmo.ru> раздел «Статьи».
8. Хантер Р. Основные концепции компиляторов. Вильямс. 2002. — 256 с.
9. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. М.: ДМК. 2000. — 320 с.
10. Акимов О.Е. Дискретная математика: логика, группы, графы. М.: Лаборатория Базовых Знаний. 2003. — 376 с.
11. Кормен Т., Лайзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. М.: МЦМНО. 2000. — 960 с.
12. Легалов А. Основы Разработки трансляторов. Использование диаграмм Вирта для представления динамически порождаемых конечных автоматов, распознающих КС(1) грамматику. <http://softcraft.ru/translat/lect/t08-04.shtml>

СОЗДАНИЕ СИСТЕМЫ АВТОМАТИЧЕСКОГО ЗАВЕРШЕНИЯ ВВОДА С ИСПОЛЬЗОВАНИЕМ ПАКЕТА *UNIMOD*

В.С. Гуров, М.А. Мазин

Научный руководитель – д.т.н., профессор А.А. Шалыто

В работе описана технология разработки системы автоматического завершения ввода, предназначенная для удобного редактирования программных текстов. Используя грамматику, задающую язык, предлагается строить конечный автомат типа Мили. Для разработки и отладки конечных автоматов используется программный пакет с открытым кодом *UniMod* (<http://unimod.sf.net>).

Введение

В работе [1] предложен метод проектирования событийных объектно-ориентированных программ с явным выделением состояний, названный «*SWITCH*-технологией». Особенность этого подхода состоит в том, что поведение в таких программах описывается с помощью графов переходов структурных конечных автоматов с нотацией, предложенной в работе [2]. *SWITCH*-технология для описания каждого автомата определяет два типа диаграмм (схемы связей и графы переходов). При наличии нескольких автоматов, кроме того, строится схема их взаимодействия. *SWITCH*-технология задает нотацию и операционную семантику используемых диаграмм.

Программный пакет с открытым кодом *UniMod* (<http://unimod.sf.net>), созданный авторами статьи, обеспечивает разработку и выполнение автоматически-ориентированных программ. Пакет, сохраняя автоматный подход, позволяет использовать *UML*-нотацию при построении диаграмм в рамках *SWITCH*-технологии. При этом схемы связей, определяющие интерфейс автоматов, строятся в нотации диаграмм классов языка *UML*, а графы переходов — в *UML*-нотации диаграмм состояний. В состав пакета *UniMod* входит встраиваемый модуль (*plug-in*) для платформы *Eclipse* (<http://www.eclipse.org>), позволяющий создавать и редактировать *UML*-диаграммы классов и состояний, которые соответствуют схеме связей и графу переходов.

На сегодняшний день интегрированные системы для разработки программ предоставляют удобные средства для работы с кодом, такие как, например:

- подсветка семантических и синтаксических ошибок;
- автоматическое завершение ввода и автоматическое исправление ошибок;
- форматирование и рефакторинг [3] кода;
- запуск и отладка программы внутри среды разработки.

В английском языке эти средства получили название «*code assist*».

В рамках создания очередной версии пакета *UniMod* перед авторами встала задача реализации системы автоматического завершения ввода при редактировании условий на переходах на *UML*-диаграмме состояний. В статье описана технология создания такой системы и ее автоматически-ориентированная реализация, выполненная с помощью пакета *UniMod*.

Постановка задачи

Автоматическим завершением ввода, применительно к редактированию программных текстов, традиционно называют технологию, позволяющую пользователю получить список строк, при добавлении которых в текст после позиции курсора программа будет синтаксически верна. Например, на рис. 1 показано, как среда разработки *Eclipse* предлагает варианты автоматического завершения ввода для текущей позиции курсора.

Сформулируем требования к проектируемой системе автоматического завершения ввода. Пусть задан язык L и на вход системы подана строка α .

1. Если поданная на вход строка α является префиксом предложения языка L ($\exists \omega : \alpha\omega \in L$), то система должна возвращать множество строк $C(\alpha) = \{\beta_i\}_{i=1..n}$, любая из которых может являться продолжением данной α ($\forall i \in [1..n] \exists \gamma : \alpha\beta_i\gamma \in L$).
2. Если поданная на вход строка α не является префиксом предложения на заданном языке ($\neg \exists \omega : \alpha\omega \in L$), то система должна с помощью дополнения строки недостающими символами или с помощью удаления лишних символов трансформировать строку в правильный префикс предложения языка. Количество дополнений и удалений должно быть как можно меньше.

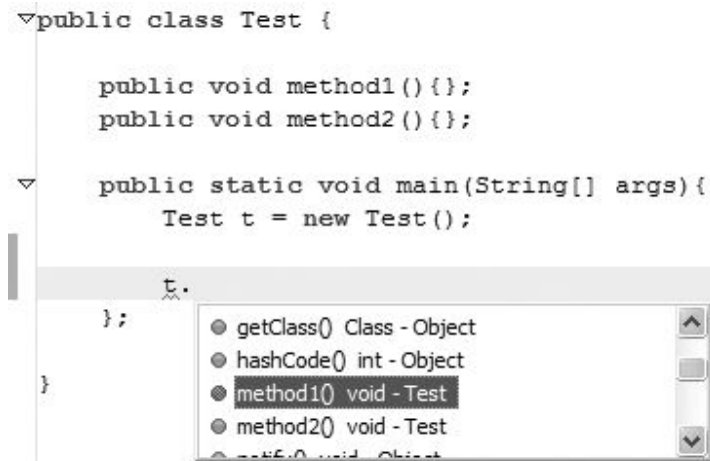


Рис. 1. Пример автоматического завершения ввода

Элементы теории построения компиляторов

Если исходный язык L задан формальной порождающей грамматикой, то очевидно, что для построения такой системы необходимо использовать методы проектирования компиляторов [5]. Существует множество инструментов для автоматического создания компиляторов по заданной грамматике. Достаточно большой список приведен, например, на сайте <http://www.kulichki.net/kit/tools/java.html>.

На рис. 2 приведена обобщенная структура компилятора. Лексический анализатор осуществляет чтение входной цепочки символов и их группировку в элементарные конструкции, называемые лексемами. Синтаксический анализатор осуществляет разбор исходной программы, используя поступающие лексемы, семантический анализ программы и построение промежуточного представления программы. Генератор кода преобразует промежуточное представление программы в объектный код. Генератор кода может быть заменен интерпретатором, при этом вместо генерации объектного кода будет выполняться интерпретация промежуточного представления программы.

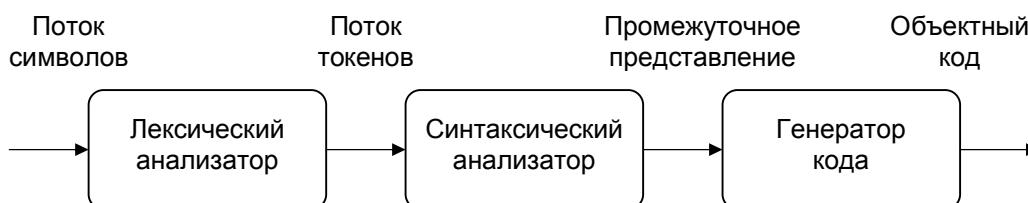


Рис. 2. Обобщенная структура компилятора

В проекте *UniMod* трансляция выражений на переходах выполняется с помощью так называемого «компилятора компиляторов» *ANTLR* [4]. Он по заданной $LL(k)$ грам-

матике строит код на языке *Java*, реализующий лексический анализатор и рекурсивный нисходящий синтаксический анализатор. Построенный синтаксический анализатор может быть использован и как распознаватель принадлежности выражения заданному грамматикой языку, и как транслятор выражений в абстрактное синтаксическое дерево. Данный анализатор не может быть использован для построения системы автоматического завершения ввода, так как в случае подачи ему на вход префикса для выражения на заданном языке вместо законченного выражения он выдает ошибку.

Одним из возможных вариантов реализации требуемой системы может быть использование нерекурсивного нисходящего синтаксического анализатора, явно использующего стек и управляемого таблицей разбора. Таблица разбора представляет собой двумерный массив $M[A, a]$, где A – нетерминал, а a – терминал (лексема) или символ конца потока $\$$. В ячейках таблицы записываются продукции грамматики, с помощью которых заменяются нетерминалы на вершине стека, пустые ячейки таблицы означают ошибки. Подробно работа такого анализатора описана в работе [5].

При подаче на вход описанному выше анализатору незавершенной строки α без символа конца потока анализатор остановится, имея какой-то нетерминал на вершине стека. В этом случае множество терминалов $C(\alpha)$, ожидаемых вслед за обработанной строкой, может быть определено как $\{\beta : M[T, \beta] \neq \emptyset\}$, где T – нетерминал на вершине стека после остановки анализатора.

Для реализации восстановления после ошибок в «режиме паники» таблица разбора может быть дополнена синхронизирующими символами, которые вписываются в некоторые пустые ячейки. При получении неожиданного терминала анализатор пропускает символы входного потока до тех пор, пока не будет обнаружен терминал, соответствующий синхронизирующему символу. Для восстановления на уровне фразы в некоторые пустые ячейки вписываются указатели на подпрограммы обработки ошибок, которые могут изменять, вставлять или удалять терминалы входного потока или элементы стека.

Описание предлагаемой технологии

В [7] показано, как создать программу нерекурсивного нисходящего синтаксического анализатора, используя автоматически-ориентированный подход, при этом таблица разбора в работе [7] оставлена и выступает в роли объекта управления.

В настоящей работе предлагается технология создания системы автоматического завершения ввода, позволяющая исключить таблицу разбора нисходящего нерекурсивного синтаксического анализатора и использующая гибкий алгоритм восстановления после ошибок на уровне фразы.

Технология основывается на том, чтобы для заданной $LL(1)$ грамматики построить конечный автомат типа Мили, который будет являться синтаксическим анализатором. Автомат должен реагировать на события, которые поставляет ему лексический анализатор. Каждому событию соответствует терминал. В работе [5, 12] приведено описание подхода для создания нисходящего синтаксического анализатора на основе диаграмм переходов. При этом предлагается записывать по одной диаграмме для каждого правила вывода грамматики. Описываемая в настоящей статье технология предлагает сворачивать все диаграммы в одну, при необходимости удаляя рекурсию с помощью метода, описанного в работе [6]. Такой подход позволяет избавиться от упоминания нетерминалов на диаграммах переходов и, следовательно, разорвать семантическую связь с исходной грамматикой. Такой разрыв позволит описывать язык только с помощью диаграммы переходов и автоматически получать реализацию распознавателя для данного языка.

При подаче на вход системе, построенной описанным выше образом, незавершенной строки автомат, реализующий синтаксический анализатор, останавливается в каком-то состоянии. События, заданные на переходах из состояния, в котором остановился автомат, определяют множество терминалов, которые могут следовать за последним терминалом, извлеченным из входной строки. После построения такого множества терминалов каждый терминал обратно преобразуется в строку символов.

Построение диаграммы переходов синтаксического анализатора

Пусть $LL(1)$ грамматика для нашего примера задана следующим множеством правил вывода:

- 1) $S \rightarrow \text{else} \mid T S'$
- 2) $S' \rightarrow \text{or} T S' \mid \varepsilon$
- 3) $T \rightarrow L T'$
- 4) $T' \rightarrow \text{and} L T' \mid \varepsilon$
- 5) $L \rightarrow \text{not} L \mid P$
- 6) $P \rightarrow \text{'(' S ')'} \mid \text{int rel N} \mid \text{bool} \mid N P'$
- 7) $P' \rightarrow \text{rel int} \mid \varepsilon$
- 8) $N \rightarrow \text{id dot id}$

Терминал **id** соответствует идентификатору, терминал **int** — целочисленной константе, терминал **bool** — булевой константе, а терминал **rel** — бинарному отношению ('>', '<', '>=', '<=', '=', '≠'). Опишем формальный процесс построения автомата типа Мили для данной грамматики.

На рис. 3 приведены диаграммы переходов для каждого нетерминала заданной грамматики, построенные с помощью метода, описанного в работе [5]. Единственным отличием этих диаграмм от диаграмм, предлагаемых в указанной работе, является наличие выделенных начального и конечного состояний, отображаемых закрашенным кругом и закрашенным кругом внутри окружности, соответственно. Из начального состояния существует всегда только один переход, в конечное состояние также ведет только один переход.

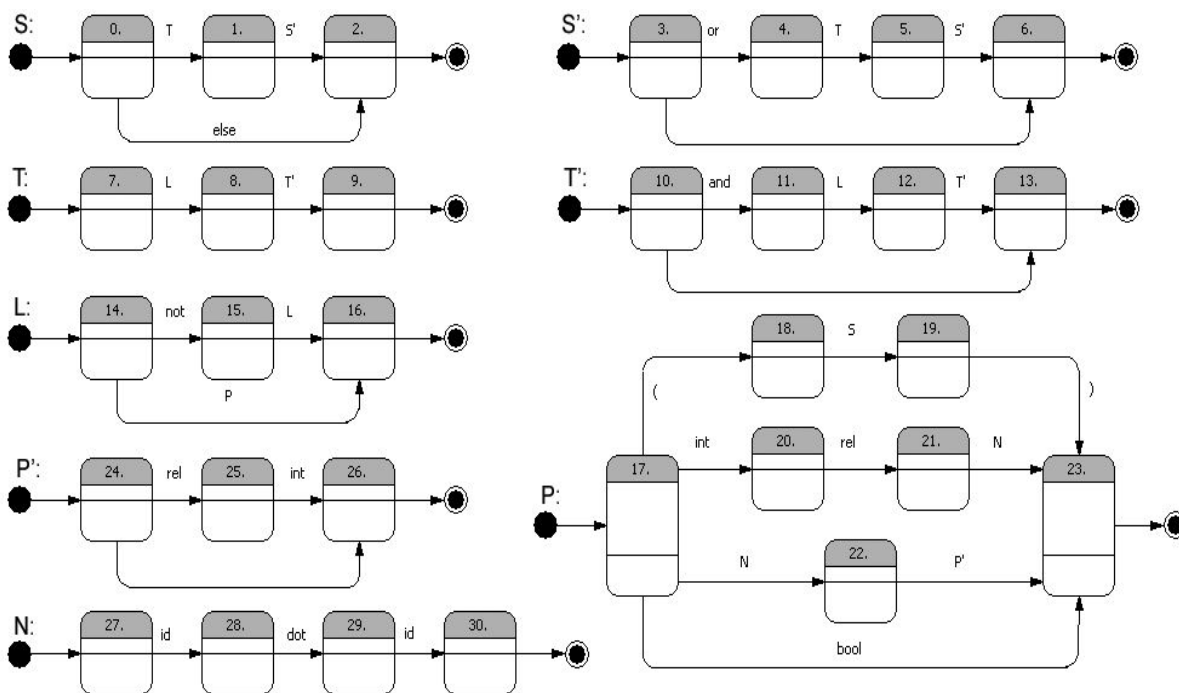


Рис. 3. Диаграммы переходов для каждого нетерминала грамматики

Состояния соответствуют позициям [8] в правилах вывода, метки на переходах – терминалам и нетерминалам, отделяющим позиции друг от друга. Если нетерминал выводит ε -правило, то из состояния, соответствующего начальной позиции, существует непомеченный переход в состояние, соответствующее конечной позиции. Непомеченные переходы также называются немотивированными.

Далее, множество диаграмм, представленных на рис. 3, можно преобразовать в одну диаграмму состояний, на которой все переходы будут помечены только терминалами. Процесс такого преобразования предполагает выполнение следующих шагов:

1. удаление правой рекурсии;
2. удаление немотивированных переходов;
3. подстановка диаграмм переходов друг в друга;
4. удаление срединной рекурсии.

Опишем каждый шаг подробно.

Удаление правой рекурсии

Наличие праворекурсивного правила вывода означает, что на диаграмме, соответствующей некоторому терминалу N , есть переход, помеченный тем же нетерминалом N , ведущий в состояние, соответствующее конечной позиции.

Например, наличие праворекурсивного правила (2) влечет наличие перехода из состояния 5 в состояние 6 на рис. 3. Для устранения правой рекурсии этот переход должен быть заменен немотивированным переходом в состояние, соответствующее начальной позиции – в состояние 3 (рис. 4).

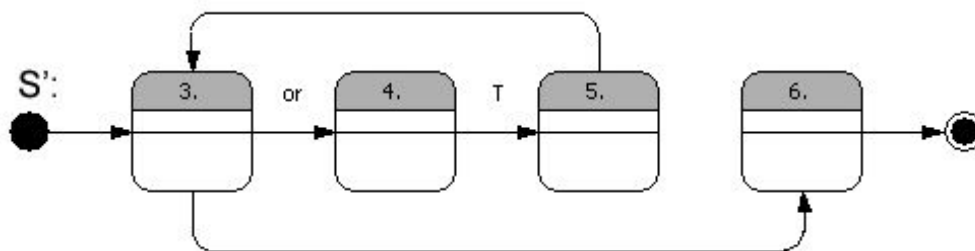


Рис. 4. Удаление правой рекурсии на диаграмме состояний для нетерминала S'

Удаление немотивированных переходов

Наличие немотивированного перехода из состояния S_1 в состояние S_2 , означает, что за позицией, соответствующей состоянию S_1 , могут следовать те же терминалы и нетерминалы, что и за позицией, соответствующей состоянию S_2 .

Для устранения немотивированного перехода выполняются следующие операции:

1. создать сложное состояние $S_{1,2}$ [9];
2. поместить состояния S_1 и S_2 внутрь состояния $S_{1,2}$;
3. все переходы из состояния S_2 заменить аналогичными переходами из состояния $S_{1,2}$.

На рис. 5 показано удаление немотивированного перехода из состояния 5 в состояние 3, присутствующего на рис. 4.

Будем говорить, что исходящий переход из состояния S_1 совпадает с исходящим переходом из состояния S_2 , если он помечен тем же символом и ведет в то же состояние.

После выделения группового состояния $S_{1,2}$ множества переходов, исходящих из состояний S_1 и S_2 , могут совпасть. В этом случае из каждой пары совпадающих переходов следует оставить только один и его начало переместить в состояние $S_{1,2}$. Все переходы, входящие в состояния S_1 и S_2 , перенаправить в состояние $S_{1,2}$. Состояния S_1 и S_2 ликвидировать.

Отметим, что данный алгоритм аналогичен вычеркиванию одинаковых записей из таблицы, задающей функцию переходов автомата [10].

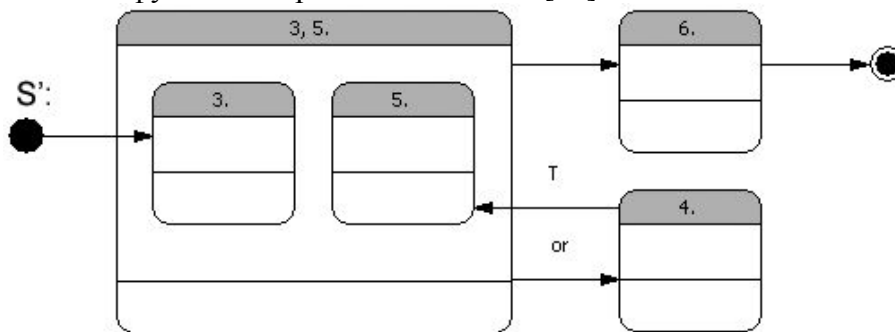


Рис. 5. Удаление немотивированного перехода из состояния 5 в состояние 3

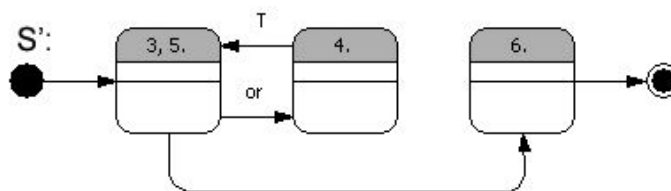


Рис. 6. Упрощение диаграммы состояний, показанной на рис. 5

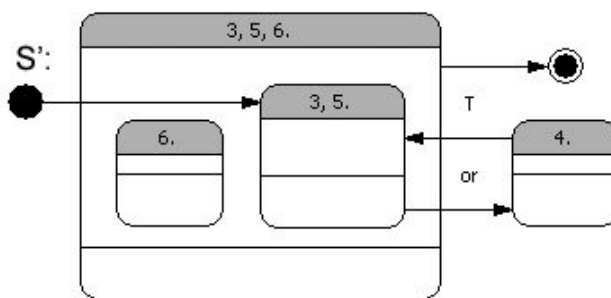


Рис. 7. Удаление немотивированного перехода из состояния 3,5 в состояние 6

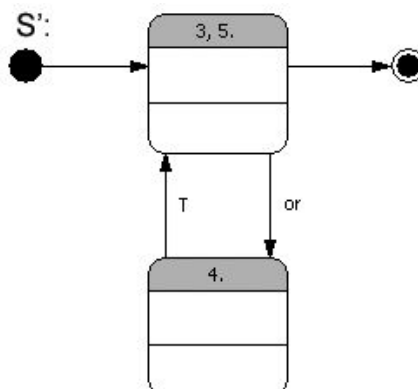


Рис. 8. Преобразованная диаграмма состояний для нетерминала S'

На рис. 5 описанный алгоритм можно применить для состояний 3 и 5. Результирующая диаграмма показана на рис. 6. На рис. 7 показано удаление немотивированного переход из состояния 3,5 в состояние 6. Состояние 6 на рис. 7 не имеет входящих переходов и, следовательно, не достижимо, поэтому оно может быть удалено. После удаления состояния 6 в состоянии 3,5,6 будет вложено единственное состояние 3,5. Начала переходов, исходящих из состояния 3,5,6, следует перенести в состояние 3,5, а само со-

стояние 3,5,6 удалить. На рис. 8 приведена диаграмма состояний для нетерминала S' после всех описанных модификаций.

Подстановка диаграмм переходов друг в друга

Диаграммы переходов могут быть упрощены подстановкой одних диаграмм в другие. Предположим, что на диаграмме для нетерминала N_1 существует переход из состояния S_1 в состояние S_2 , помеченный нетерминалом N_2 . Заменяем такой переход на немотивированный из состояния S_1 в состояние, следующее за начальным на диаграмме переходов для нетерминала N_2 . Добавим переход из состояния, предшествующего конечному, на диаграмме переходов для нетерминала N_2 в состояние S_2 . Отметим, что указанную подстановку необходимо выполнять, только если $N_1 \neq N_2$, так как в противном случае имеет место срединная рекурсия, удаление которой будет описано ниже.

После выполнения такой подстановки возникшие немотивированные переходы следует устранить описанным ранее способом. При этом сначала устраняется немотивированный переход из состояния S_1 , а затем переход в состояние S_2 .

На рис. 9 продемонстрирована подстановка диаграммы переходов для нетерминала L в диаграмму переходов для нетерминала T' . На рис. 10 приведена диаграмма переходов после устранения немотивированных переходов.

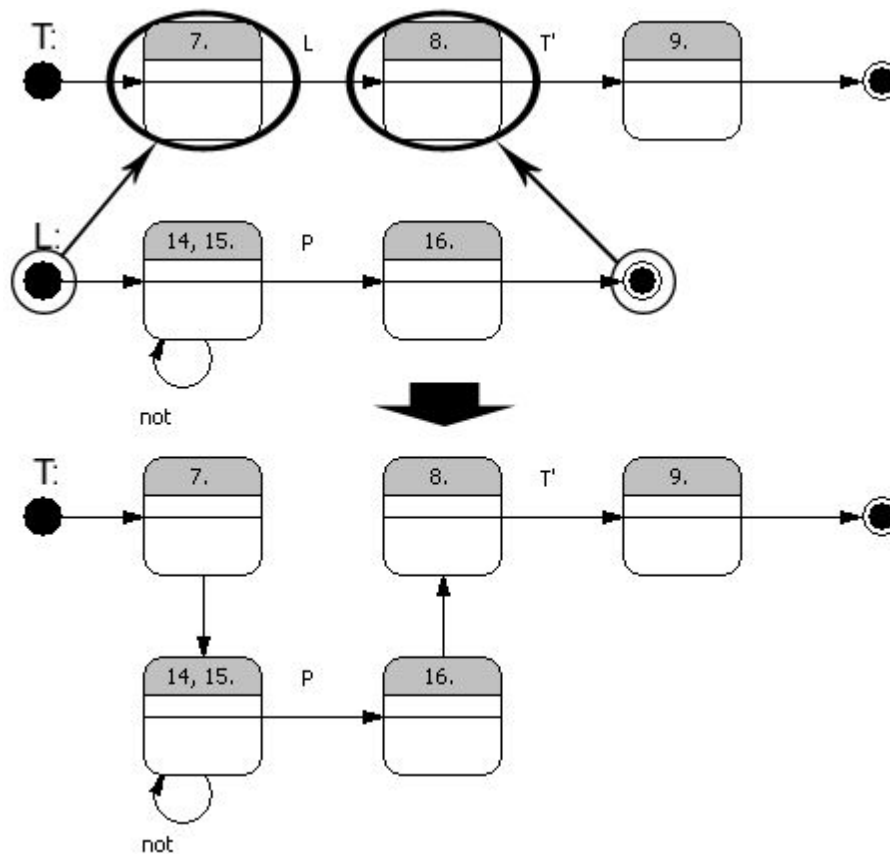


Рис. 9. Подстановка диаграмм переходов друг в друга

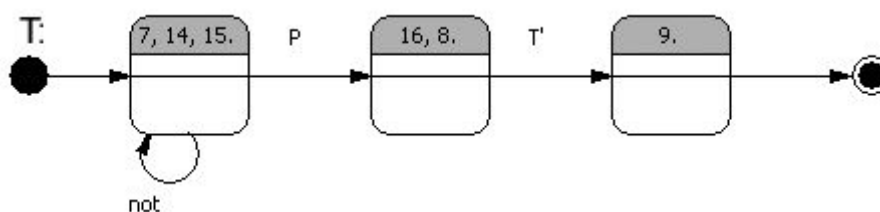


Рис. 10. Устранение немотивированных переходов после подстановки

Если некоторый нетерминал N_1 неоднократно присутствует на диаграмме для нетерминала N_2 , то каждый переход, помеченный N_1 , необходимо заменить соответствующей диаграммой переходов. В результате количество однотипных подграфов на диаграмме N_2 чрезмерно возрастает.

Предлагается преобразовать диаграмму N_2 таким образом, чтобы нетерминал N_1 встречался на ней минимальное число раз. Для этого используется следующий метод: если несколько переходов, помеченных нетерминалом N_1 , ведут в одно и то же состояние S , то можно выделить составное состояние и заменить все эти переходы единственным переходом, помеченным нетерминалом N_1 , исходящим из группового состояния и входящим в состояние S .

На рис. 11 данный метод применен для диаграммы переходов нетерминала S .

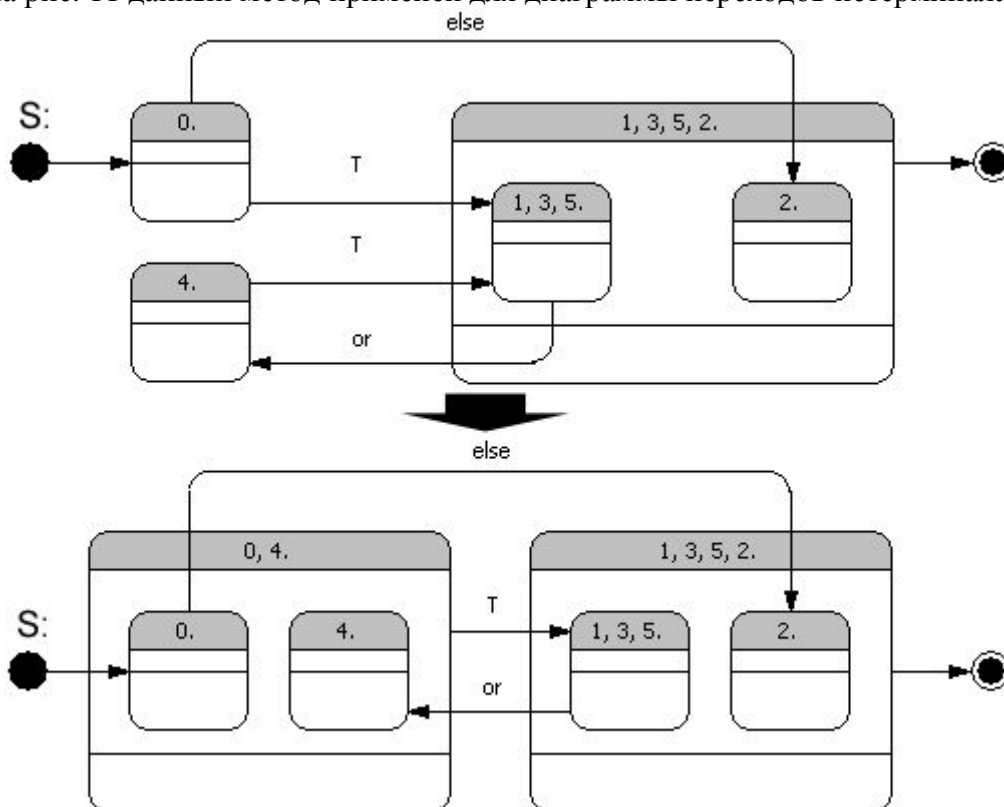


Рис. 11. Удаление двух переходов, помеченных нетерминалом T , на диаграмме переходов для нетерминала S

Удаление срединной рекурсии

В результате выполнения предыдущих шагов исходное множество диаграмм переходов преобразуется во множество диаграмм, возможно, имеющих срединную рекурсию. Для описанной выше грамматики исходное множество диаграмм, показанное на рис. 3, преобразуется в одну диаграмму, приведенную на рис. 12. На этой диаграмме присутствует срединная рекурсия – переход из состояния 18 в состояние 19 .

Для исключения переходов, образующих срединную рекурсию, предлагается использовать метод, предложенный в работе [6]:

- пусть на диаграмме переходов для нетерминала N существует рекурсивный переход из состояний S_1 в состояние S_2 , помеченный нетерминалом N ;
- заменим такой переход двумя немотивированными переходами. При этом первый из них должен выходить из состояния S_1 , а входить в состояние, следующее за начальным состоянием на диаграмме. Второй переход должен выходить из состояния, предшествующего конечному, а входить в состояние S_2 ;

- на первом переходе должно выполняться действие по добавлению в стек метки M_{S_2} , соответствующей исходному целевому состоянию S_2 , а на втором переходе – действие по извлечению метки M_{S_2} из стека, при условии, что метка M_{S_2} находится на вершине стека.

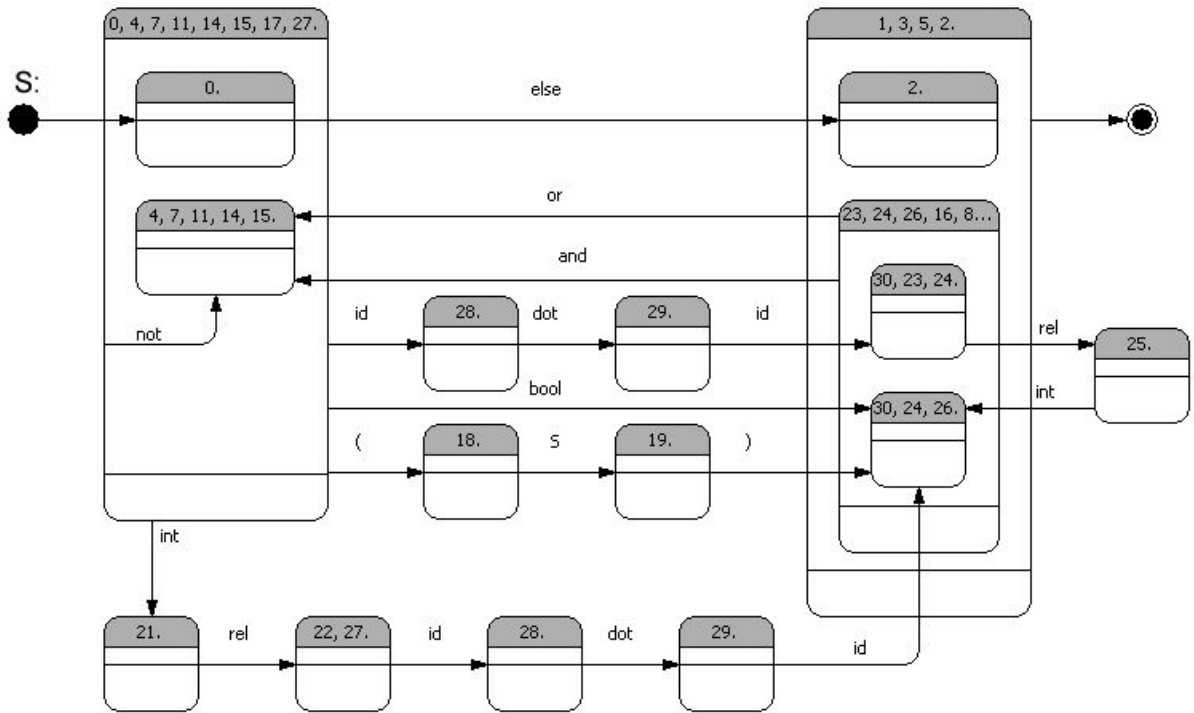


Рис. 12. Диаграмма состояний со срединной рекурсией

На рис. 13 показана диаграмма, приведенная на рис. 12, с удаленной срединной рекурсией.

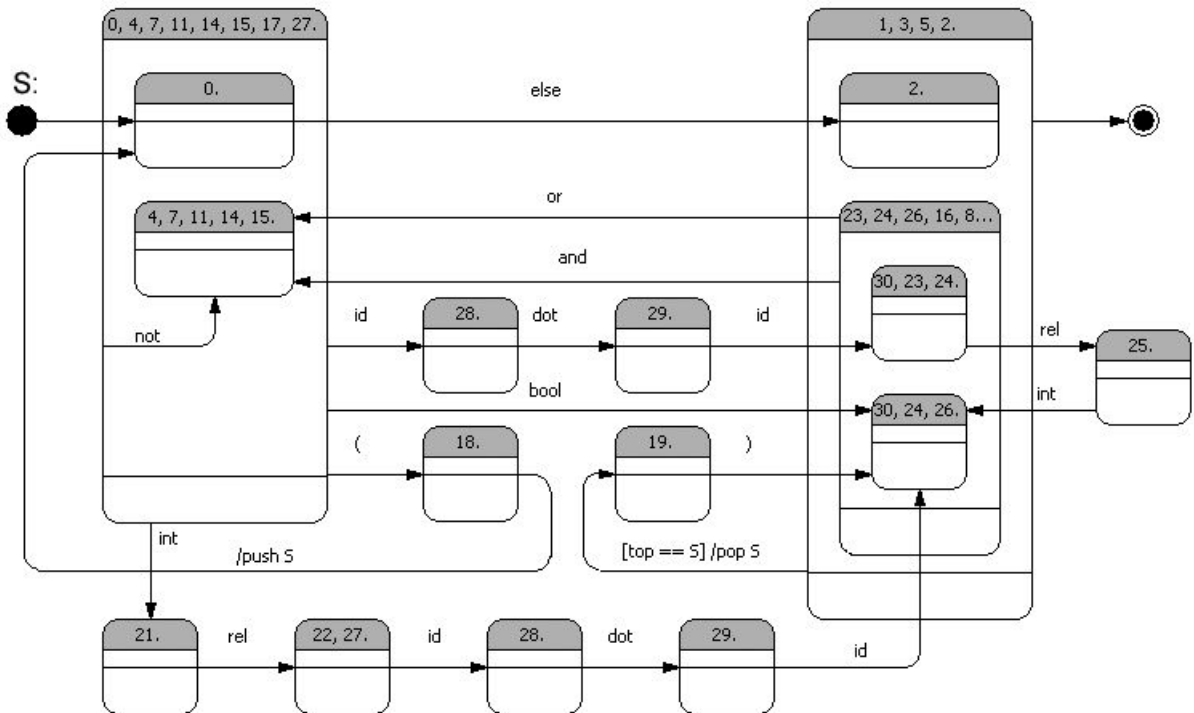


Рис. 13. Диаграмма состояний с удаленной срединной рекурсией

Отметим, что если преобразованное множество диаграмм состоит более чем из одной диаграммы, то после удаления срединной рекурсии в каждой из них необходимо продолжить процесс преобразования с шага 3 – подстановка диаграмм переходов друг в друга.

Модель разрабатываемой системы

Описанные выше шаги приводят к построению диаграммы переходов для автомата типа Мили. Поставим в соответствие каждому терминалу событие, поставляемое лексическим анализатором, и создадим схему связей автомата [1] в виде *UML*-диаграммы классов [9].

На рис. 14 приведена такая схема. При этом на ней слева показан объект *p1*, соответствующий лексическому анализатору, в центре – объект *A*, соответствующий автомату Мили, а справа – объект управления *o1*, соответствующий стеку.

На рис. 15 показана *UML*-диаграмма состояний автомата, построенная на основе диаграммы, приведенной на рис. 13, с помощью замены терминалов событиями и замены действий на переходах ссылками на методы объекта управления *o1*. Состояния 18 и 19 на рис. 15 отсутствуют из-за удаления немотивированных переходов.

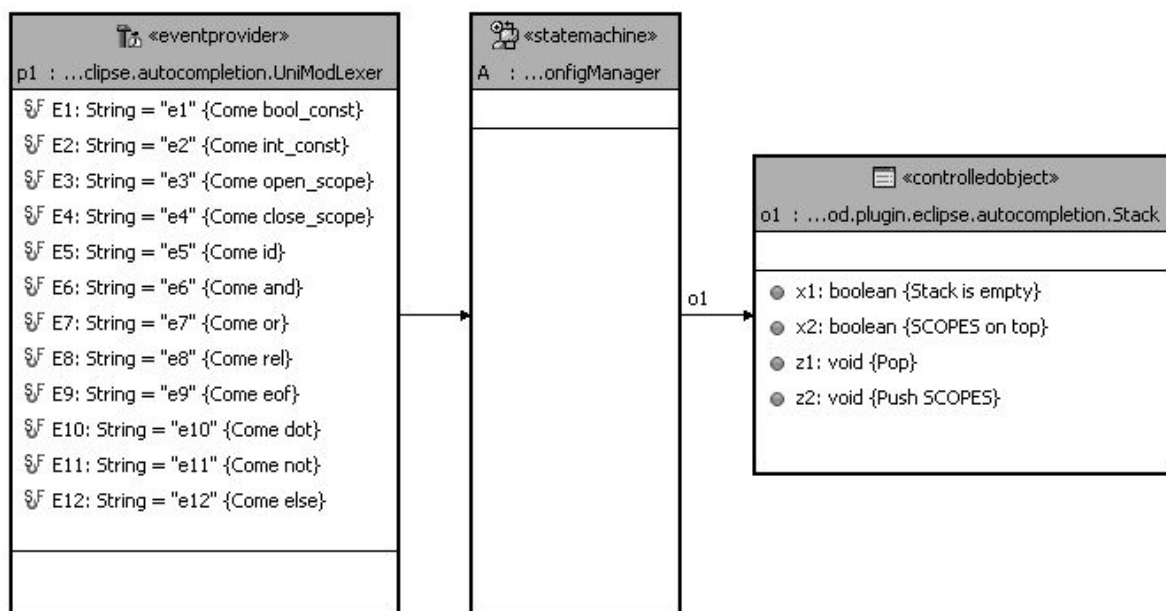


Рис. 14. Схема связей автомата

Полученная модель системы состоит из двух *UML*-диаграмм (рис. 14, 15) и описывает распознаватель для языка, заданного приведенной выше грамматикой. Отметим, что информация о приоритете операций была потеряна в ходе преобразований, и, поэтому, модель может быть использована для распознавания принадлежности выражений языку, но не для трансляции выражений.

Выражение, принадлежащее языку, поданное на вход распознавателю, приводит автомат *A* в финальное состояние. При подаче на вход выражения, являющегося префиксом какого-либо выражения, принадлежащего языку, автомат остановится в каком-то состоянии, множество исходящих переходов из которого определяет множество возможных следующих терминалов.

Если выражение языку не принадлежит и не является префиксом какого-либо выражения, принадлежащего языку, то автомат *A* остановится в состоянии, в котором было получено событие, для которого не существовало исходящих переходов при теку-

щих значениях входных переменных. В этом случае множество возможных следующих терминалов можно определить только для последнего правильно обработанного терминала.

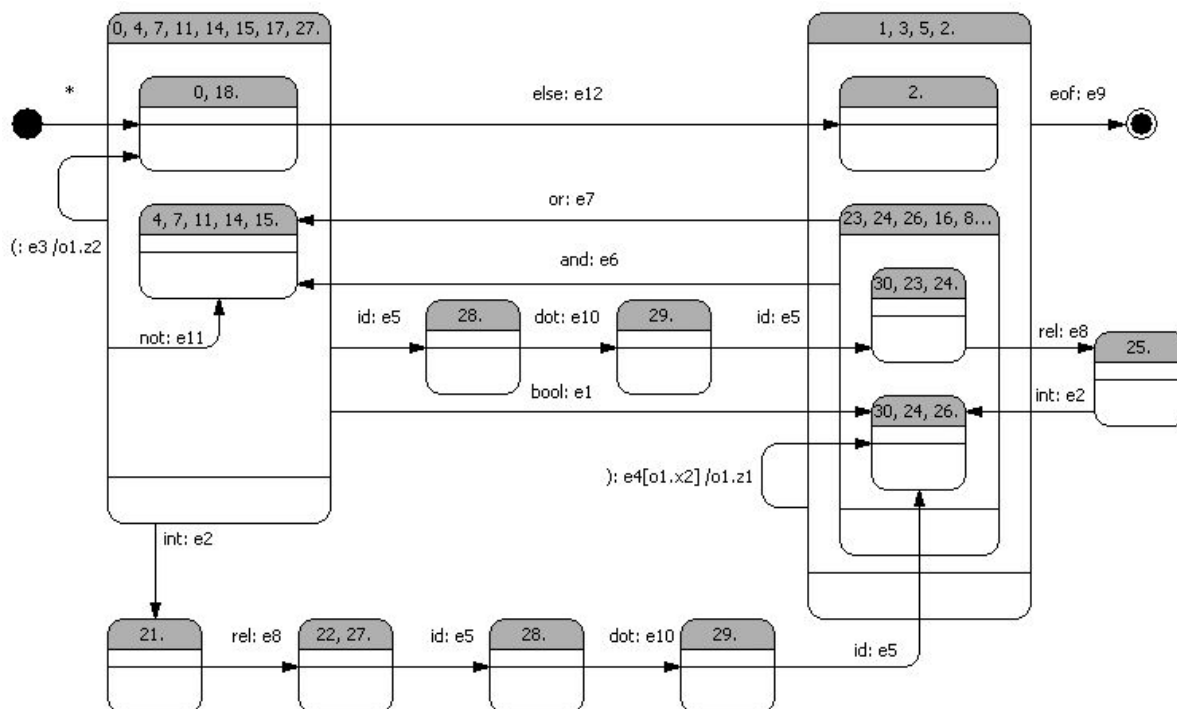


Рис. 15. Диаграмма состояний автомата

Восстановление после ошибок

Перейдем к реализации второго предъявляемого к системе требования – обработка ошибочных строк. Для его реализации автомат A необходимо модифицировать таким образом, чтобы он корректно восстанавливался в случае подачи на вход выражения, не являющегося префиксом какого-либо выражения принадлежащего языку. При этом автомат должен всегда останавливаться в состоянии, в которое существует переход по событию, соответствующему последнему терминалу, извлеченному из поданного на вход выражения.

Существует несколько возможных вариантов реализации восстановления автомата после ошибки [5]. Например, можно для каждого состояния добавить такой исходящий переход, ведущий в конечное «ошибочное» состояние, что он будет срабатывать в случае отсутствия какого-либо другого исходящего перехода для пришедшего события и текущих значений входных переменных. Это приведет к тому, что при появлении в процессе распознавания первого же ошибочного терминала автомата завершит работу в «ошибочном» состоянии. Однако из «ошибочного» состояния нет исходящих переходов, и, следовательно, множество возможных последующих терминалов будет пустым.

Авторы предлагают использовать альтернативный алгоритм обработки ошибочных ситуаций, основанный на локально оптимальной коррекции входного потока терминалов от лексического анализатора. Такой подход также называют восстановлением на уровне фразы.

Пусть из состояния S нет исходящего перехода для пришедшего события e , соответствующего некоторому терминалу. Тогда коррекция потока может осуществляться двумя способами: дополнением потока недостающими терминалами; пропуском лишних терминалов в потоке.

Чтобы автомат A , находясь в состоянии S , пропустил в потоке терминал, соответствующий пришедшему событию e , необходимо добавить в автомат петлю в состоянии S по событию e . Тогда, находясь в состоянии S и получив событие e , автомат останется в состоянии S , т.е. проигнорирует пришедшее событие и, как следствие, пропустит терминал в потоке.

Чтобы автомат A , находясь в состоянии S , дополнил поток недостающими терминалами, следует выполнить указанные ниже операции:

1. найти достижимое из S состояние S_h , такое, что в нем существует исходящий переход по событию e . Если из состояния S достижимы несколько таких состояний, то выберем ближайшее из них;
2. если из ближайшего найденного состояния S_h есть переход в некоторое состояние S_t по событию e при условии c , то необходимо добавить в автомат переход из состояния S в состояние S_t по событию e при условии c . Отметим, что отсутствие условия трактуется как тождественная истина.

Последовательность терминалов, соответствующих событиям, которыми помечен кратчайший путь из состояния S в состояние S_h , можно использовать для вставки в поток перед терминалом, соответствующим пришедшему событию e .

Если лексический анализатор позволяет заглядывать на произвольное количество терминалов вперед, то можно применять оба способа коррекции одновременно, выбирая оптимальный способ в процессе разбора.

Для выбора оптимального способа авторы предлагают использовать следующее правило:

1. при получении ошибочного терминала в текущем состоянии вычислим количество терминалов, которыми нужно дополнить поток;
2. вычислим количество терминалов, которое нужно пропустить в потоке, до следующего обрабатываемого в текущем состоянии терминала;
3. выполним коррекцию, требуемое количество терминалов для которой минимально.

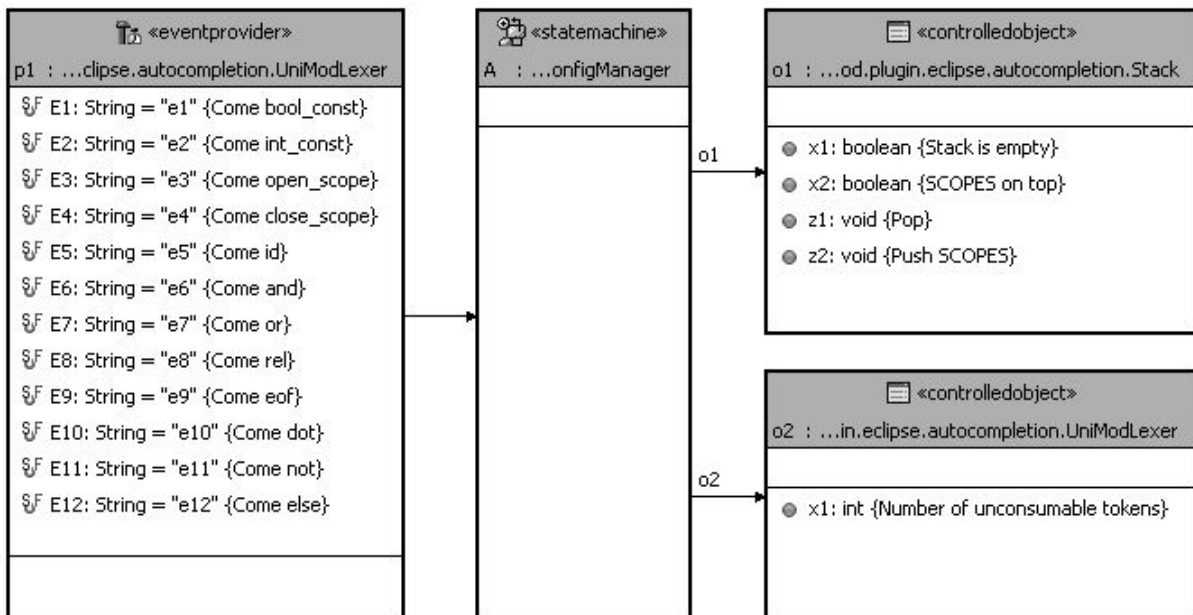


Рис. 16. Схема связей модели распознавателя с лексическим анализатором в качестве объекта управления

Для реализации этого способа коррекции к автомату A распознавателя в качестве объекта управления добавим лексический анализатор (рис. 16). Лексический анализатор предоставляет автомату распознавателя целочисленную входную переменную $o2.x1$. Ее значение равно числу терминалов, которые необходимо пропустить в потоке,

до следующего терминала, обрабатываемого в текущем состоянии. Если входной поток вообще не содержит терминалов, обрабатываемых в текущем состоянии, то значение переменной $o2.x1$ больше любого наперед заданного целого числа.

В автомат A добавляются переходы, реализующие и добавление и пропуск терминалов в потоке. Переходы, реализующие дополнение потока, помечаются следующим условием: длина пути из состояния S в состояние S_h меньше или равна значению входной переменной $o2.x1$. Переходы, реализующие пропуск терминалов, помечаются отрицанием того же условия. Если для состояния S не существует состояния S_h , то переход, удаляющий лексему, выполняется безусловно.

Например, в состоянии 21 нет переходов по событию $e10$ – в этом состоянии появление во входном потоке терминала **dot** (точка) не ожидается. Чтобы обработать ошибочное появление этого терминала, необходимо добавить два перехода, исходящих из состояния 21 (рис. 17). Ближайшее состояние, в котором обрабатывается событие $e10$ – состояние 28 . Длина пути из состояния 21 в состояние 28 равна двум. Поэтому условие на петле в состоянии 21 по событию $e10$ имеет вид $o1.x1 < 2$. Таким образом, в случае, если сразу за терминалом **dot** в потоке следует терминал **rel** (отношение), то терминал **dot** игнорируется. Если следует какой-нибудь другой терминал, то целесообразно сразу перейти в состояние 29 , то есть добавить в поток отсутствующие терминалы **rel** и **id** (идентификатор).

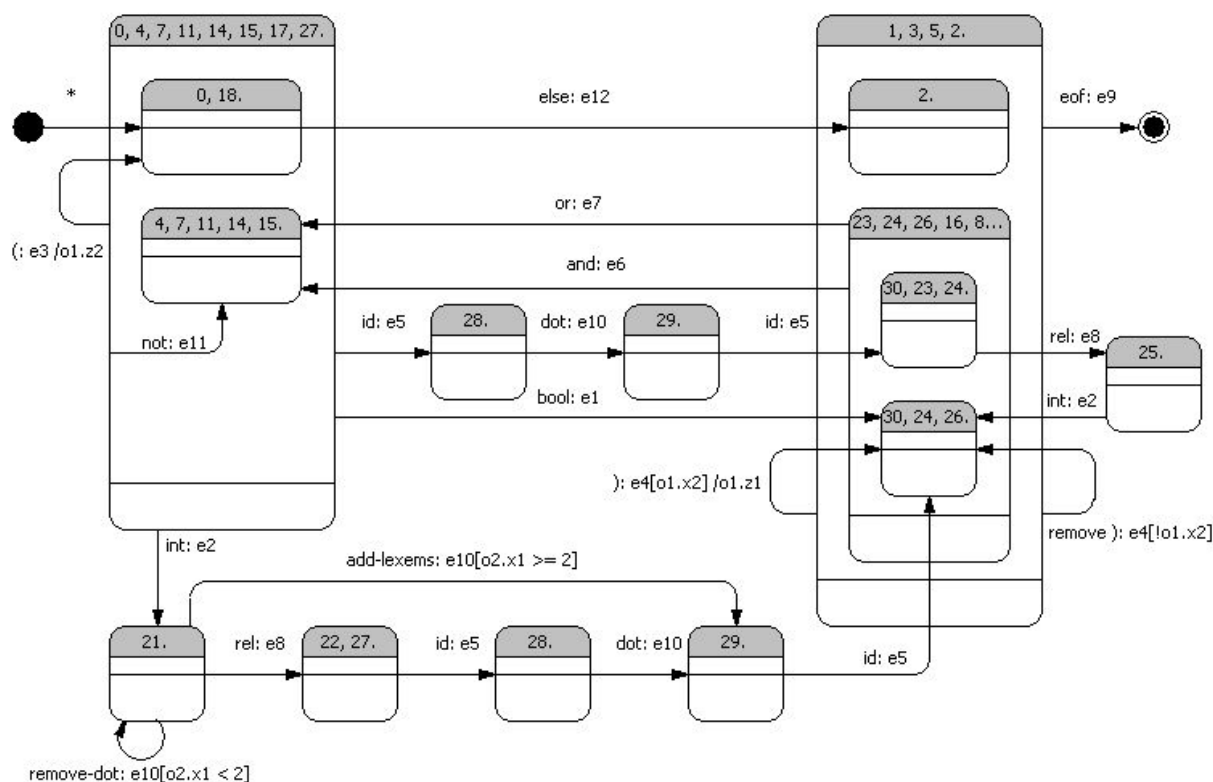


Рис. 17. Добавление переходов, корректирующих поток, в состоянии 21

Описанные выше преобразования могут быть выполнены автоматически для любой диаграммы переходов, так как ближайшее состояние, в котором обрабатывается неожиданный терминал для данного состояния, может быть вычислено, используя, например, алгоритм Флойда-Уоршала [11].

Получение множества строк для автоматического завершения ввода

Предлагаемый алгоритм коррекции входного потока позволяет вычислить множество вариантов завершения как для выражений, являющихся префиксами принадлежащих языку выражений, так и для ошибочных выражений.

После того, как автомат распознавателя, дополненный корректирующими переходами, обработает все терминалы, извлеченные из поданного на вход выражения, он окажется в некотором состоянии S . Для построения множества вариантов завершения следует определить множество переходов, исходящих из S , условия на которых при текущих значениях входных переменных истинны. Терминалы, соответствующие событиям, которыми помечены эти переходы, должны быть преобразованы обратно во множество строк. Например, терминал **id** должен быть преобразован во множество имен переменных, а терминал **and** – в строку «&&». Полученное множество строк и будет множеством вариантов завершения.

Пример работы системы

Приведем пример построения множества вариантов завершения.

Пусть на вход распознавателю подана строка:

```
! o1.x1 &&.
```

Лексический анализатор преобразует ее в поток терминалов:

```
not id dot id and,
```

которому соответствует последовательность событий:

```
e11, e5, e10, e5, e6.
```

В процессе обработки этих событий автомат изменяет состояния в следующем порядке:

```
(0, 18) → (4, 7, 11, 14, 15) → (28) → (29) → (30, 23, 24) → (4, 7, 11, 14, 15).
```

Состояние $(4, 7, 11, 14, 15)$, в котором остановился автомат, содержит исходящие переходы для событий

```
e1, e2, e3, e5, e11.
```

Этим событиям соответствуют терминалы

```
bool, int, '(', id, not,
```

которые преобразуются в строки:

```
"true", "false", "(", "o1", "o2", "o22", "!".
```

Эти строки и формируют множество вариантов завершения для строки, поданной на вход распознавателю.

На рис. 18 показан фрагмент среды разработки с встроенной системой автоматического завершения ввода, описанной в настоящей статье.

Property	Value
Event	e9
Guard	! o1.x1 &&
Name	(
Output	! o1 o2 o22 true false

Рис. 18. Пример автоматического завершения ввода

Заключение

В работе [5] отмечено, что нерекурсивные нисходящие синтаксические анализаторы можно строить, используя диаграммы переходов, записанные для каждого нетерминала исходной грамматики. Настоящая работа предлагает подход для построения всего одной диаграммы переходов для исходной грамматики. На базе построенной диаграммы реализуется система автоматического завершения ввода. Также отметим, что в известной авторам литературе описание формального метода построения подобных систем отсутствует. Данная работа устраняет указанный пробел.

Реализация системы автоматического завершения ввода для следующей версии проекта *UniMod* выполнена с помощью предыдущей версии проекта, вследствие чего часть проектной документация была получена «автоматически», так как диаграммы, созданные с помощью *UniMod* редактора, являются автоматными программами и могут быть включены в проектную документацию без изменений. Разработка последующих версий средств разработки с помощью предыдущих является общепринятой практикой и позволяет говорить о зрелости программного продукта.

Литература

1. Шалыто А.А., Туккель Н.И. Танки и автоматы // ВУТЕ/Россия. 2003. № 2. С. 69–73. <http://is.ifmo.ru/> (раздел «Статьи»).
2. Шалыто А.А., Туккель Н.И. SWITCH-технология – автоматный подход к созданию программного обеспечения «реактивных» систем // Программирование. 2001. № 5. С. 45–62. <http://is.ifmo.ru/> (раздел «Статьи»).
3. Фаулер М. Рефакторинг. Улучшение существующего кода. М.: Символ-Плюс, 2003. 623 с.
4. Patг T.J., Quong R.W. ANTRL: A Predicated-LL(k) Parser Generator // Software — Practice And Experience. 1995. №25(7). P. 789–810.
5. Ахо А., Сети Р., Ульман Д. Компиляторы: принципы, технологии и инструменты. М.: Вильямс, 2001. 768 с.
6. Шалыто А.А., Туккель Н. И., Шамгунов Н.Н. Реализация рекурсивных алгоритмов на основе автоматного подхода. // Телекоммуникации и информатизация образования. 2002. № 5.
7. Шалыто А.А., Штучкин А.А., Совместное использование теории построения компиляторов и SWITCH-технологии (на примере построения калькулятора). <http://is.ifmo.ru> (раздел «Статьи»).
8. Хантер Р. Основные концепции компиляторов. М.: Вильямс, 2002. 256 с.
9. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. М.: ДМК, 2000. 320 с.
10. Акимов О.Е. Дискретная математика: логика, группы, графы. М.: Лаборатория Базовых Знаний, 2003. 376 с.
11. Кормен Т., Лайзерсон Ч., Ривест Р. Алгоритмы. Построение и анализ. М.: МЦМНО. 2000. 960 с.
12. Легалов А. Основы разработки трансляторов. Использование диаграмм Вирта для представления динамически порождаемых конечных автоматов, распознающих КС(1) грамматику. <http://softcraft.ru/translat/lect/t08-04.shtml>

ПРОЕКТИРОВАНИЕ ПОДСИСТЕМЫ ФИЛЬТРАЦИИ ДАННЫХ ДЛЯ УНИВЕРСАЛЬНОГО КОНТРОЛЛЕРА УПРАВЛЕНИЯ АВТОРУЛЕВЫМ УСТРОЙСТВОМ

А.М. Озерский

Научный руководитель – д.т.н., профессор И.П. Гуров

Рассмотрен подход к проектированию универсальных систем управления авторулевыми устройствами и подсистем фильтрации данных для таких устройств. Предложен метод фильтрации данных с использованием дискретной фильтрации Калмана с учетом характеристик управляемой системы.

Введение

Управление авторулевым устройством с целью удержания судна на маршруте является важной прикладной задачей, требующей, в частности, применения широкого спектра методов фильтрации и анализа сигналов.

Авторулевым устройством называется комплекс систем, способный удерживать судно на заданном курсе в пределах некоторого диапазона внешних условий. Система управления авторулевым устройством, или контроллер авторулевого устройства, предназначена для выработки значения курса, необходимого для удержания судна на заранее выбранном маршруте.

В процессе такого управления необходим анализ сигналов от различных датчиков, в том числе гироскопа, магнитного компаса, различных типов лагов и спутниковых систем позиционирования, таких как GPS или Глонасс. При вычислении управляющих воздействий необходимо учитывать как математическую модель судна, так и модель, описывающую алгоритмы управления авторулевого устройства.

При общепринятом подходе к проектированию контроллера авторулевого устройства алгоритмы такого контроллера интегрируются с алгоритмами самого авторулевого устройства, что позволяет обеспечить максимальную точность управления и гибкость настроек такой системы. Однако при таком подходе необходимо, чтобы авторулеванное устройство и его контроллер разрабатывались совместно, и разработанный контроллер будет пригоден лишь для управления соответствующим авторулевым устройством.

В ряде случаев более перспективным представляется создание универсального контроллера авторулевого устройства, не столь тесно интегрированного с авторулевым устройством и обеспечивающего при этом возможность подключения нескольких различных типов авторулевых устройств. Такой подход позволяет создать единую систему управления и этим уменьшить влияние человеческого фактора за счет облегчения обучения судоводителей.

Далее в статье будут рассмотрены проектирование подсистемы фильтрации данных, предназначенной для выработки управляющих воздействий на различные типы авторулевых устройств.

Основы построения математической модели судна для управления авторулевым устройством

Задача проектирования системы для управления направлением движения судна путем изменения угла кладки руля решается при разработке внутренних алгоритмов и аппаратуры авторулевого устройства. Наиболее распространенный подход к этой задаче заключается в том, чтобы построить математическую модель судна, по возможности используя наименьшее число параметров.

Так, например, для оценки угла дрейфа (угол между направлением с носа судна на корму и направлением движения центра вращения судна) при установившейся угловой скорости вращения судна используют формулу

$$\beta_C = \frac{-q + \sqrt{q^2 + 4h_1 r_{31} s \delta_R}}{2h_1 r_{31}}, \quad (1)$$

где β_C – угол дрейфа, δ_R – угол перекладки руля, q_i , r_i , h_i , s_i – коэффициенты, а для оценки угловой скорости в том же режиме вращения судна – формулу

$$\bar{\omega}_C = \frac{s_{31} \delta_R + q_{31} \beta_C}{r_{31}}, \quad (2)$$

где $\bar{\omega}_C$ – угловая скорость.

Таким образом, при работе авторулевого в соответствии с выбранным алгоритмом удержания курса определяется стратегия выхода на нужный курс, после чего в зависимости от выбранной математической модели судна производится поддержание необходимой угловой скорости вращения для выхода на нужный курс. При этом в различных моделях авторулевых устройств даже для одного судна могут использоваться различные стратегии удержания курса и различные мат. модели.

Проектирование модели управляемого объекта для универсального контроллера авторулевого устройства

В отличие от рассмотренной выше математической модели судна, используемой для создания алгоритмов авторулевых устройств, в случае универсального контроллера управляемым объектом является связка судна и авторулевого устройства, как указано на рис. 1.

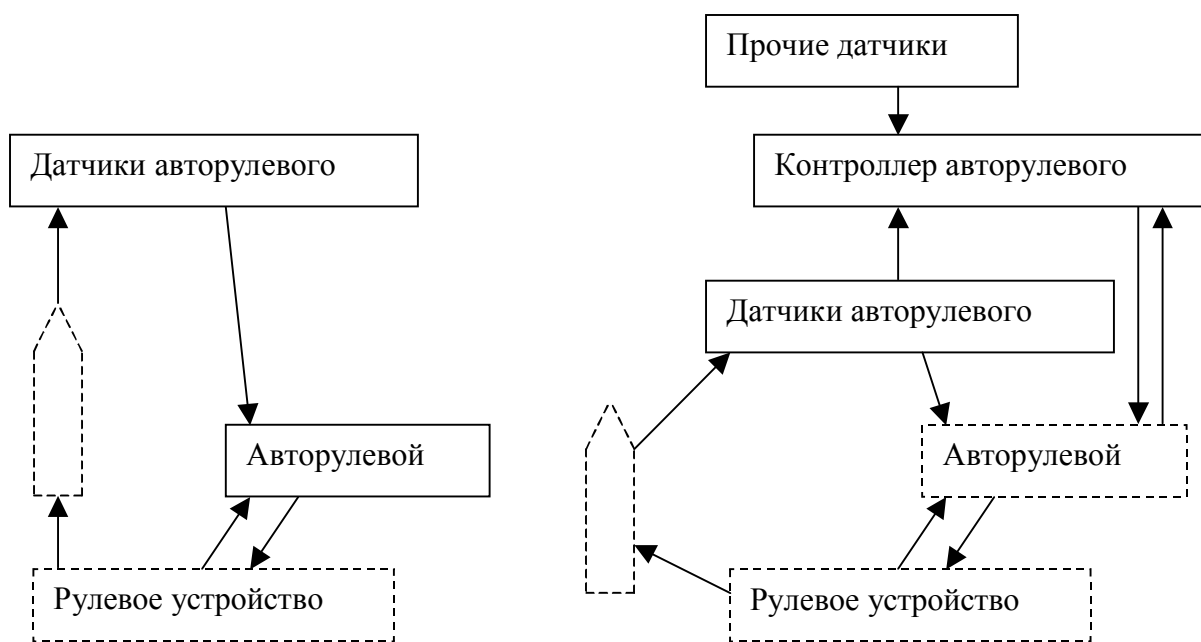


Рис. 1. Управление судном в случае авторулевого устройства (слева) и контроллера авторулевого устройства (справа). Пунктиром выделены управляемые объекты

Подсистема фильтрации данных в рамках контроллера авторулевого устройства должна обеспечивать надежную фильтрацию данных в реальном времени. Кроме этого, немаловажным является также обеспечение верифицируемости соответствующего программного кода, и, следовательно, простота алгоритмов фильтрации данных. При проектировании подсистемы фильтрации данных по возможности должна быть обеспечена независимость различных каналов обработки данных, чтобы избежать выхода из строя системы при отказе одного из датчиков.

При этом авторулевое устройство следует рассматривать как «черный ящик» с заранее неизвестными алгоритмами управления движением судна. Таким образом, при проектировании универсальных алгоритмов управления движением судна следует учитывать наиболее общие свойства различных авторулевых устройств, что позволит добиться универсальности системы управления.

К таким свойствам, например, можно отнести типичное время реакции на управляющее воздействие, а также время отработки управляющего воздействия. В этом случае систему в установившемся режиме можно описать уравнением

$$H(t) = H(t-k) + \omega(t-k) \cdot (t-k) + A(C(t-k)) + n(t), \quad (3)$$

где $H(t)$ – направление движения судна в момент времени t , $\omega(t)$ – угловая скорость судна, $A(C(t))$ – влияние действий авторулевого устройства, основанных на векторе управляющих воздействий C в момент времени t , k – время реакции системы «авторулевой+судно», $n(t)$ – шум. Таким образом, для управления конкретным авторулевым на конкретном судне достаточно оценить время k и отклик авторулевого устройства на команду $A(C(t))$, а также произвести фильтрацию сигнала с целью уменьшить воздействие шумов. Отклик $A(C(t))$ может быть представлен в виде многомерной таблицы, количество измерений которой совпадает с размерностью вектора $C(t)$. Таблица A может быть построена по результатам измерений или экстраполяцией.

Данные, принимаемые датчиков, отслеживающим движение судна, можно рассматривать как сумму полезных сигналов и шума. При этом к полезным сигналам относятся, например, сигналы, позволяющие определить средний снос судна вследствие течения, а к шумам – воздействие отдельных волн или порывов ветра. Полезность сигнала в данном случае означает необходимость выдачи соответствующего полученному сигналу управляющего воздействия на авторулевоу с целью компенсации отклонения судна от заданного маршрута.

Таким образом, как легко заметить, полезные сигналы обладают значительно большей длительностью, чем шум. Критерием полезности сигнала может служить тот факт, что авторулевоу устройство способно скомпенсировать полезный сигнал за время, меньшее, чем длительность возмущающего воздействия. Следовательно, задачу фильтрации полезных сигналов можно свести к задаче выделения сигналов с периодом большим, чем характерное время реакции системы. При этом основным свойством модели управляемой системы являются её временные характеристики, описывающие длительность переходных процессов.

Учитывая необходимость реагирования в реальном времени, а также требования к простоте используемого алгоритма фильтрации, которые вытекают из необходимости тщательной верификации используемых в подобных системах алгоритмов, оптимальным для решения данной задачи представляется использование фильтров Калмана первого порядка.

Предположим, что измеренный сигнал описывается уравнением наблюдения

$$s_e(t) = s_1(t) + n(t), \quad (4)$$

где $s_e(t)$ – измеренное значение сигнала в момент времени t , $s_1(t)$ – данные, соответствующие текущему состоянию измеряемого объекта, $n(t)$ – шум наблюдения. Тогда линейный дискретный фильтр Калмана первого порядка описывается уравнениями

$$\hat{s}(t) = \alpha s_p(t-1) + \beta (s_p(t-1) - s_e(t)), \quad (5)$$

$$s_p(t) = \hat{s}(t-1),$$

где $\hat{s}(t)$ – оценка сигнала, $s_p(t-1)$ – предсказание значения сигнала в момент времени $t-1$, α и β – коэффициенты. В момент времени $t=0$ предполагается $\hat{s}(0) = s_e(0)$.

Такие фильтры могут использоваться для выделения низкочастотных составляющих сигнала. Полоса пропускания фильтра может варьироваться, например, в зависи-

мости от модели судна, настроек авторулевого или погодной обстановки. Пример фильтрации данных курса от датчика гирокомпаса с использованием фильтра Калмана первого порядка приведен на рис. 2. Аналогичная обработка может быть применена и к данным других датчиков.

Ошибка! Ошибка связи.

Рис. 2. Результат фильтрации данных курса от датчика гирокомпаса

Таким образом, данные из каждого канала будут обрабатываться независимо, что повышает общую надежность системы.

Заключение

Алгоритмы универсального контроллера авторулевого устройства значительно отличаются от классических алгоритмов управления конкретным авторулевым устройством. Предложенный подход с использованием линейного фильтра Калмана и обобщенной модели авторулевого устройства позволяет реализовать такие алгоритмы, отличающиеся простотой реализации и верификации, а также относительно небольшим числом параметров модели управляемого объекта. Система управления движением судна, разработанная на основе описанных принципов, показала достаточную точность управления различными типами авторулевых устройств, как на модельных испытаниях, так и при эксплуатации на нескольких судах.

Литература

1. Бендат Дж., Пирсон А. Прикладной анализ случайных данных. М.: Мир, 1989.
2. Гуров И.П., Жиганов П.Г., Озерский А.М. Анализ акустических сигналов на основе метода фильтрации Калмана. // Научно-технический вестник СПбГИТМО (ТУ). 2002. Вып. 6. С. 200–206.
3. Макс Ж. Методы и техника обработки сигналов при физических измерениях / М.: Мир, 1983.
4. Обнаружение изменения свойств сигналов и динамических систем / Под ред. Бассвиля М. М.: Мир, 1989.
5. Справочник по теории корабля / Под ред. Войткунского Я.И. Л.: Судостроение, 1985.
6. International standard IEC 62065 2002-03 – Switzerland: IEC Central Office, 2002.
7. Kalman R.E. A new approach to linear filtering and prediction problems. // Trans. ASME, J. Basic Eng. 1960. V.82. P. 35–45.

ОЦЕНКА КАЧЕСТВА ПЕРЕДАЧИ ГОЛОСА В СЕТЯХ IP-ТЕЛЕФОНИИ

М.Ю. Будько

Научный руководитель – д.т.н., профессор Т.И. Алиев

В настоящей работе анализируются стандарты, используемые для оценки качества передачи голоса, рассматриваются способы их применения в конкретных системах. Предлагается использование метода статистического анализа для оценки качества передачи голоса в определенных направлениях.

Введение

Проблема оценки качества услуг телефонии была актуальна всегда. Как минимум, об этом свидетельствует тот факт, что организация ITU имеет отдельное направление, в рамках которого разрабатываются стандарты, относящиеся к вопросам качества телефонной связи. Использование методов цифровой обработки сигналов внесло новые проблемы в эту сферу деятельности, а в настоящее время к ним прибавились последствия переноса голоса в сети передачи данных.

Основная сложность использования сетей передачи данных для транспортировки голоса заключается в особенностях их организации, а именно в работе по принципу «наибольших усилий». В этом случае сеть выделяет данным столько ресурсов, сколько имеется на момент их появления, что является причиной возникновения задержек и потери пакетов. Такая ситуация не является критичной для обычных приложений, но крайне неблагоприятна для голосового трафика.

Основными факторами, влияющими на качество работы IP-телефонии, являются:

- задержка, т.е. время, необходимое пакетам для прохождения от источника к получателю;
- джиттер или время, на которое изменяется значение задержки;
- нарушение порядка следования пакетов;
- потеря пакетов.

Оценивая эти величины, можно достаточно точно определить степень соответствия тех или иных каналов для передачи мультимедийной информации. Однако это не единственный способ.

Обзор существующих способов оценки качества передачи голоса

Рассмотрим некоторые основные стандарты организации ITU, в которых представлены требования к среде передачи голосовых данных, а также приводятся способы оценки качества работы голосовых сервисов.

1. P.11 – Эффекты, вносимые средой передачи.
2. P.800 – Методы субъективной оценки качества передачи.
3. P.830 – Субъективная оценка работы кодеков телефонного и широкополосного диапазонов.
4. P.861 – Объективная оценка качества голосовых кодеков телефонного диапазона (300–3400 Hz).
5. G.113 – Искажения при передаче, возникающие при обработке голоса.
6. G.114 – Время передачи в одну сторону.

Как видно из представленного списка, методы оценки разбивают на две группы – субъективные и объективные. Это связано с тем, что телефония существует исключительно для людей, и их оценки всегда будут весомее любого алгоритма. Чтобы определить качество работы объективного метода, необходима его привязка к каким-либо эталонным критериям, выработка которых может осуществляться только с помощью субъективных методов.

В качестве основного критерия качества речи организацией ITU используется показатель MOS (Mean Opinion Score), принимающий значения в соответствии с пятибалльной шкалой от единицы до пяти. Оценка MOS введена в стандарте P.800 [1] (ранее P.80) и широко используется в других спецификациях, когда необходимо продемонстрировать влияние того или иного параметра на общую картину соединения. Таким образом, приводится следующее соответствие MOS и субъективных впечатлений о качестве речи: 5 – отличное (excellent), 4 – хорошее (good), 3 – довольно хорошее (fair), 2 – плохое (poor), 1 – очень плохое (bad). Приведенные оценки в соответствии со стандартом выставляются оценщиками по результатам прослушивания голосового канала. Кроме этого, в стандарте P.800 рассматриваются различные показатели, позволяющие оценить, например, сложность восприятия речи или уровень громкости, а также условия, при которых необходимо проводить эксперименты.

Стандарт P.830 [2] имеет более узкую направленность и описывает условия и способы оценки качества голоса после преобразования с помощью кодеков. Также приводятся зависимости MOS от битовой скорости кодеков (рис. 1), соотношения сигнал/шум (рис. 2), уровня громкости и др.

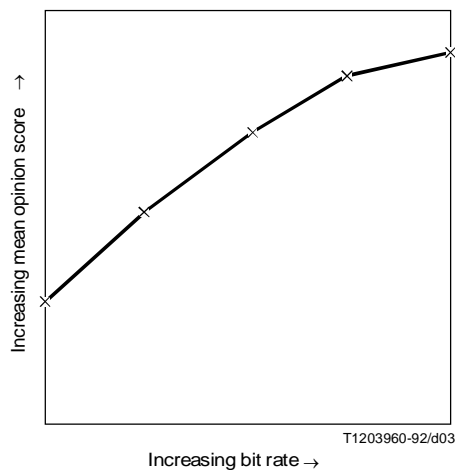


Рис.1. Зависимость MOS от битовой скорости

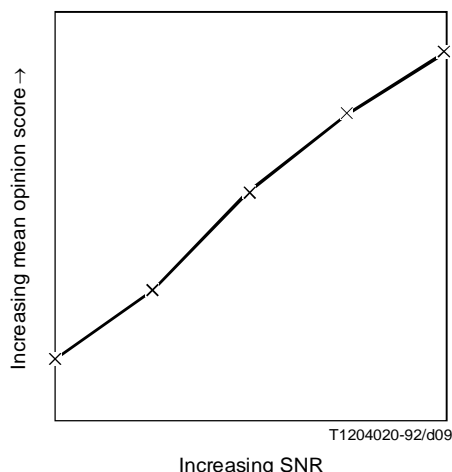


Рис. 2. Зависимость MOS от соотношения сигнал/шум (SNR)

Как можно видеть, на рисунках не указаны конкретные значения, а приведен лишь общий вид зависимости. Основной особенностью стандартов P.800 и P.830 является то, что они не привязывают способы оценки, которые в них рассматриваются, к

каким-нибудь конкретным реализациям протоколов или кодеков, а лишь дают описание того, каким образом следует оценивать их работу.

Каким бы хорошим ни был метод субъективной оценки, у него есть недостатки, связанные со сложностью его применения. Таким образом, был разработан способ объективной оценки качества работы кодеков PSQM (Perceptual Speech Quality Measure), изложенный в стандарте P.861 [3]. Принцип работы алгоритма заключается в сравнении закодированного и исходного сигналов (рис. 3).

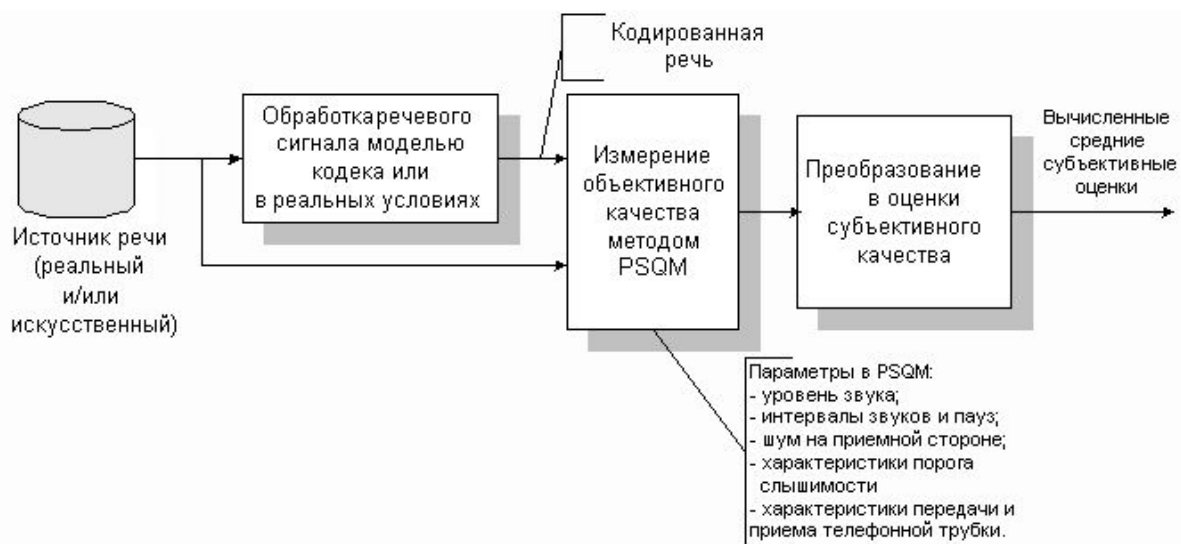


Рис. 3. Схема измерения объективного качества речи по методу PSQM

Значения, получаемые в результате измерений, являются критерием сравнительного ухудшения качества DCR (Degradation Category Rating), который, также как и MOS, описывается в стандарте P.800 и называется PSQM-оценкой. Однако максимальное значение этого показателя равно 6.5 и не имеет однозначного соответствия со шкалами DCR или MOS, так как для различных языков и условий эксперимента оно может отличаться.

С точки зрения практической ценности может быть интересным стандарт P.11 [4], в котором рассматривается влияние различных физических процессов, а также характеристик канала на итоговую оценку, выраженную в виде MOS. В нем предлагается другая оценка, называемая «индексом качества передачи» (transmission quality index) и вычисляемая по формуле

$$I = I(S/N) I(BW) I(ST),$$

где $I(S/N)$ – индекс снижения уровня сигнала и линейного шума; $I(BW)$ – индекс для полосы пропускания; $I(ST)$ – индекс для боковых частот. Критерии I и MOS связаны достаточно сложной экспоненциальной зависимостью.

Похожий подход для вычисления показателя качества используется в стандарте G.113 [6] и называется «рассчитываемым планируемым параметром ухудшения» CPIF (Calculated Planning Impairment Factor). В отличие от индекса качества передачи для его расчета используется простая формула:

$$I_{cpif} = I_o + I_q + I_{dte} + I_{dd} + I_e - A,$$

где I_o – параметр ухудшения качества, обусловленный неоптимальным уровнем громкости и/или высоким шумом в канале; I_q – параметр ухудшения качества, обусловленный шумами квантования в ИКМ; I_{dte} – параметр ухудшения качества, обусловленный акустическим эхом; I_{dd} – параметр ухудшения качества, обусловленный передачей речи на большое расстояние (задержка); I_e – параметр ухудшения качества, обусловлен-

ный специальными устройствами, в частности низкоскоростными кодеками; A – параметр ожидания, вытекающий из предполагаемого пользователем уровня качества.

Для определения значения параметров приводятся таблицы, в которых конкретным физическим величинам, кодекам или явлениям ставится в соответствие определенное число. Оно и подставляется в формулу. В результате получается показатель, учитывающий основные факторы, которые влияют на качество передачи аудио информации. Для применения в IP-телефонии следует добавить еще один параметр, отражающий уровень потери пакетов.

При необходимости оценки влияния задержек на качество передачи голоса можно воспользоваться стандартом G.114 [7]. В нем приводится следующая зависимость области применения и односторонней задержки:

- 0–150 ms – приемлемо;
- 150–400 ms – приемлемо при наличии участков спутниковой связи;
- более 400 ms – неприемлемо.

В стандарте приводятся требования к времени распространения сигналов по каналам национального и международного масштаба, а также задержки, вносимые кодеками, коммутаторами, различного рода магистральными сетями. Рассматриваются зависимости, отражающие влияние времени распространения сигнала на оценку MOS.

Недостатки существующих методов оценки качества передачи голоса

Проведенный обзор показал, что вопросы оценки качества передачи голоса рассмотрены в стандартах ITU достаточно подробно. Кроме того, эта проблема исследуется организациями ESTI, EURESCOM, IETF, MMCF [8]. Однако у всех рекомендаций объективного характера есть один недостаток, связанный с тем, что они предназначены для определения качества передачи голоса в конкретном направлении при наличии полной информации о состоянии узлов коммутации и магистральных каналов. Исходя из сказанного, представляется невозможным оценить пригодность проектируемой сети, если она состоит из участков, информация о которых скрыта от оператора. Ситуация еще более усложнится, если эти участки будут отвечать за терминацию звонков по многим направлениям. Таким образом, единственно доступным остается субъективный метод. Но в рассматриваемом случае его использование практически невозможно, так как очень сложно провести прослушивание и оценку всех возможных направлений. Следует учитывать, что любые изменения в сокрытой сети могут сильно повлиять на полученные представления.

Описание метода статистического анализа состояний вызовов

Из вышесказанного следует, что необходима разработка метода, позволяющего оценивать качество передачи голоса в определенных направлениях с минимальными затратами ресурсов. Решение указанной проблемы может состоять в том, чтобы анализировать статистические данные о состоянии вызовов, проходящих через сеть другого оператора. При этом желательно использовать максимально возможное количество параметров, значения которых свидетельствуют о качестве соединения. Кроме того, использование предлагаемого подхода позволит выявить достоинства и недостатки различных способов организации сети.

В качестве простейшего примера применения подобного метода можно привести эксперимент, выявляющий степень влияния элементов дифференцированного обслуживания трафика на успешность установления соединений в сети IP-телефонии. Условия проведения испытания заключались в том, что несколько человек в течение дня для осуществления городских звонков пользовались исключительно услугами IP-телефонии, причем все звонки маршрутизировались через сеть одного и того же опера-

тора в течение всего времени эксперимента, составляющего восемь дней. Особенность заключается в передаче голосовой информации по тому же каналу, что и данные, пропускная способность которого ограничена 256 кбит/с. Схема подключения изображена на рис. 4.

Изначально голосовая информация и трафик данных не имеют преимуществ друг перед другом. Однако в середине эксперимента пакеты, идущие с адреса голосового шлюза, получают более высокий приоритет. Вся информация о звонках накапливается в привратнике (сервере, управляющем зоной сети IP-телефонии) для дальнейшего анализа. В результате была определена доля успешных соединений за каждый день и построена зависимость на все время измерений (рис. 5).

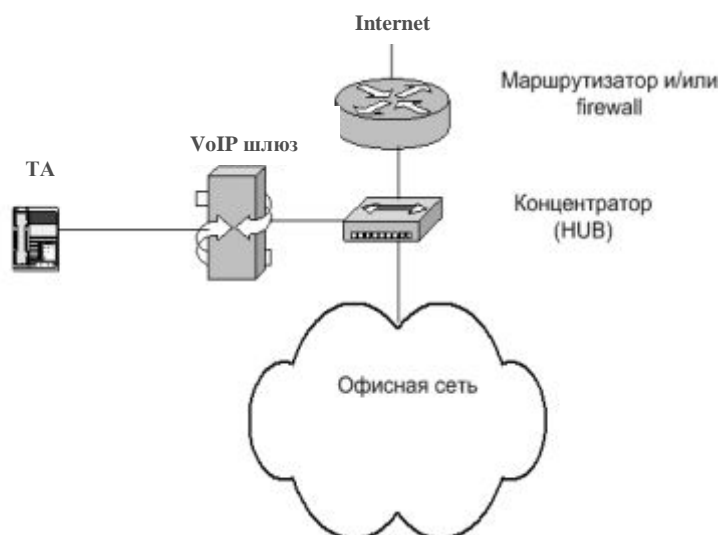


Рис. 4. Схема подключения пользователя IP-телефонии к Internet

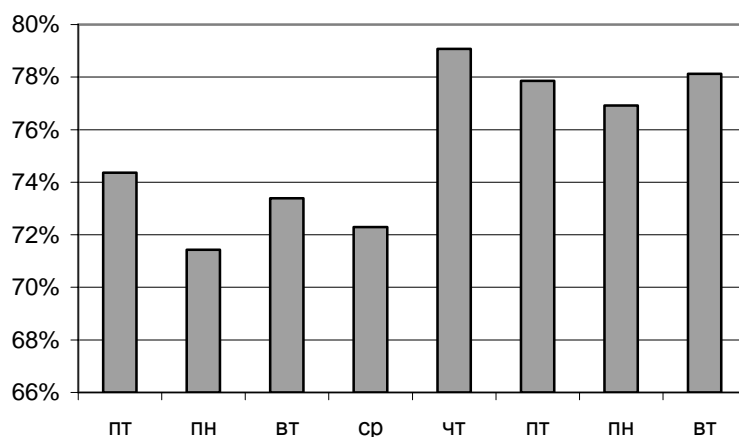


Рис. 5. Отношения успешных и неуспешных попыток соединений за 8 дней

Из представленного рисунка видно, что отношение успешных и неуспешных попыток соединения увеличилось в среднем на 5,1%. Причем интересно то, что зависимость по одним и тем же дням неделям сохраняется.

Заключение

Анализ причин выявленной в результате эксперимента зависимости не является целью настоящей работы, однако приведенный пример показывает, как, используя све-

дения о состоянии вызовов в сети, можно оценить качество ее функционирования. Таким образом, можно сделать вывод о перспективности разработки метода, позволяющего в автоматическом режиме, основываясь на особенностях функционирования VoIP протоколов, работа которых, в свою очередь, отражается в статистических зависимостях состояний вызовов, вносить изменения в алгоритм работы сети.

Литература

1. ITU-T, «Recommendation P.800, Methods for subjective determination of transmission quality». Женева, ITU, 1996.
2. ITU-T, «Recommendation P.830, Subjective performance assessment of telephone-band and wideband digital codecs». Хельсинки, ITU, 1996.
3. ITU-T, «Recommendation P.861, Objective quality measurement of telephone-band (300 – 3400 Hz) speech codecs». Женева, ITU, 1996.
4. ITU-T, «Recommendation P.11, Effect of transmission impairments». Хельсинки, ITU, 1993.
5. ITU-T, «Recommendation G.113, Transmission impairments». Хельсинки, ITU, 1996.
6. ITU-T, «Recommendation G.114, One-way transmission time». Хельсинки, ITU, 1996.
7. Росляков А.В., Самсонов М.Ю. Модели и методы оценки качества услуг IP-телефонии / axenet.ru

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ КОДИРОВАНИЯ РЕЧИ

М.Б. Шалаева

Научный руководитель – д.т.н., профессор А.Ю. Тропченко

Цель исследования заключается в выборе общих для большинства алгоритмов кодирования речи функциональных блоков, их анализе и предложении решений по оптимизации вычислений. В настоящей статье рассматриваются блоки прямого и обратного преобразований между временной и частотной областями представления сигналов.

Введение

В настоящее время большую популярность приобрела компьютерная телефония – технология, основанная на передаче голосовых сообщений через сети, изначально предназначенные для трафика данных. Ее достоинством является снижение затрат на междугородные и международные переговоры для многих компаний и частных лиц, недостатком – относительно низкое качество синтезированной речи по сравнению с традиционными телефонными сетями общего пользования (ТфОП).

Весь путь, проходимый голосовой информацией в сетях с пакетной коммутацией, в общем случае можно разложить на этапы, представленные на рис. 1.

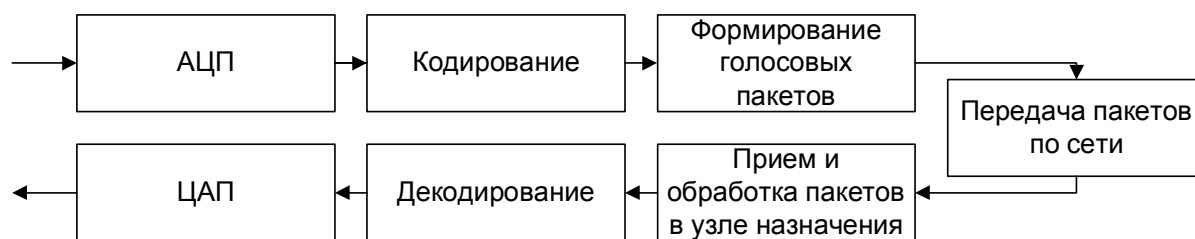


Рис. 1. Передача речи в сетях с пакетной коммутацией

Речевые пакеты формируют трафик реального времени. Факторы, негативно влияющие на качество переданной речи:

- задержки (время запаздывания) и их вариации (джиттер), вносимые:
 - операционной системой;
 - устройствами кодирования/декодирования;
 - сетью передачи данных;
 - буфером устранения джиттера и др.
- эхо;
- потери пакетов.

Таким образом, для обеспечения конкурентоспособности современных телекоммуникационных технологий необходимо совершенствование:

- сетевых политик;
- алгоритмов преобразования речи.

Основная часть

Рассмотрим последние. Учет основных требований к качеству речевого сегмента и временной задержке, как правило, требует компромисса.

С увеличением сложности алгоритма для уменьшения требуемой полосы пропускания увеличивается время обработки в устройствах кодирования/декодирования. Хотя качество сигнала в основном напрямую связано со скоростью цифрового потока, сложные алгоритмы кодирования способны достичь более высоких отношений качества и скорости.

- Используемые технологии преобразования речи можно разделить на две группы:
- аппроксимация (кодирование) формы речевой волны;
 - параметрическое компандирование речи (вокодерные преобразования).

В последней технологии моделируется процесс речеобразования человека. Для этого в кодере из речевого сигнала вычисляются определенные параметры, передаваемые к декодеру, в котором они применяются для восстановления формы исходного сигнала. Использование исключительно параметрических методов приводит к потере натуральности звучания голоса и большой чувствительности к фоновым шумам.

Один из способов повышения эффективности использования полосы пропускания канала связи состоит в применении гибридных методов, основанных на принципах линейного предсказания и объединяющих параметрическое компандирование и кодирование формы волны.

Большинство гибридных кодеров используют замкнутое кодирование (метод «анализ через синтез») на передающей стороне, что позволяет подкорректировать определенные параметры посредством сравнения результата синтеза с оригиналом. Это, безусловно, увеличивает время обработки сигналов, но обеспечивает лучшие показатели при передаче.

На рис. 2 изображены сглаженные зависимости оценок общественного мнения MOS от требований к битовой скорости потока, построенные автором статьи по усредненным результатам исследований ITU Study Group 15 и данным Р. Кокса (IEEE Communications Magazine, сентябрь 1997 г.)

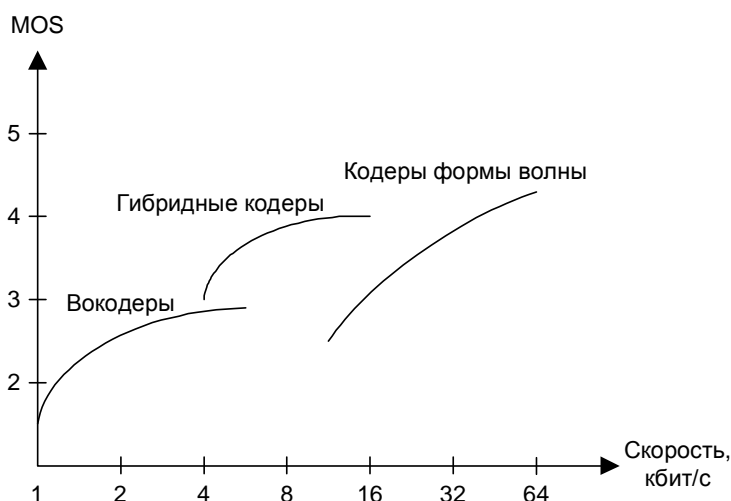


Рис. 2. Зависимость оценок MOS от скорости потока для кодеров формы волны, вокодеров и гибридных кодеров

На рис. 3 изображены сглаженные зависимости общих задержек алгоритмов от битовой скорости потока. Численные значения задержек взяты из описаний рекомендаций ITU-T, ETSI и др., размещенных на сайтах: www.axenet.ru, www.vocal.com.

В последних разработках кодеров широко применяются:

- алгоритмы долговременного и кратковременного предсказания;
- кодовые книги, хранящие различные виды сигналов возбуждения;
- переменная скорость кодирования на основе классификаций речевых сегментов и др.

При кодировании голосового сигнала используется его спектральное представление, которое позволяет:

- уменьшить объем передаваемых данных;
- осуществить фильтрацию – селекцию желаемой полосы частот в обрабатываемом сигнале:

- подавлять шумы обнулением компонент на нежелательных частотах;
 - выделить достаточную полосу частот для воспроизведения разборчивой речи и особенностей (тембра) говорящего, т.е. содержащую три первых формантных частоты (как правило, используется полоса частот от 300 до 3400 Гц);
- выполнить классификацию сегмента сигнала (вокализованный, невокализованный глухой, невокализованный фрикативный звук, шум) и др.

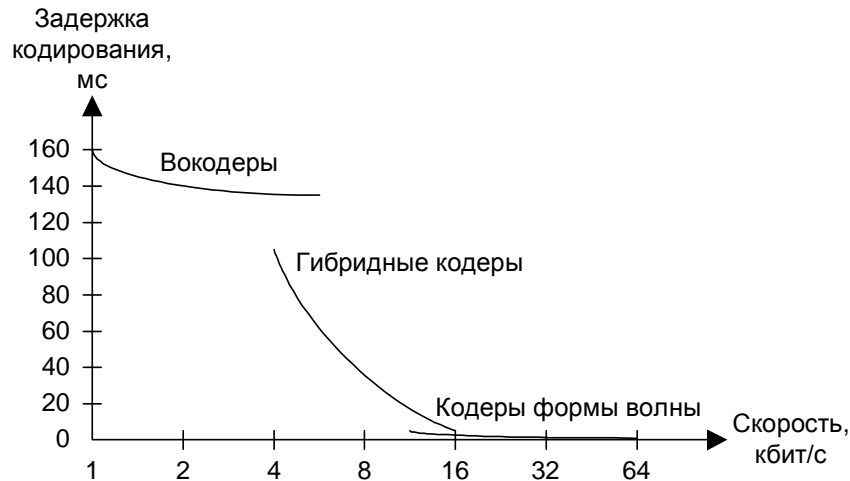


Рис. 3. Зависимость задержки кодирования от скорости потока для кодеров формы волны, вокодеров и гибридных кодеров

Следует отметить, что в некоторых случаях операцию фильтрации целесообразнее выполнять путем свертки исходных данных с фильтрующей последовательностью, т.е. без проведения анализа в частотной области.

Анализ стандартизованных и частных разработок показал, что в качестве основной операции для:

- вычисления спектра мощности сигнала и «родственных» задач, например, кепстрального анализа;
- оценивания передаточной функции и импульсного отклика;
- быстрого вычисления свертки и др.

используется дискретное преобразование Фурье (ДПФ):

$$F(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(nT) \times \left[\cos\left(\frac{2\pi}{N} kn\right) - j \sin\left(\frac{2\pi}{N} kn\right) \right] = \frac{1}{N} \sum_{n=0}^{N-1} x(nT) \times e^{-j\frac{2\pi}{N} kn}$$

(пояснение обозначений не приводится в виду их общеизвестности).

При решении задачи вычисления спектра ДПФ преобразует коэффициенты при косинусоиде и синусоиде в действительную и мнимую части $F(k)$, модуль же $F(k)$ указывает на присутствие во входной последовательности $x(nT)$ периодической составляющей с частотой $\omega = 2\pi k/N$.

Для восстановления сигнала по его спектру и в ряде других задач используется обратное ДПФ:

$$x(nT) = \sum_{k=0}^{N-1} F(k) \times \left[\cos\left(\frac{2\pi}{N} kn\right) + j \sin\left(\frac{2\pi}{N} kn\right) \right] = \sum_{k=0}^{N-1} F(k) \times e^{j\frac{2\pi}{N} kn}$$

Коэффициент N^{-1} указан в уравнении прямого преобразования для обеспечения равенства $F(0)$ среднему значению $x(nT)$. Это соответствует исходной посылке о том,

что нулевой коэффициент ряда Фурье равен постоянной составляющей периодического колебания.

Указанный коэффициент во многих источниках можно встретить только в обратном преобразовании. Расхождения в формулах вызваны тем, что математически результаты прямого и обратного преобразований различаются в N раз. В практике же вычислений наличие рассматриваемого коэффициента часто не имеет значения.

В силу приведенных аргументов в последующих формулах ДПФ коэффициент N^{-1} не указывается.

ДПФ характеризуется:

- необходимостью при вычислении спектра и восстановлении сигнала работать с комплексными числами;
- разложением входной последовательности сигнала по ортогональным базисам.

Первая особенность указывает на большую вычислительную сложность алгоритма – примерно N^2 операций (комплексных сложений и умножений), а вторая – на возможность ее уменьшения за счет использования таких свойств ядра преобразования, как:

- симметричность матрицы;
- цикличность значений элементов матрицы – косинуса и синуса;
- присутствие N различающихся по величине значений в матрице, содержащей N^2 элементов;
- мультипликативность элементов матрицы.

Таким образом, используются разновидности алгоритмов быстрого преобразования Фурье (БПФ), учитывающие указанные свойства. Число операций при выполнении БПФ близко к $N \log_2 N$.

Преобразование Фурье последовательности отсчетов сигнала в общем случае определяется на бесконечном интервале времени,

$$F(k) = \sum_{n=-\infty}^{\infty} x(nT) \times e^{-j\frac{2\pi}{N}kn},$$

и не имеет смысла для нестационарных сигналов, например, речевых, в силу изменчивости спектра речи во времени. Поэтому для характеристики энергии речевого сигнала используется кратковременное преобразование Фурье, реализуемое введением в уравнение весовой функции («временного окна») вида $h(nT)$:

$$F(k, nT) = \sum_{r=-\infty}^n x(rT) \times h(nT - rT) \times e^{-j\frac{2\pi}{N}kn}.$$

Используя свертку, предыдущее равенство можно записать как

$$F(k, nT) = \left[x(nT) \times e^{-j\frac{2\pi}{N}kn} \right] \times h(nT).$$

Так как ряд Фурье предназначен для периодических сигналов, вводится предположение, что после последнего отсчета входной последовательности следует точное ее повторение. Следовательно, порождаются не имеющие никакого отношения к действительности разрывы и изломы. Они искажают спектр сигнала, приводя к появлению боковых лепестков значительного уровня в спектральной области.

Для ослабления этого эффекта сигнал перед выполнением ДПФ умножают на уменьшающуюся от центра к краям весовую функцию. В результате становится менее значительной величина «скачков» на стыках сегментов и, как следствие, уровень нежелательных боковых лепестков, но при этом происходит некоторое расширение спектральных пиков.

В настоящее время существует большое разнообразие весовых функций. Распространение одних обусловлено вычислительной простотой, других – возможностью получения лучших результатов. Распространенные типы «окон»:

- экспоненциальное окно

$$w_y(n) = a^n, \quad 0 \leq n \leq N-1;$$

$$w_y(n) = 0, \quad 0 < n, n > N-1;$$

- прямоугольное окно

$$w_n(n) = 1, \quad 0 \leq n \leq N-1;$$

$$w_n(n) = 0, \quad 0 < n, n > N-1;$$

- окно Хемминга

$$w_x(n) = 0,54 - 0,46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

$$w_x(n) = 0, \quad 0 < n, n > N-1.$$

«Окно» применяется для выделения сегмента речевого сигнала в окрестности некоторого момента времени с целью его последующей обработки:

- вычисления функции кратковременной энергии;
- среднего числа переходов через нуль;
- автокорреляционной функции.

Таким образом, ввиду возможности адаптации практически к любым входным последовательностям (неудобными для описания являются функции с разрывами и резкими изломами) преобразование Фурье нашло широкое применение в медицине, экономике, вычислительной технике и других областях знаний. Основными его недостатками являются:

- несимметричность прямого и обратного преобразований;
- относительно большая вычислительная сложность – N^2 операций над комплексными числами, как было отмечено ранее.

В настоящее время известны и другие типы преобразований, которые часто считаются модификациями или упрощениями ДПФ. К ним относятся:

- дискретное преобразование Хартли (ДПХ), в котором в качестве базиса используются не косинусы и синусы, а их суммы;
- дискретное косинусное преобразование, в котором используются только косинусные составляющие. Позволяет представлять исключительно четные сигналы, т.е. $f(-x) = f(x)$, но решает проблему, связанную с разрывами;
- дискретное синусное преобразование, в котором используются только синусные составляющие. Позволяет представлять исключительно нечетные сигналы, т.е. $-f(-x) = f(x)$;
- преобразование Уолша-Адамара, которое использует синусоиды, квантованные по уровню только двумя значениями: ± 1 . Основным достоинством подобного преобразования является низкая вычислительная сложность относительно других преобразований;
- прочие.

ДПХ аналогично ДПФ, но имеет симметричные прямое и обратное преобразования и позволяет избавиться от операций над комплексными числами, т.е. представляет собой наиболее удобный способ вычислений. Для получения ДПФ необходимо переводить действительную функцию в комплексную область, а затем вновь в действительную, на что используется дополнительное машинное время и память. Использование ДПХ позволяет избежать этих переходов, поскольку оно определено в действительной области. ДПХ по сравнению с ДПФ не получило широкого распространения. В результате обзора областей применения ДПХ выявлено, что оно нашло свою нишу в компьютерной томографии.

Прямое и обратное преобразования Хартли имеют вид:

$$H(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(nT) \times \left[\cos\left(\frac{2\pi}{N} kn\right) + \sin\left(\frac{2\pi}{N} kn\right) \right] = \frac{1}{N} \sum_{n=0}^{N-1} x(nT) \times \text{cas}\left(\frac{2\pi}{N} kn\right);$$

$$x(nT) = \sum_{k=0}^{N-1} H(k) \times \text{cas}\left(\frac{2\pi}{N} kn\right).$$

Основные свойства ядра преобразования Хартли:

- симметричность матрицы;
- цикличность значений элементов матрицы;
- отсутствие мультипликативности элементов матрицы.

Легко показать, что преобразования Фурье и Хартли связаны друг с другом следующими соотношениями:

$$H(k) = \text{Re}[F(k)] - \text{Im}[F(k)],$$

$$F(k) = \frac{H(k) + H(-k)}{2} + j \frac{H(k) - H(-k)}{2}.$$

Таким образом, фактически два рассматриваемых преобразования равноценны.

Для организации быстрых вычислений используются рекуррентные соотношения.

Примеры рекуррентных соотношений для ДПФ и ДПХ, соответственно:

$$F(k) = F_{\text{четн}}(k) + F_{\text{неч}}(k) \times e^{-j\frac{2\pi}{N}k}$$

(последовательное применение этого метода может свести преобразование к операциям типа «бабочка»);

$$H(k) = H_{\text{четн}}(k) + H_{\text{неч}}(k) \times \cos\left(\frac{2\pi}{N}k\right) + H_{\text{неч}}(N-k) \times \sin\left(\frac{2\pi}{N}k\right).$$

Хотя для ДПХ рекуррентная формула выглядит сложнее, выигрыш при вычислениях очевиден – при ДПФ используются комплексные числа, т.е. операция сложения состоит из двух операций, а умножения – из шести, итого – восемь операций. Преобразование Хартли содержит пять операций и, кроме того, позволяет упростить реализацию и сократить объем занимаемой памяти.

Заключение

В работе показано, что существует возможность замены широко используемого ДПФ на менее известное, но более эффективное ДПХ со следующими достоинствами:

- преобразование симметрично – прямое и обратное преобразования находятся одинаково;
- преобразование действительно, что уменьшает объем требуемой памяти и время преобразования;
- возможность использования быстрых алгоритмов преобразования.

В настоящей работе предложена модификация блоков прямого и обратного преобразований между временной и частотной областями представления сигналов посредством использования прямого и обратного ДПХ вместо ДПФ.

Литература

1. Быков С.Ф., Журавлев В.И., Шалимов И.А. Цифровая телефония / Уч. пос. для вузов. М.: Радио и связь, 2003.
2. Вегешна Ш. Качество обслуживания в IP-сетях. М.–СПб– Киев: Вильямс, 2003.
3. Гольдштейн Б.С., Пинчук А.В., Суховицкий А.Л., IP-телефония. М.: Радио и связь, 2001.
4. Дрю Х., Внутренний мир Microsoft TCP/IP / Пер. с англ. Киев: DiaSoft, 2000.
5. Рабинер Л.Р., Голд Б., Теория и применение цифровой обработки сигналов / Пер. с англ. М.: Мир, 1978.
6. Рабинер Л.Р., Шафер Р.В., Цифровая обработка речевых сигналов / Пер. с англ. М.: Радио и связь, 1981.

ВЕРИФИКАЦИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА ПРОГРАММЫ С ИСПОЛЬЗОВАНИЕМ КОМПЛЕКСНЫХ КУБИЧЕСКИХ ПОКРЫТИЙ

А.В. Сидоров

Научный руководитель – д.т.н., профессор О.Ф. Немолочнов

В работе предлагается метод верификации программ и вычислительных процессов на основе комплексных кубических покрытий.

Введение

Верификация – это установление соответствия между различными объектами. В применении к программному обеспечению наиболее часто возникают задачи установления соответствия между двумя программами, либо между программой и спецификацией [1].

Задача сравнения программ между собой возникает в случае проведения модернизации программы, когда необходимо выполнить проверку двух версий программы и выяснить, не было ли при модернизации внесено в код дополнительных, нежелательных ошибок. Также задача верификации программ возникает в учебных системах, когда необходимо установить качество выполнения учебного задания.

В случае верификации программы и спецификации необходимо устанавливать соответствие между моделями разного уровня, что в общем виде представляет собой сложную и неоправданно трудоемкую задачу. Спецификация заведомо лишена подробностей реализации, присущих программе. В этом случае имеет смысл верификация, направленная на выявление несоответствий (ошибок программы) определенного вида, например, несоответствия в потоках управления.

В задачах верификации программного обеспечения важную роль играют методы статического анализа, т.е. анализа кода программы без выполнения его на вычислительной машине. Методы статического анализа активно используются для построения наборов входных данных для тестирования, а также для выявления некоторых классов ошибок в программах.

В данной работе предлагается метод полной или частичной верификации двух программ между собой, основанный на анализе комплексных кубических покрытий логических условий. Кроме того, предлагается идея использования контрольных точек, позволяющая применять метод для верификации моделей разного уровня, а именно программы и спецификации.

Графо-аналитическая модель программы

Программа, реализующая некоторый вычислительный процесс, может быть представлена в виде графа как множества вершин и связей между ними. Каждая команда есть вершина графа, а дуги графа соответствуют передаче управления между командами. В таком графе дуги образуются либо линейно в порядке следования операторов, либо с помощью скачков условной и безусловной передачи управления, нарушающей естественный линейный ход выполнения программы. На графе выделяются начальная и конечная вершина, содержащие, соответственно, точки входа и выхода из программы.

В некоторых случаях удобно сократить размерность неструктурированного графа программы по числу вершин и дуг, объединив некоторые вершины в замкнутые множества (множества, имеющие одну точку входа и одну точку выхода) и в результате укрупнив анализируемые объекты до уровня структурного элемента программы. Например, удобно выделить циклы в отдельные структуры и проанализировать изменение переменных в теле цикла отдельно от основной программы, а в дальнейшем рассматривать эти структуры как единичные вершины графа основной программы, не порождающие дополнительных ветвей выполнения.

Наиболее формализованной моделью является модель в виде бинарного или булева графа, в котором дуги графа задают только связи между вершинами и, следовательно, не яв-

ляются нагруженными. Эта модель позволяет в дальнейшем в полной мере воспользоваться математическим аппаратом для анализа программы. Большинство современных вычислительных машин имеют одноадресную систему команд, и, следовательно, граф вычислительного процесса такой машины автоматически является бинарным. В случае, когда граф не является бинарным, возможно его преобразование к бинарной форме.

Комплексные кубические покрытия логических условий

Рассмотрим булев граф программы $BG(P)$, состоящий из линейных и условных вершин и связей между ними. Линейная вершина содержит одну точку входа и одну точку выхода, это может быть либо одна линейная команда (т.е. команда, не являющаяся командой передачи управления), либо последовательность линейных команд. Условная вершина имеет одну точку входа и две точки выхода, задающие адреса ветвления (две ветви) в зависимости от выполнения или невыполнения условия, задаваемого в вершине. В общем случае условная вершина может содержать не только одну команду условного перехода, но и несколько команд, формирующих само условие. Процедуру верификации можно применять не только к графу $BG(P)$, но и к подграфам графа $BG(P)$, содержащим один вход и один выход.

Рассмотрим всевозможные пути выполнения программы, которые будут соответствовать путям на графе $BG(P)$, соединяющим начальную и конечную вершину. Каждый путь l на графе $BG(P)$ состоит из последовательности условных и линейных вершин. Условные вершины определяют потоки управления или логику программы, а линейные вершины определяют вычисление переменных или выполнение других линейных операций.

Условия, которые определяют путь l , можно закодировать булевыми переменными, принимающими значение 1 или 0 в зависимости от выполнения или невыполнения условия. Для логических условий, заданных булевыми переменными, возможно построить кубическое покрытие $C(P) = \{c_1, c_2, \dots, c_n\}$, где каждый куб c_i содержит в себе условия, определяющие путь l_i на графе $BG(P)$. Координаты куба c_i могут принимать значения 0, 1 и x (значение x соответствует условию, не влияющему на формирование данного пути). Покрытие $C(P)$ полностью соответствует определению покрытия в исчислении кубических комплексов, поэтому к покрытиям $C(P)$ и кубам c_i применимы алгебро-топологические операции: пересечение (\cap), звездчатое произведение ($*$) и вычитание ($\#$) [4]. Эти операции позволяют устанавливать отношения включения, поглощения и эквивалентности между различными покрытиями.

Покрытие $C(P)$ содержит в себе логическую часть программы, т.е. наборы логических условий, определяющих пути выполнения программы. Дополним покрытие $C(P)$ множеством, определяющим результат работы программы для каждого пути. Для этого каждому кубу c_i из покрытия $C(P)$ ставится в соответствие множество r_i , содержащее результат выполнения программы. В самом широком определении множество r_i содержит линейную последовательность операторов, через которые проходит путь l_i . Однако такое определение результата работы программы несет в себе слишком много подробностей, что затрудняет дальнейший анализ и сравнение результатов. В более узком определении результата r_i это может быть формула или набор формул, определяющих выходную переменную функции или глобальную переменную программы. Покрытие $C(P)$, дополненное множеством результатов работы программы, будем называть комплексным кубическим покрытием $\tilde{C}(P)$.

Построение комплексных кубических покрытий

Рассмотрим фрагмент кода $F(P)$ программы P , подлежащий верификации. Это может быть вся программа целиком, отдельная процедура или функция, либо любой другой участок кода, имеющий одну точку входа и одну точку выхода.

Выделим в $F(P)$ входные переменные, составив список их перечисления $iVL(F)$. Если $F(P)$ – это функция (или процедура), то список $iVL(F)$ может совпадать со списком входных параметров функции при условии отсутствия использования в теле функции глобальных переменных.

Кроме того, следует определить, что считается результатом работы программы. Как уже было сказано выше, в качестве переменной (или набора переменных), хранящей результат работы, можно выбрать возвращаемое значение функции.

Например, для функции M_1 (см. рис. 1), вычисляющей максимум трех чисел x_1, x_2, x_3 , список $iVL(M_1)$ равен $\langle x_1, x_2, x_3 \rangle$, а в качестве результата работы следует выбрать значение переменной r .

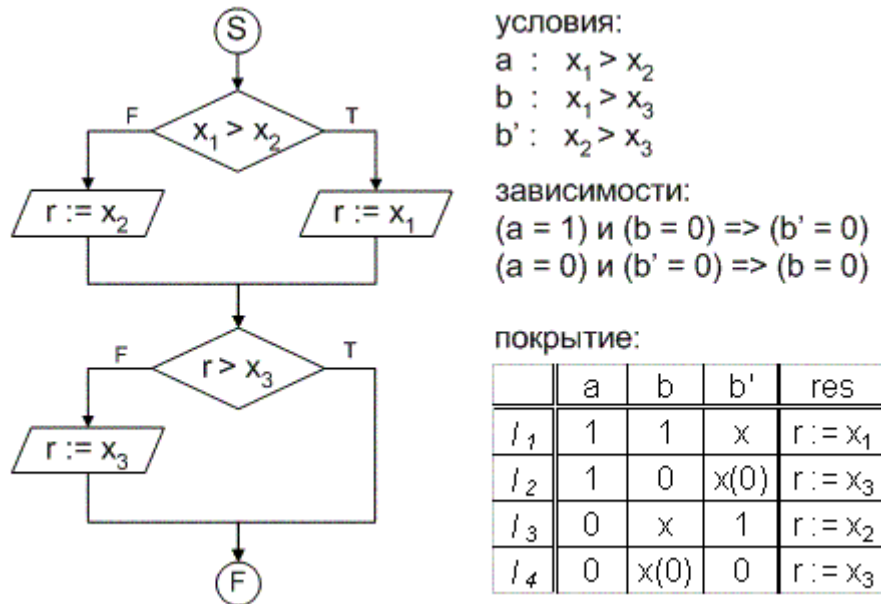


Рис. 1. Пример функции M_1 , вычисляющей максимум трех чисел x_1, x_2, x_3

При верификации программы и спецификации выбор результата работы должен в первую очередь определяться возможностями спецификации как модели меньшего уровня подробности. Другими словами, спецификация описывает лишь некоторые свойства программы, исходя из которых и следует делать выбор результата. Например, спецификация может устанавливать правила вычисления одной из переменных, значение которой в данном случае следует принять за результат работы.

Если спецификация описывает потоки управления программы, то в качестве результата работы программы можно принять последовательность прохождения контрольных точек программы; в качестве контрольных точек можно принять вызовы ключевых функций или заранее внесенные в код отметки в виде комментариев.

Обязательное условие, накладываемое на список входных переменных $iVL(F)$ – множество этих переменных должно однозначно определять все условия ветвления и результат выполнения. В указанном выше случае выбора в качестве списка $iVL(F)$ входных параметров функции это условие выполнено автоматически.

После того, как выделен список входных параметров $iVL(F)$ и определены правила представления результата, производится процесс построения и активизации путей на графе программы.

Для каждого пути, соединяющего начальную и конечную вершину графа фрагмента $F(P)$, выделяем множество условий, определяющих этот путь. Так как путь можно представить в виде линейной последовательности команд, то, двигаясь по этой последовательности операторов в обратном направлении от конкретного условия a к началу участка $F(P)$, можно преобразовать каждое условие, представив его в виде выражения, зависящего только от входных переменных. Будем называть этот процесс активизацией условия. Например, путь l_2 графа функции M_1 определяет следующую линейную последовательность команд и условий $\{(x_1 > x_2) = T; r := x_1; (r > x_3) = F; r := x_3\}$, таким образом, активизация условий этого пути даст два условия $a : x_1 > x_2$ и $b : x_1 > x_3$.

Один и тот же условный оператор в программе может фактически породить несколько условий, так как само условие, записанное в операторе, в процессе активизации может быть интерпретировано разными выражениями (не эквивалентными между собой) для разных путей, проходящих через этот условный оператор. Например, условие $r > x_3$ функции M_1 для путей l_1 и l_2 , проходящих через вершину $r := x_1$, будет выражено в виде условия $b: x_1 > x_3$, а для путей l_3 и l_4 , проходящих через вершину $r := x_2$ – в виде условия $b': x_2 > x_3$.

Одной из важных проблем, возникающих в процессе верификации, являются программные циклы. Переменные, изменяющиеся в теле цикла, в результате можно представить в виде рекуррентных выражений. Имеет смысл заранее выделить циклические параллельные структуры и, с помощью итерационно-рекурсивных покрытий, построить рекуррентные зависимости каждой переменной, изменяющейся в теле цикла. Таким образом, цикл можно в дальнейшем рассматривать как одиночную вершину, в которой осуществляется итеративное преобразование значений переменных. Подробное рассмотрение циклов выходит за рамки данной работы.

Если наложить на входные переменные условие неизменности значений в области $F(P)$, то процесс активизации условия будет оканчиваться в момент, когда из формулы для условия будут исключены все переменные, кроме входных. В идеальном случае, если все условия в теле верифицируемой функции независимы, записаны через входные переменные, которые не изменяются в теле функции, то активизация условий не требует преобразования выражений (см., например, функцию M_2 на рис. 2). Хорошим примером таких функций являются функции вычисления интервальных формул.

После того, как активизация всех путей выполнена, необходимо провести анализ выражений, определяющих условия. Некоторые условия могут оказаться эквивалентными или инверсными, их следует объединить, проведя преобразование покрытия $\tilde{C}(P)$. Условия, определяющие некоторые пути, могут оказаться несовместны. Такие пути следует исключить из покрытия $\tilde{C}(P)$. Кроме того, при построении покрытия необходимо учитывать, что некоторые группы условий могут оказаться зависимыми или частично-зависимыми. Например, условия a , b и b' покрытия функции M_1 являются частично-зависимыми (рис. 1), поэтому в покрытии для пути l_2 условие b' установлено в 0, несмотря на то, что оно не является определяющим условием пути l_2 . Рассмотрение методов определения зависимостей между условиями выходит за рамки данной работы, однако это также является важной темой для исследований.

Анализ комплексных кубических покрытий

Пусть имеются два фрагмента кода $F_1(P_1)$ и $F_2(P_2)$, подлежащих верификации. Рассмотрим комплексные кубические покрытия этих фрагментов \tilde{C}_1 и \tilde{C}_2 . Покрытия \tilde{C}_1 и \tilde{C}_2 должны быть построены с учетом входных параметров, между которыми может быть установлено взаимно однозначное соответствие, а также должны быть определены единые правила определения результата работы.

Т.к. все условия, входящие в покрытия \tilde{C}_1 и \tilde{C}_2 , выражены через входные переменные, между которыми установлено отношение эквивалентности, то возможно установить соответствие между условиями покрытий \tilde{C}_1 и \tilde{C}_2 . При этом, аналогично анализу условий внутри покрытия, необходимо провести анализ на зависимость условий покрытий \tilde{C}_1 и \tilde{C}_2 между собой. Само по себе установление соответствия и зависимостей между условиями может выявить расхождения в работе фрагментов $F_1(P_1)$ и $F_2(P_2)$.

После того, как установлены зависимости между условиями покрытий \tilde{C}_1 и \tilde{C}_2 , возможно провести непосредственное сравнение логической части покрытий. Для этого исполь-

зуются операции вычитания (#) и пересечения (\cap) кубических покрытий, рассмотренные подробно в [5].

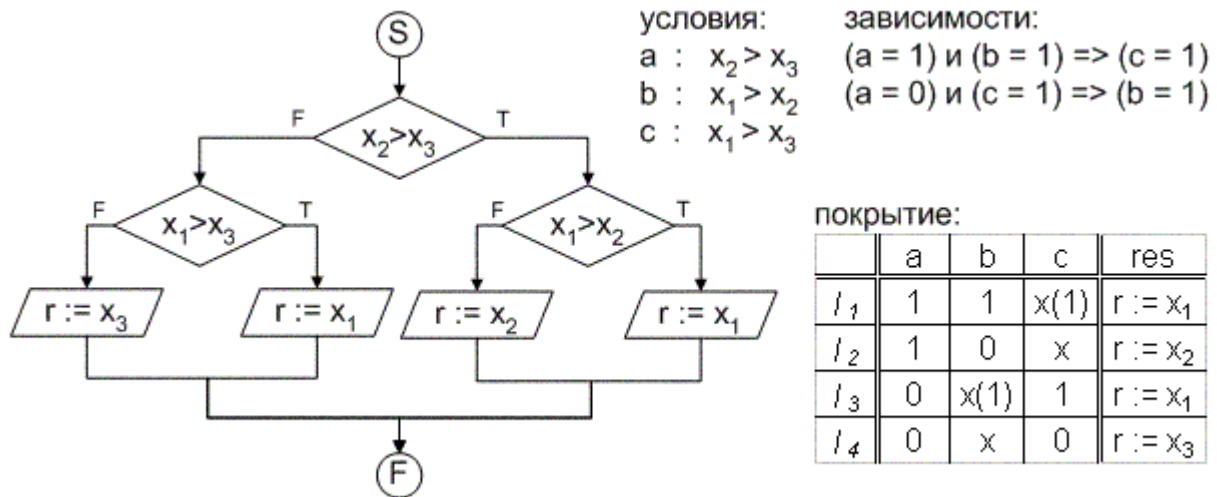


Рис. 2. Пример функции M_2 , верифицируемой с функцией M_1

Для каждого возможного результата $r_1^i \in \tilde{C}_1$ рассмотрим логическую часть покрытия $\tilde{C}_1[r_1^i]$, которая имеет в правой части результат r_1^i , а также логическую часть покрытия $\tilde{C}_2[r_2^i]$, которая имеет в правой части результат r_2^i , соответствующий результату r_1^i .

Возможны следующие ситуации:

1. $\tilde{C}_1[r_1^i] \# \tilde{C}_2[r_2^i] = \emptyset, \tilde{C}_2[r_2^i] \# \tilde{C}_1[r_1^i] = \emptyset$. Логические условия фрагментов $F_1(P_1)$ и $F_2(P_2)$ по вычислению результата r_1^i и r_2^i эквивалентны;
2. $\tilde{C}_1[r_1^i] \# \tilde{C}_2[r_2^i] = \emptyset, \tilde{C}_2[r_2^i] \# \tilde{C}_1[r_1^i] \neq \emptyset$. Фрагмент $F_2(P_2)$ имеет дополнительную логику по вычислению результата r_2^i ;
3. $\tilde{C}_1[r_1^i] \# \tilde{C}_2[r_2^i] \neq \emptyset, \tilde{C}_2[r_2^i] \# \tilde{C}_1[r_1^i] = \emptyset$. Фрагмент $F_1(P_1)$ имеет дополнительную логику по вычислению результата r_1^i ;
4. $\tilde{C}_1[r_1^i] \# \tilde{C}_2[r_2^i] \neq \emptyset, \tilde{C}_2[r_2^i] \# \tilde{C}_1[r_1^i] \neq \emptyset, \tilde{C}_2[r_2^i] \cap \tilde{C}_1[r_1^i] \neq \emptyset$. Логические условия фрагментов $F_1(P_1)$ и $F_2(P_2)$ по вычислению результата r_1^i и r_2^i имеет общую часть, а также различающиеся части;
5. $\tilde{C}_2[r_2^i] \cap \tilde{C}_1[r_1^i] = \emptyset$. Логические условия фрагментов $F_1(P_1)$ и $F_2(P_2)$ по вычислению результата r_1^i и r_2^i полностью различны.

Таким образом, перебор всех возможных $r_1^i \in \tilde{C}_1$ и $r_2^i \in \tilde{C}_2$, позволит установить соответствие между покрытиями \tilde{C}_1 и \tilde{C}_2 и, соответственно, между рассматриваемыми фрагментами программ $F_1(P_1)$ и $F_2(P_2)$.

При верификации рассматриваемых в качестве примеров функций M_1 и M_2 соответствия условий можно выразить соотношениями $a_1 = b_2, b_1 = c_2, b'_1 = a_2$. Проведя соответствующую перестановку столбцов и выполнив операцию вычитания покрытий, устанавливаем, что $\tilde{C}_1 \# \tilde{C}_2 = \emptyset, \tilde{C}_2 \# \tilde{C}_1 = \emptyset$, т.е. функции M_1 и M_2 эквивалентны.

Заключение

В данной работе предложен метод верификации вычислительных процессов программ на основе комплексных кубических покрытий. Разработка эффективных методов и моделей

статического анализа программ является актуальной проблемой в современных условиях ужесточения требований к качеству программных изделий.

Литература

1. Липаев В.В. Обеспечение качества программных средств. Методы и стандарты. М.: СИНТЕГ, Серия «Информационные технологии», 2001.
2. Зыков А.Г., Немолочнов О.Ф., Поляков В.И. Верификация в исследовательских, учебных и промышленных системах // Научно-технический вестник СПб ГИТМО(ТУ). Выпуск 11. Актуальные проблемы анализа и синтеза сложных технических систем / Гл. ред. В.Н. Васильев. СПб.: СПб ГУ ИТМО , 2003. С 146–151.
3. Липаев В.В. Функциональная безопасность программных средств. М., СИНТЕГ, Серия «Управление качеством», 2004.
4. Немолочнов О.Ф., Зыков А.Г., Поляков В.И. Кубические покрытия логических условий вычислительных процессов и программ. // Научно-технический вестник. СПб ГУ ИТМО, 2004.
5. Немолочнов О.Ф. Методы технической диагностики. Методические указания к курсовой работе. Л.: ЛИТМО, 1976.

ОЦЕНКА ФУНКЦИОНИРОВАНИЯ REAL-TIME OLAP СИСТЕМ**В.А. Климанов****Научный руководитель – д.т.н., профессор А.Г. Коробейников**

Идея Real-Time OLAP заключается в том, что большинство данных, запрашиваемых пользователем, являются агрегированными, поэтому нет необходимости хранить детальные данные как в хранилище данных (далее ХД), так и в многомерной базе данных (далее МБД), как при использовании MOLAP архитектуры. Целью данной работы является исследование функционирования систем OLAP реального времени.

Введение

Классический OLAP предполагает отражение изменений, произошедших в транзакционных источниках, не непрерывно, а в определенные моменты времени. Частота синхронизации куба может колебаться в самых широких пределах в зависимости от природы приложения, скорости обновления и объемов данных [2].

В ситуациях крайне динамично развивающихся бизнес-процессов, когда значительная часть данных в хранилище устаревает очень быстро, а аналитик должен работать не с исторической, а с постоянно обновляемой информацией (например, биржевые торги), такой подход неприемлем. Требуется сразу по мере поступления даже одиночной транзакции отражать ее в кубе и пересчитывать с ее учетом агрегации. Приложения подобного рода относятся к области OLAP реального времени (real-time OLAP).

В данной работе производится исследование функционирования работы приложений, построенных на основе OLAP реального времени, производится оценка минимального времени между запросами в таких системах и максимального числа таких запросов.

Особенности функционирования систем OLAP реального времени

Появившаяся в SQL Server 2000 поддержка ROLAP-измерений и индексированных (материализованных) представлений (indexed views) позволяет реализовать системе OLAP реального времени ценой меньших усилий с точки зрения администрирования. Если форматом хранения раздела куба является ROLAP, Analysis Services по умолчанию пытаются создать индексированное представление для хранения агрегатов данного раздела. В отличие от классических, индексированные представления содержат не только определение запроса, но и его результаты, которые обновляются всякий раз, когда происходят изменения в исходных данных [3]. С этой точки зрения индексированные представления ведут себя подобно обычным индексам, что позволяет поддерживать содержащиеся в них агрегаты в постоянно актуальном состоянии. Специально выделенный поток (listener thread) служит средством коммуникации, с помощью которого аналитический сервер получает от реляционного источника данные об изменениях, произошедших в таблицах, соответствующих ROLAP-разделам и ROLAP-измерениям, и автоматически обрабатывает их в фоновом режиме, очищая серверный кэш. Включение оповещений достигается установкой свойства Enable Real-Time Update при создании ROLAP-раздела или ROLAP-измерения.

Использование OLAP реального времени налагает определенные ограничения, так как в качестве реляционного источника при этом должен использоваться только SQL Server 2000 (иначе не будет работать механизм оповещений), причем, очевидно, Enterprise Edition (должны присутствовать индексированные представления). Аналитический сервер также должен быть построен на основе SQL Server 2000 Enterprise

Edition, так как функциональность ROLAP-измерений включена только в эту версию, как отмечалось выше. ROLAP-разделы, построенные на основе индексированных представлений, не могут включать меры с агрегатными функциями $\min()$, $\max()$, должны основываться на таблицах (не на представлениях), источником меры должно быть поле типа `not null`, и т.д.

Платой за реальное время являются потери в скорости выполнения OLAP-запросов. SQL Server 2000 позволяет их снизить, допуская непустое множество агрегатов за счет поддержки индексированных представлений. Кроме того, гибкая архитектура куба предполагает дальнейшую оптимизацию. Например, можно разбить куб на два раздела: меньший, ROLAP, соответствующий ситуации реального времени, основанный на данных, скажем, текущего месяца, и MOLAP-раздел, содержащий всю остальную историческую информацию.

Несколько модифицированный вариант этой идеи состоит в том, чтобы создавать не разделы, а отдельные кубы для текущих и архивных данных и строить объединяющий их виртуальный куб, с которым и будут работать конечные пользователи.

Для наглядности изобразим структуру типичной OLAP системы реального времени:

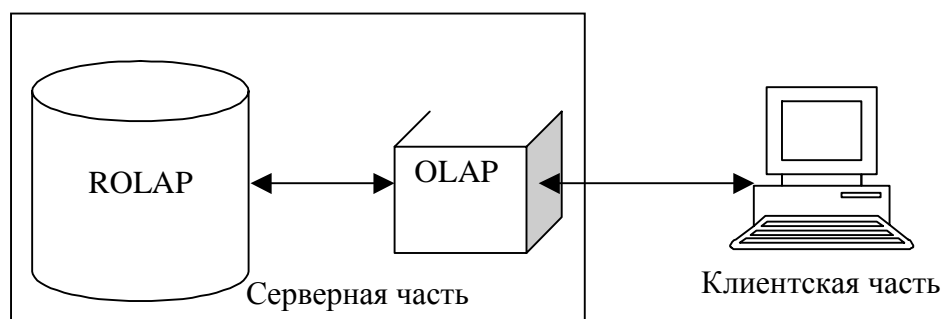


Рис. 1. Структура Real-Time OLAP

Упрощенно процесс функционирования системы OLAP реального времени выглядит так. В реляционное хранилище приходит транзакция, обновляющая (дополняющая) значения какого-либо измерения. По этому событию вызывается пересчет агрегированных значений куба. Естественно, это занимает определенное время – назовем его временем расчета агрегатов t . После выполнения сервером описанных операций пользователь получает в клиентском приложении обновленные агрегаты с учетом каждой поступившей транзакции обновления измерений. Эти транзакции поступают на сервер через некоторый промежуток времени – назовем его интервалом обновляющих транзакций I .

Оценка ограничений на функционирование Real-Time OLAP

Для проведения экспериментов, связанных с OLAP реального времени, использовался Microsoft SQL Server 2000 SP2, установленной на рабочей станции с одним процессором тактовой частоты 600 МГц и 256 Мб оперативной памяти, под управлением Windows 2000 Professional. Для выполнения экспериментов использовалось небольшое приложение, написанное на Visual Basic, инициирующее обновляющие транзакции на сервере через определенные промежутки времени. При этом производились замеры времени расчета агрегатов для каждой транзакции. В качестве клиента использовалась рабочая станция под управлением Windows 2000 Professional с процессором тактовой частоты 1,8 ГГц и 256 Мб оперативной памяти.

Экспериментальный куб данных содержал три измерения. Каждое измерение содержало 1000 значений, объединенных в сбалансированную иерархию. Таким образом, куб, используемый для определения основных характеристик Real-Time OLAP системы, имел структуру, представленную в табл. 1.

Уровень	Dim1	Dim2	Dim3
1	1	1	1
2	10	10	10
3	100	100	100

Таблица 1. Структура МБД

Единичная обновляющая транзакция добавляла 10 значений на нижний уровень иерархии, в результате чего инициировался пересчет агрегатов на верхних уровнях. После проведения каждой серии экспериментов исходный куб восстанавливался исходным числом значений измерений.

В общем случае поток обновляющих транзакций может быть распределен по случайному закону. В данной работе мы рассмотрим случай, когда интервал обновляющих транзакций является детерминированным, т.е. временные значения этого интервала постоянны: $I = I_1 = I_2 = \dots = I_N = \text{const}$.

Для достижения максимальной производительности, доступной для данной аппаратной конфигурации сервера, важно, чтобы соблюдалось следующее условие:

$$T < I, \quad (1)$$

где T – время расчета агрегатов после поступления единичной обновляющей транзакции. Однако данный случай в работе сервера встречается нечасто и может быть рассмотрен лишь как идеальный вариант. На практике же бывает так, что транзакции поступают с интервалом меньшим, чем время обработки одной транзакции. В этом случае сервер задействует встроенные средства параллельной обработки, используя общие ресурсы процессора и памяти, в результате чего время выполнения одной транзакции увеличивается по сравнению со временем, как если бы выполнялась только лишь одна эта транзакция.

Очевидно, что время, затрачиваемое на выполнение расчета агрегатов для каждой транзакции, зависит в общем случае от интервала следования обновляющих транзакций и их числа:

$$t_i = t(I_i, N) = t(I, N). \quad (2)$$

Поэтому для определения минимального интервала следования обновляющих транзакций необходимо варьировать обе эти величины.

Для наглядности построим графики зависимостей полученных результатов. Зависимость времени расчета агрегатов от числа обновляющих транзакций представлена на рис. 2. На рис. 2 интервалы следования обновляющих транзакций связаны следующей зависимостью: $I_1 < I_2 < I_3 < I_4$.

Экспериментально было показано, что при достижении некоторого значения числа обновляющих транзакций N_{\max} время расчета агрегатов становилось бесконечно большим. Таким образом, было показано, что для Real-Time OLAP систем существует некоторое максимальное значение числа обновляющих транзакций, выше которого сервер уже не может их обработать:

$$t \rightarrow \infty \Big|_{N=N_{\max}}. \quad (3)$$

При достижении этого значения время расчета агрегатов резко увеличивается, а время ожидания ответа на запрос клиентского приложения также стремится к бесконечности, что следует из зависимости (3).

Работа OLAP системы реального времени ограничивается также еще одной величиной – минимальным интервалом следования обновляющих транзакций I_{\min} . На рис. 3 представлена зависимость времени расчета агрегатов от интервала между обновляющими транзакциями.

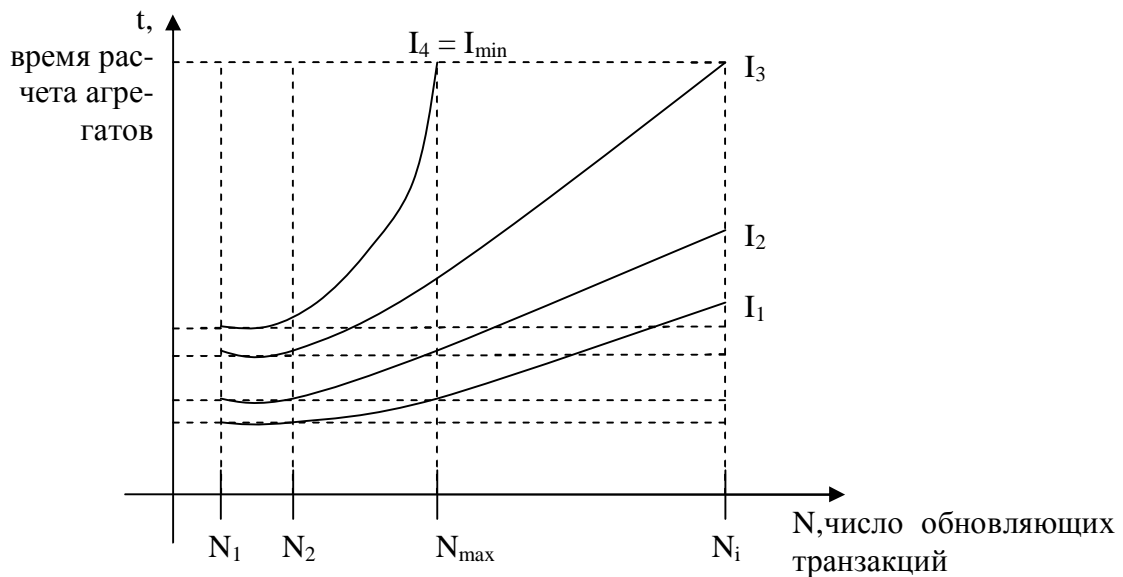


Рис. 2. Время расчета агрегатов в зависимости от числа обновляющих транзакций

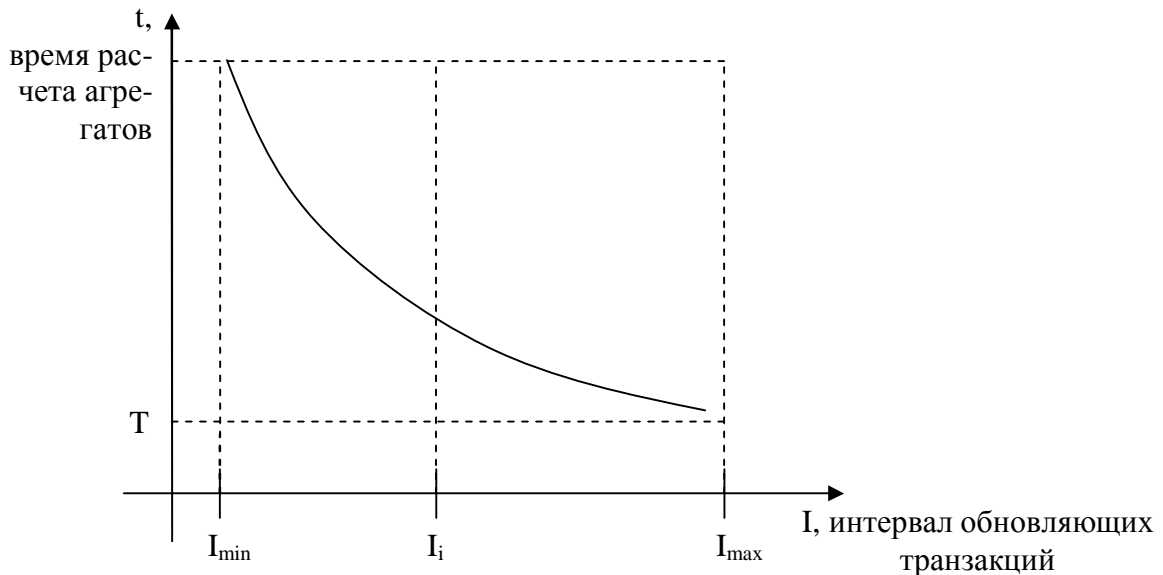


Рис. 3. Зависимость времени расчета агрегатов от интервала обновляющих транзакций

Как видно из рис. 3, время расчета агрегатов становится бесконечно большим при достижении некоторого интервала обновляющих транзакций I_{\min} :

$$t \rightarrow \infty \Big|_{I=I_{\min}}. \quad (4)$$

При достижении этого значения время расчета агрегатов резко увеличивается, и время ожидания ответа клиентского приложения также стремится к бесконечности, что следует из зависимости (4).

Таким образом, экспериментально мы показали, что для любой OLAP системы реального времени существуют ограничения на ее нормальное функционирование ввиду наличия определенного числа обновляющих транзакций и минимального интервала их следования. В данном эксперименте для описанной аппаратной платформы эти значения составили:

$$\begin{aligned} N_{\max} &= 35 \text{ обновляющих транзакций,} \\ I_{\min} &= 5,5 \text{ сек.} \end{aligned}$$

Заключение

В данной работе были определены особенности функционирования OLAP систем реального времени, а также основные критические характеристики данной архитектуры, влияющие на время ответа на пользовательские запросы, – максимальное число обновляющих транзакций и минимальный интервал их следования. Экспериментальная часть работы показывает важность учета этих характеристик при проектировании подобных систем.

Литература

1. Хрусталёв Е.М. Агрегация данных в OLAP-кубах. // Открытые системы. 2003. № 5. С. 33–38.
2. Архипенков С.Я. Хранилища данных. М.: Диалог-МИФИ, 2002. 560 с.
3. Шуленин А.А. Масштабируемость аналитических систем. // Windows & .Net Magazine/RE. 2002. № 2. С. 13–17.

МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ HOLAP СИСТЕМ

А.К. Дорожкин

Научный руководитель – к.т.н., профессор В.В. Кириллов

Идея гибридного OLAP (HOLAP) заключается в том, что большинство данных, запрашиваемых пользователем, являются агрегированными, поэтому нет необходимости хранить детальные данные, как в хранилище данных (далее ХД), так и в многомерной базе данных (далее МБД) как при использовании MOLAP архитектуры. Целью данной работы является разработка модели, отображающей эффективность применения HOLAP архитектуры в зависимости от характера данных, хранимых в многомерной базе данных (далее МБД).

Введение

Преимущество HOLAP подхода перед MOLAP заключается в том, что при раздельном хранении агрегированных и детальных данных исключается дублирование детальных данных в ХД и МБД, за счет чего большее число агрегатов может быть предварительно рассчитано, что ведет к уменьшению времени отклика системы. Однако данный подход имеет недостаток, связанный с тем, что при обращении к детальным данным на сеть, в которой функционирует система, приходится двойная нагрузка, так как сначала данные извлекаются сервером МБД из ХД (реляционная база данных), а уже затем передаются пользователю. Поэтому при описании данного подхода необходимо говорить о таком понятии, как степень избыточности сетевой нагрузки, которую определим как отношение размера данных, передаваемых от ХД к МБД, к общему размеру данных, передаваемых по сети. Далее по тексту мы будем называть степень избыточности сетевой нагрузки степенью избыточности HOLAP. Эту степень избыточности можно определить как

$$R_{HOLAPi} = \frac{C_{det}}{C_{det} + C_{agg}}, \quad (1)$$

где C_{det} – размер детальных данных, C_{agg} – размер агрегированных данных, включающий в себя не только предварительно рассчитанные агрегированные данные, хранимые в МБД, но и рассчитанные на лету на основе детальных данных.

Другим аспектом функционирования HOLAP является вероятность обращения к детальным данным (П), от которого напрямую зависит степень избыточности HOLAP. Следует отметить, что под обращением к детальным данным следует понимать не только непосредственное отображение детальных данных в клиентском приложении, но и обращение к детальным данным для расчета агрегированных значений на лету. Чем меньше эта вероятность, тем меньше задержка ответа на запрос пользователя, который в общем случае состоит из трех составляющих: t_{ext} – времени извлечения данных из источника, которое, согласно [4], прямо пропорционально числу записей; t_{agg} – время агрегации на лету [3]; t_{net} – время передачи данных по сети [4]. Составляющие t_{ext} и t_{net} , в свою очередь, состоят из двух частей, относящихся как к реляционному, так и к многомерному источникам данных:

$$t_{user} = t_{ext} + t_{agg} + t_{net},$$

$$t_{user} = \left(C_{rdb} * t_b + t_{query}^{rdb} + C_{mdb} t_b + t_{query}^{mdb} \right) + O_{agg} * t_{el} + \left(\frac{C_{mdb}}{P_{mdb}} + \frac{C_{rdb}}{P_{rdb}} \right),$$

где индекс mdb обозначает характеристики, относящиеся к участку сети между многомерной базой данных и пользовательским приложением, rdb – к участку сети между реляционным источником и многомерной базой данных. Если C_{mdb} содержит в себе как агрегированные, так и детальные данные, то C_{rdb} – только детальные данные.

Основные характеристики модели HOLAP системы

Объем данных, запрашиваемый пользователем, можно представить как

$$C_{user} = C_{agg}^{per} + C_{agg}^{on-fly} + C_{det}^{user} + C_{det}^{on-fly},$$

где C_{agg}^{per} – размер предварительно агрегированных данных, хранящихся в МБД, C_{agg}^{on-fly} – размер агрегированных данных, которые рассчитываются налету и, соответственно, требуют извлечения детальных данных из реляционного источника, C_{det}^{user} – размер детальных данных, запрашиваемых пользователем, C_{det}^{on-fly} – размер детальных данных, необходимых для организации агрегации на лету. Именно два последних элемента отвечают за степень избыточности HOLAP. Таким образом, степень избыточности HOLAP можно выразить как

$$R_{holap} = \frac{C_{det}^{user} + C_{det}^{on-fly}}{C_{agg}^{per} + C_{agg}^{on-fly} + 2 * C_{det}^{user} + C_{det}^{on-fly}}. \quad (2)$$

Для наглядности изобразим распределение этих элементов на схеме (рис. 1).

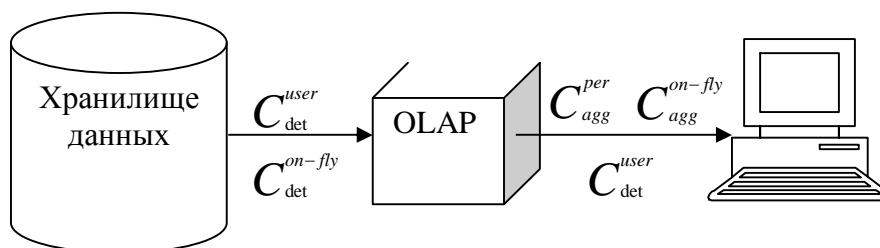


Рис. 1. Сетевая структура HOLAP

Теперь мы вплотную подошли к выражению степени избыточности HOLAP в зависимости от вероятности обращения к детальным данным и зависимости времени ответа от этой вероятности. В общем виде степень избыточности HOLAP в зависимости от вероятности обращения к детальным данным выглядит следующим образом:

$$R_{holap}(\Pi) = \frac{\Pi * (C_{det}^{user} + C_{det}^{on-fly})}{C_{agg}^{per} + \Pi * (C_{agg}^{on-fly} + 2 * C_{det}^{user} + C_{det}^{on-fly})}.$$

Вывод зависимости размера агрегированных данных от размера детальных является достаточно трудоемкой задачей, выходящей за рамки данной работы. Для наглядности построим график зависимости R_{HOLAP} от вероятности обращения к детальным данным (рис. 2).

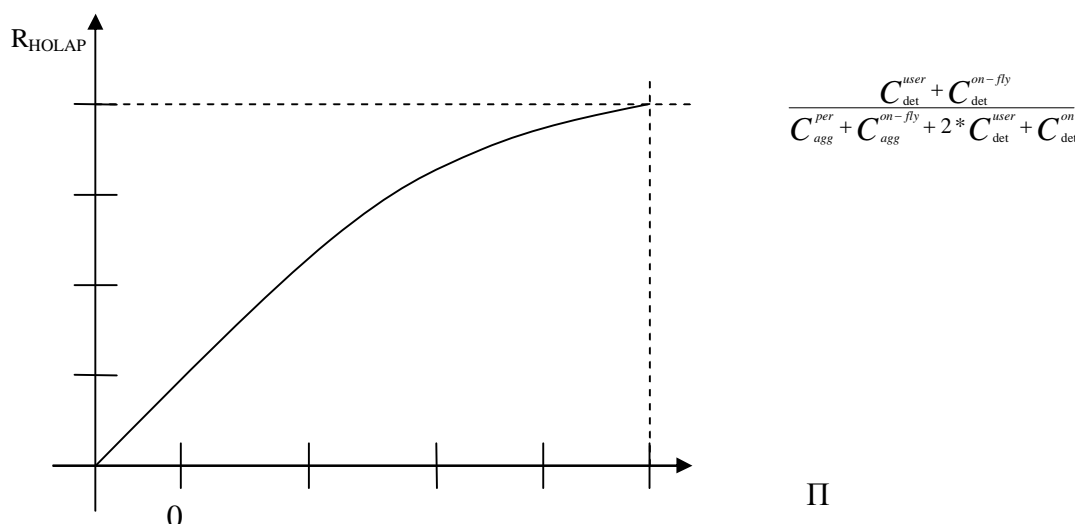


Рис. 2. Зависимость степени избыточности HOLAP от вероятности обращения к детальным данным

Определившись с такими показателями, как время ответа на запрос и избыточность HOLAP, попытаемся выразить их взаимосвязь. Для определения зависимости $t_{user}(R)$ воспользуемся выражением (1), которое запишем как $R_{HOLAP} = \frac{C_R}{C_n + C_R}$, где C_R –

размер избыточных данных (данные относящиеся к ХД), а C_n – размер необходимых данных (многомерные данные). Отсюда получаем, что $C_R = C_n \frac{R_{HOLAP}}{1 - R_{HOLAP}}$. Таким образом,

время ответа на запрос в зависимости от степени избыточности можно выразить как

$$t_{user}(R) = (C_R * t_b + t_{query}^{rdb} + C_n t_b + t_{query}^{mdb}) + O_{agg} * t_{el} + \left(\frac{C_n}{P_{mdb}} + \frac{C_\tau}{P_{rdb}} \right).$$

Для наглядности введем следующие коэффициенты: k_e – среднее время извлечения одного байта из источника данных, k_a – среднее время агрегации одного байта, k_n – среднее время передачи одного байта по сети. Таким образом, время ответа на запрос пользователя определяется следующим образом:

$$\begin{aligned} t_{user}(R) &= k_e C_n + k_e C_n \frac{R_{HOLAP}}{1 - R_{HOLAP}} + k_a C_n \frac{R_{HOLAP}}{1 - R_{HOLAP}} + k_n C_n + k_n C_n \frac{R_{HOLAP}}{1 - R_{HOLAP}} = \\ &= C_n \left((k_e + k_a + k_n) \frac{R_{HOLAP}}{1 - R_{HOLAP}} + (k_e + k_n) \right). \end{aligned}$$

Зависимость времени ответа на запрос от степени агрегации в общем виде можно записать как

$$t_{user}(R) \sim C_n \left(\frac{k_e + k_n}{1 - R_{HOLAP}} + k_a \frac{R_{HOLAP}}{1 - R_{HOLAP}} \right). \quad (3)$$

Графически такая зависимость изображена на рис. 3.

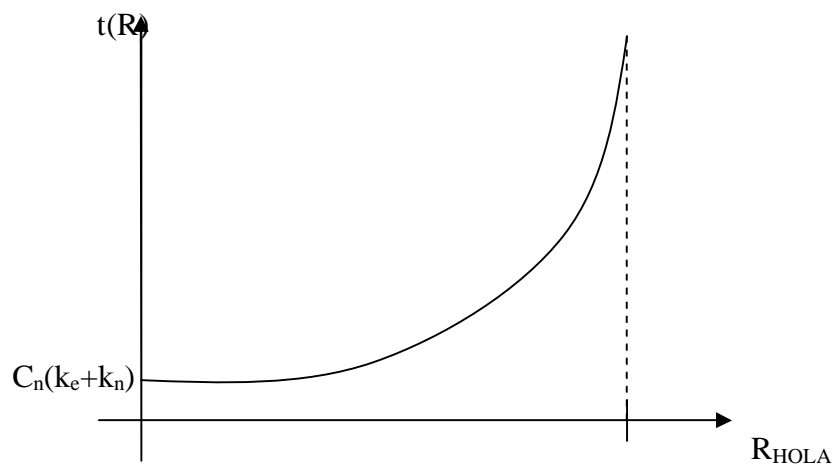


Рис. 3. Время ответа на запрос в зависимости от степени избыточности HOLAP

Оценка адекватности модели HOLAP

Для проведения экспериментов, связанных с HOLAP архитектурой, использовался Microsoft SQL Server 2000 SP2, установленной на рабочей станции с одним процессором тактовой частоты 600 МГц и 256 Мб памяти, под управлением Windows 2000 Professional. В качестве ХД использовался сервер базы данных Oracle, подключение к которому осуществлялось с помощью Oracle OLE DB Provider 9.2.0.4. Для выполнения экспериментов мы использовали небольшое приложение, написанное на Visual Basic, выполняющее подключение к OLAP серверу по ADO MDB и производящее извлечение данных. При этом производились замеры времени. В качестве клиента использовалась рабочая станция под управление Windows 2000 Professional с процессором с тактовой

частотой 1,8 ГГц и 256 Мб памяти. Для оценки передаваемых по сети объемов данных использовался, так же как и в предыдущих экспериментах, DU Meter 2.2.

Для всесторонней оценки работы HOLAP системы необходимо, чтобы извлекаемая и МБД информация содержала в себе как детальные, так и агрегированные данные. Многомерный куб, используемый для определения основных характеристик HOLAP системы имел структуру, представленную в табл. 1.

Уровень	Dim1	Dim2	Dim3
1	1	1	1
2	10	100	10
3		1000	20
4			100
5			1000

Таблица 1. Структура МБД

В рамках эксперимента многомерная база данных была полностью загружена данными, т.е. плотность детальных данных была равна 1. В рамках данного эксперимента мы не использовали расчет агрегатов для наглядности расчетов.

Исходя из указанных выше параметров МБД, можно определить степень избыточности HOLAP для всей базы данных, которая, согласно выражению (2), в данном случае будет равна

$$R_{holap} = \frac{C_{det}^{user}}{C_{agg}^{per} + 2 * C_{det}^{user}} = \frac{10 * 10^6}{17.4 * 10^6 + 2 * 10 * 10^6} = 0,267.$$

Для оценки соответствия рассчитанного значения степени избыточности HOLAP реальным показаниям были произведены замеры объемов данных, поступающих их ХД на сервер МБД, а также от МБД в пользовательское приложение. В качестве тестового набора данных использовалась кросс-таблица, по строкам которой были расположены члены измерения Dim2, а по столбцам – Dim3. Таким образом, количество извлекаемых ячеек определяется как произведение числа всех членов данных измерений, т.е. равно $1101 * 1131 = 1245231$, количество ячеек детальных данных составляет $1000 * 1000 = 1000000$. Исходя из этого, степень избыточности HOLAP извлекаемых данных равняется

$$\frac{1000000}{2 * 1000000 + 1245231} = 0,30814.$$

Вторым аспектом в данной модели HOLAP системы является зависимость времени ответа на пользовательский запрос от степени избыточности HOLAP. В рамках эксперимента по этому вопросу использовалась модель данных с полностью рассчитанными агрегатами. Для изменения степени агрегации менялось количество членов в верхних уровнях иерархий измерений Dim2 и Dim3, как показано в табл. 2 (число членов на первом и последнем уровне оставалось постоянным).

№	Dim2.l2	Dim3.l4	Dim3.l3	Dim3.l2
1	100	100	20	10
2	200	200	40	20
3	500	500	100	50
4	1000	1000	200	100

Таблица 2. Изменение распределения членов измерений по уровням иерархий

Результаты измерений трафика на сервере МБД и клиенте, а также времени ответа на запрос приведены в табл. 3 (значения трафика приведены в МБ, а времени в секундах):

№	Сервер МБД	Клиент	Время ответа на запрос
1	13,25	27,22	528
2	13,27	29,79	565
3	13,37	42,66	572
4	13,5	70,21	885

Таблица 3. Результаты измерения трафика и времени ответа на запрос

Расчеты, произведенные на основе результатов измерений, сведены в табл. 4.

Число членов в измерении		Степень избыточности HOLAP		Ошибка (%)
Discip	Grups	Расчетное	Измеренное	
1101	1131	0,308	0,327	5,8
1201	1261	0,284	0,308	7,6
1501	1651	0,223	0,238	6,4
2001	2301	0,151	0,161	6,1

Таблица 4. Результаты расчетов степени избыточности HOLAP

Для наглядности построим графики зависимостей полученных результатов. Зависимость степени избыточности HOLAP от количества членов в уровнях агрегации представлена на рис. 4. Как видно из рис. 4, с ростом числа агрегатов степень избыточности OLAP уменьшается.

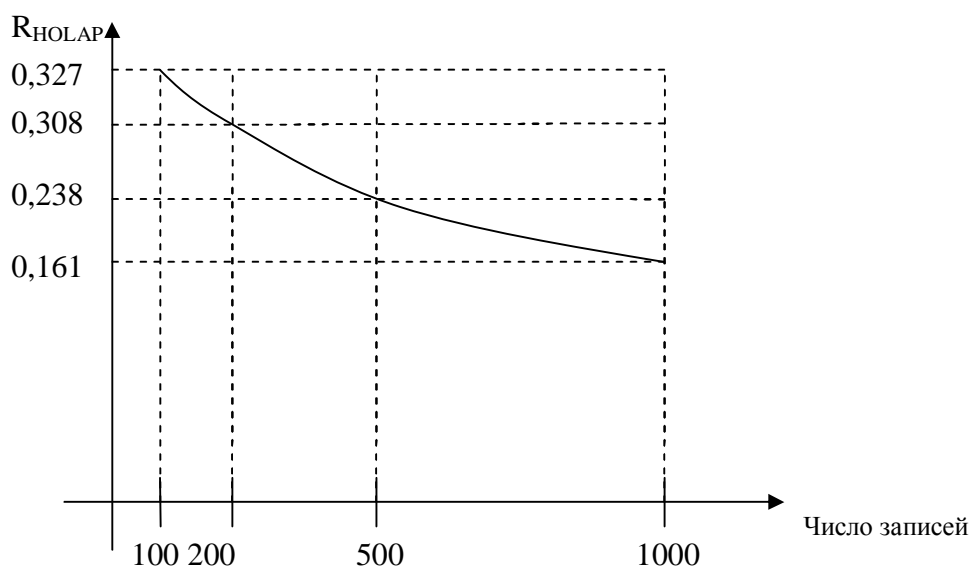


Рис. 4. Степень избыточности HOLAP от числа членов измерений

Время ответа на пользовательский запрос, полученное в результате экспериментов, необходимо предварительно «нормализовать», так как замеры производились для различного количества ячеек. Поэтому для отображения зависимости времени ответа на запрос от степени избыточности HOLAP необходимо полученное выражение разделить на число ячеек извлекаемого кубоида. Результаты «нормализации» изображены на

рис. 5 (на графике приведено среднее время извлечения одной ячейки в миллисекундах). Как видно из рис. 5, время ответа на пользовательский запрос уменьшается с увеличением количества членов измерений. Так как в данном эксперименте с увеличением количества членов измерений уменьшается степень избыточности HOLAP, то, следовательно, с увеличением степени избыточности HOLAP увеличивается время ответа на пользовательский запрос, т.е. график на рис. 5 демонстрирует точно такую же зависимость, как и выражение (3), что говорит об адекватности разработанной модели.

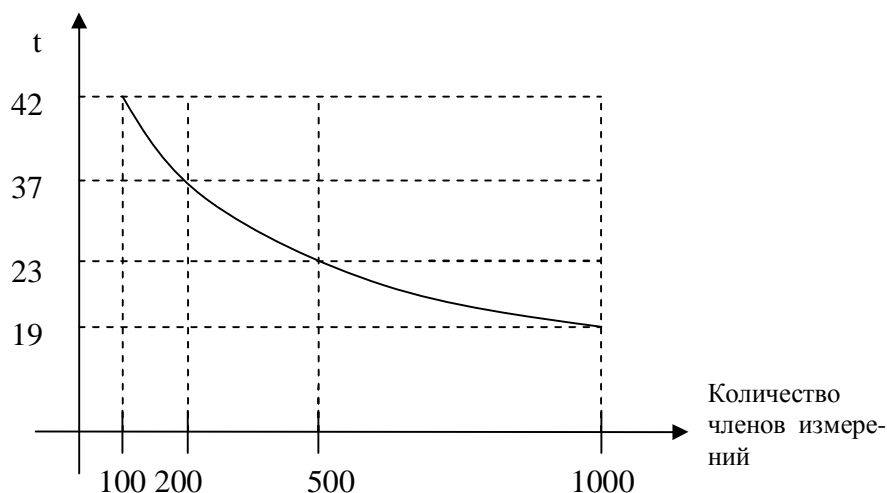


Рис. 5. Время ответа на запрос в зависимости от количества членов измерений

Заключение

В данной работе мы рассмотрели разработанную модель гибридного OLAP (HOLAP), определили основные характеристики данной архитектуры и вывели основные закономерности, влияющие на время ответа на пользовательские запросы в зависимости от распределения данных в рамках данной архитектуры. Экспериментальная часть работы показывает адекватность разработанной модели.

Литература

1. Архипенков С.Я. Хранилища данных. М.: Диалог – МИФИ, 2002. 528 с.
2. Олифер Н.А., Олифер В.Г. Компьютерные сети. СПб: Питер, 2000. 672 с.
3. Хрусталёв Е.М., Агрегация данных в OLAP-кубах. Алеф Консалтинг & Софт, 2000. 11 с.
4. V. Narinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. // Proc. ACM SIGMOD, 1996. P. 205–216.

РАЗРАБОТКА РОССИЙСКОГО СЕГМЕНТА GRID ДЛЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ПРИЛОЖЕНИЙ

И.В. Шошмина

Научный руководитель – д.ф.-м.н., профессор А.В. Богданов

Координация вычислений с размещением данных является серьезной проблемой в Grid, значительно влияющей на эффективность выполнения приложений. В статье рассматривается возможный набор из уже существующих программных средств, позволяющий хотя бы частично решить эту проблему для параллельных задач на Grid.

1. Введение

Исходя из сложнейших задач и активно развивающихся технологий, современная наука создается как усилиями отдельных ученых, так и взаимодействием целых научных сообществ. Рост вычислительных мощностей, систем хранения, сетевых и программных технологий, высокие требования ученых подстегнули создание и развитие технологии объединения компьютерных ресурсов Grid. Она разрабатывается, чтобы помочь ученым, исследователям, инженерам различных областей науки в решении научных и технических проблем, обеспечив их прозрачным, надежным, удобным доступом к вычислительным ресурсам. Grid – это динамически меняющаяся распределенная разнородная среда, содержащая большое количество разнообразных компьютерных, программных, информационных ресурсов и совокупность механизмов, позволяющих управлять (в широком смысле) этой средой [1]. Предполагается, что в идеале Grid будет всепроницающей, подобно электрической сети, т.е. ученый в любой точке, имеющей подключение к глобальной компьютерной сети Интернет, сможет произвести необходимые вычисления на соответствующим требованиям его приложения ресурсах.

Несмотря на интенсивное развитие Grid, многие реальные приложения, связанные с высокопроизводительными вычислениями, до сих пор остаются в значительной степени не охваченными Grid технологией.

Мы предлагаем реализовать комбинированную модель Grid на российском сегменте Grid для высокопроизводительных приложений.

Цель данного исследования – подбор программного обеспечения промежуточного уровня для Grid высокопроизводительных приложений.

Статья построена следующим образом: в разделе 2 кратко изложено развитие Grid архитектуры, в разделе 3 описаны три модели Grid, в разделе 4 приводится физическая архитектура и базовая конфигурация Grid для высокопроизводительных приложений, в разделе 5 обсуждаются возможные пакеты программ для реализации функций комбинированной модели Grid, разделом 6 статья завершается.

2. Современное развитие архитектуры Grid

Термин Grid является одним из перегруженных терминов в компьютерной науке. В рассматриваемом контексте термин Grid употребляют для обозначения как инфраструктуры в целом, содержащей разнообразные ресурсы и программные средства их управления, так и для обозначения только программных средств промежуточного уровня для управления ресурсами. Мы далее будем придерживаться определения, данного во введении, подразумевающего под Grid инфраструктуру в целом.

В классическом изложении для структурирования Grid используется архитектура Grid, которая определяет фундаментальные компоненты системы, задает их цели и функции, описывает принципы взаимодействия компонент. Для сохранения наиболее общего подхода в описании архитектуры используются так называемые протоколы Grid [1].



Рис. 1. Многоуровневая архитектура протоколов Grid (из [1], русскоязычный перевод из [2])

Архитектура протоколов Grid является многоуровневой (см. рис. 1). На нижнем, базовом уровне расположены гетерогенные, динамически изменяющиеся, географически распределенные ресурсы (компьютеры, системы хранения данных, сети), а на верхнем уровне находятся сами приложения, которые и должны на этих ресурсах выполняться. Протоколы, отвечающие за организацию выполнения пользовательских приложений на ресурсах, находятся на промежуточных уровнях. Собственно, они и являются наиболее специфичной для Grid частью. Протоколы определяют требования к сервисам Grid. Сервисы Grid, в свою очередь, реализуются программным обеспечением промежуточного уровня. В соответствии с классической концепцией Grid был разработан пакет Globus Toolkit 2 [3].

Изначально в архитектуре протоколов Grid подразумевалось, что нижележащие уровни ничего не знают о вышестоящих, тогда как вышестоящие должны обращаться, в основном, к ближайшему расположенному ниже уровню. При реализации идей Grid на практике стройную иерархическую архитектуру протоколов сохранить не удалось. Существуют некоторые сервисы, например Replica Catalog – каталог дублированных данных, которые не могут быть отнесены к какому-нибудь определенному уровню. Оказалось, что, несмотря на наличие некоторых уровней в реальной инфраструктуре Grid, более точно область контроля и управления ресурсами описывается концепцией OGSA (Open Grid Services Architecture – открытая архитектура Grid сервисов) [4]. В OGSA произошел переход на новый уровень обобщения – основной единицей инфраструктуры стал сервис Grid. Сервисы Grid, в отличие от классического подхода, – это вычислительные ресурсы, ресурсы данных, сети, программы, базы данных, – т.е. все, что составляет Grid, включая и само ПО промежуточного уровня. OGSA предлагает унифицированный подход ко всем своим аппаратным и программным компонентам. При разработке концепции OGSA наибольшее внимание отводилось вопросам унифицированного создания, модификации, удаления сервиса Grid, тогда как функционал программного обеспечения инфраструктуры остался без должной проработки. На основе OGSA был создан программный пакет Globus Toolkit 3. Однако он оказался нестабильным, очень требовательным к ресурсам [5] и потому значительного практического распространения не нашел.

Концепцию Grid OGSA сменила WSRF (Web Services Resource Framework) [6], которая, с одной стороны, организует сервисы Grid через Web сервисы, с другой стороны, позволяет максимально унифицировать работу с сервисами Grid, за счет чего удастся разделить функции сервисов, улучшить их спецификацию, повысить управляемость. Globus Toolkit 4 будет основан на WSRF.

В работе [7] можно найти описание разделения сервисов Grid (для ПО gLite) по функциям: безопасность, доступ, мониторинг и сбор информации, управление зада-

ниями и управление данными. Таким образом, сервисы Grid образуют множества в зависимости от выполняемых ими функций, а множества пересекаются определенным образом, обеспечивая общий функционал инфраструктуры Grid.

Концепции OGSA, WSRF требуют доработки для создания реальных Grid на их основе.

3. Комбинированная модель Grid

Практический подход при разработке Grid ближе всего к исходной классической концепции.

Для создания реального Grid необходимо наличие следующих базовых сервисов (Grid Common Services) [8, 9]: средства, обеспечивающие безопасность в Grid; механизм обнаружения ресурсов в Grid; Grid механизм, инициирующий выполнение заданий; базовый механизм управления передачей данных; механизм планирования заданий Grid. Кроме того, в набор основных сервисов входит средство реализации коммуникаций, учитывающее структуру безопасности Grid, (например, Globus I/O). В Grid, работающих для реальных практических задач, эти сервисы обеспечиваются пакетом Globus 2.

Хотя при разработке инфраструктуры Grid в проектах разработчики исходят из конкретных целей и задач, на данный момент на практике можно выделить две основных модели Grid: Grid данных и вычислительный Grid. Основное отличие этих моделей в типах приложений, для решения которых они предназначены. Grid данных занимаются приложениями, обрабатывающими большие массивы данных; тогда как вычислительные Grid направлены на сегодняшний момент на использование в слабо связанных вычислительных приложениях или на параметрические исследования, предполагающие запуск одного и того же приложения на нескольких системах в пакетном режиме (подробнее о моделях Grid в [9]). Эти модели едины в базовых сервисах, но для решения их специфических задач каждая модель надстраивает свои дополнительные программные средства.

Как было сказано во введении, высокопроизводительные параллельные приложения редко решаются на современных Grid. В некоторых работах это объясняется большими издержками, возникающими при передаче данных, нестабильной и неоднородной природой Grid [10].

С нашей же точки зрения, решение сложных задач, предъявляющих высокие требования как к вычислительным, так и к информационным ресурсам, возможно при условии адаптации алгоритмов для работы на Grid, доработки существующего (или создания нового) программного обеспечения уровня Grid для работы с подобными приложениями. Теоретические и практические предпосылки новой модели Grid – смешанной или, иначе, комбинированной модели – рассмотрены нами в [9]. Эта модель функционально совмещает в себе характеристики и Grid данных, и вычислительной Grid. Комбинированная модель Grid должна обеспечивать выполнение приложений с большим объемом вычислений при интенсивном обмене данных на динамически меняющейся среде разнородных компьютерных ресурсов.

4. Базовая конфигурация российского сегмента Grid для высокопроизводительных вычислений

Для экспериментов по работе высокопроизводительных параллельных приложений на Grid нами создается российский сегмент Grid. Он основывается на ресурсах трех организаций:

- Института высокопроизводительных вычислений и информационных систем (Санкт-Петербург),

- Института теплофизики экстремальных состояний Объединённого института высоких температур РАН (Москва),
- Института вычислительной математики и математической геофизики СО РАН (Новосибирск).

Для обеспечения базовых сервисов Grid (см. раздел 3) выбраны широко распространенный пакет Globus Toolkit 2.4: GSI обеспечивает безопасность, GIS/MDS отвечают за работу информационного сервиса (механизма обнаружения ресурсов), Globus GRAM инициирует выполнение задания в Grid, GridFTP является средством передачи данных в Grid, Globus I/O – библиотека ввода/вывода, совместимая с системой защиты.

В дополнение к Globus Toolkit 2.4 инсталлирован пакет LCG-2, который был разработан в проекте LHC [11], создающем крупнейший и мощнейший в мире ускоритель частиц. Пакет LCG изначально направлен на поддержку «вычислительного Grid данных». Хотя сейчас этот проект относят более к проектам, реализующим Grid данных, однако использование LCG для комбинированного Grid, активно работающего с данными, представляется разумным.

LCG позволяет упорядочить базовые сервисы Grid и упрощает их использование, при необходимости дополняя базовый функционал.

Минимальная базовая конфигурация Grid сегмента состоит из 5 компьютерных систем [12], на которых устанавливаются соответствующие элементы LCG: LCFG Server – сервер конфигурации, Computing Element (CE) – вычислительный элемент, Storage Element (SE) – элемент хранения данных, User Interface – элемент для пользовательского интерфейса и Working Node – рабочий узел.

Через элемент с пользовательским интерфейсом, собственно, и осуществляется доступ к ресурсам сегмента Grid. Пользователь входит на эту машину, чтобы запросить Grid ресурсы, установить свое задание на выполнение, получить его вывод, положить данные в удаленную зону хранения и вообще пользоваться доступными сервисами Grid.

С сервера конфигурации LCFG Server осуществляется установка и конфигурация, как первоначальная, так и вторичная, всех элементов управления базового уровня пакета LCG.

Вычислительный элемент является основной рабочей точкой на локальный сайт. Он отвечает за Globus Gatekeeper Grid сервис и совместно с рабочими узлами предоставляет вычислительные ресурсы пользователю. Этот узел обеспечивает единый интерфейс нижележащих вычислительных ресурсов, включая прием заданий, распределение их на выполнение, возвращение результатов работы. На нем же может быть расположена локальная система управления заданиями, которая работает с рабочими узлами.

Рабочие узлы находятся за вычислительным элементом, собственно на них и осуществляется выполнение пользовательских вычислительных заданий. На них устанавливается пользовательское ПО. На них не запускаются Globus демоны, но есть некоторые Globus API для доступа Grid сервисов и для получения информации о состоянии узла и заданий. На практике сайт содержит несколько рабочих узлов.

Элемент хранения данных обеспечивает единый доступ к большим пространствам данных. Он прячет детали работы конечных систем хранения данных и обеспечивает универсальный доступ Grid пользователю. Элемент хранения может контролировать большие дисковые массивы, системы хранения данных и тому подобное.

LCG-2 и GT 2.4, с одной стороны, обладают стабильностью классической архитектуры протоколов Grid. С другой стороны, они совместимы с Grid сервисами в понимании OGSA и потому дают возможность для дальнейшего развития Grid (обратим внимание, именно архитектура OGSA предполагает возможность универсального подключения новых сервисов к Grid).

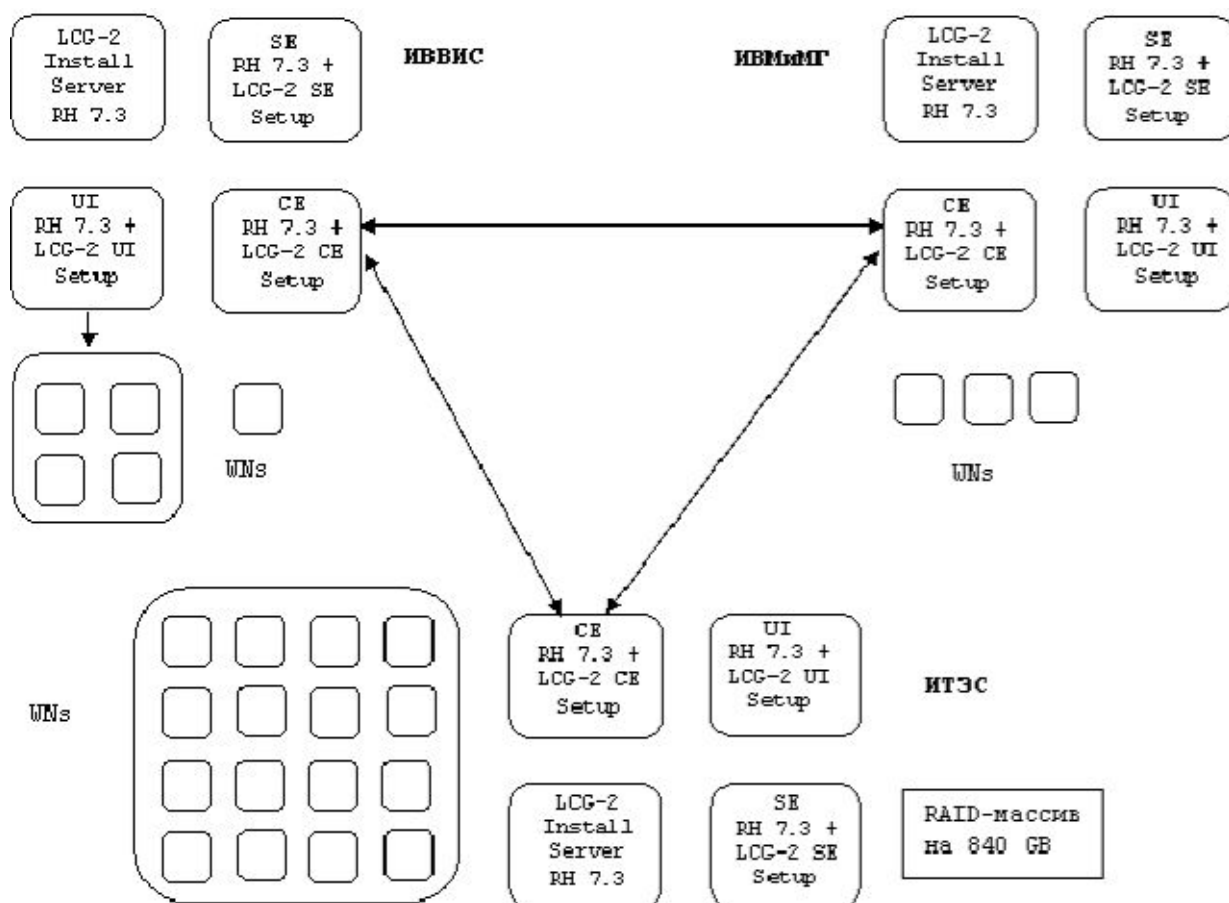


Рис. 2. Российский сегмент Grid для высокопроизводительных приложений

На данный момент базовая архитектура сегмента Grid для высокопроизводительных вычислений соответствует приведенному описанию (см. рис. 2).

5. Выбор ПО, надстраиваемого над базовыми Grid сервисами

Именно дополнительное программное обеспечение, добавляемое к базовым сервисам, завершает инфраструктуру и формирует определенную модель Grid. В [9] нами было описано дополнительное ПО, которое используется для Grid данных и вычислительных Grid. В процессе нашего исследования мы собираемся подобрать ПО, чтобы обеспечить выполнение функций комбинированной модели Grid.

Приведем сервисы, которые должны быть добавлены к базовым:

- среда обмена сообщениями параллельных приложений в Grid,
- управление заданиями пользователя на внешнем уровне,
- управление данными на внешнем уровне,
- универсальная точка доступа пользователя к ресурсам.

Приложения, для которых предназначен комбинированный Grid, относятся к разряду «тяжелых», т.е. отличаются интенсивными параллельными вычислениями с использованием больших объемов данных. Таким образом, основной целью при разработке инфраструктуры Grid для высокопроизводительных приложений является обеспечение координации выполняемых вычислений с входными/выходными данными в Grid. Как уже обсуждалось в [9], законченного решения для этой проблемы в Grid сейчас нет, поэтому мы будем подбирать ПО, которое максимально приспособляемо для данной цели.

Рассматривая класс высокопроизводительных приложений, отправной точкой при создании Grid для высокопроизводительных приложений мы считаем среду обмена сооб-

щениями. MPI и PVM поддерживают модель программирования с распределенной памятью. MPICH-G2 [13], версия ПО для Grid на базе MPI, для обмена сообщениями использует библиотеку Globus I/O, основанную на TCP/IP, взаимодействует с базовыми сервисами Grid при запуске приложения. Для увеличения скорости передачи и координации с вычислениями в MPICH-G2 используется технология Globus Data Conversion, которая позволяет накапливать подготовленные данные в буферах и высылать их сразу при запросе, а также организовывать очереди из данных. MPICH-G2 позволяет использовать MPI производителя на конкретных системах, а также поддерживает «гетерогенные» соединения, т.е. соединения между системами разных производителей.

Развитие PVM сейчас не поддерживается производителем. По этой причине PVM не интегрирован с сервисами Grid. Однако совместно с системой управления заданиями Condor-G PVM может быть использован в Grid [14, 15]. Condor Job Manager (управляющий заданиями) запускает задание с помощью GRAM, который, в свою очередь, запускает систему управления заданиями Condor на локальной системе. Как только механизм Condor Glide-In стартует, обеспечивается среда для работы PVM. Наиболее управляемыми на Grid с Condor-PVM оказываются приложения, использующие парадигму мастер-работники, так как Condor-G имеет соответствующее программное средство MW. Несмотря на то, что применение Condor-G совместно с PVM на Grid выглядит более проблематичным, однако для PVM, в отличие от MPI, имеются средства, обеспечивающие отказоустойчивость и балансировку нагрузки DPVM [16]. Проблема устойчивости в Grid стоит очень остро. Сбой на одном узле, на котором выполнялось задание параллельного приложения, приводит к необходимости повторения всех вычислений на всех узлах заново, поэтому откидывать возможность восстановления с последней сохраненной точки (а именно ее предоставляет DPVM) представляется неразумным.

Для управления параллельными заданиями и данными на внешнем уровне, т.е. уровне распределения заданий между сайтами, требуется некоторая специализированная система. Для управления заданиями в Grid известны достаточно развитые системы PBSPro, Moab Silver, но они являются коммерческими (нас же интересует свободно распространяемое ПО). В то же время для полноценного управления данными в Grid, допускающими Grid-кэширование, совместное распределение данных и заданий, а также дублирование данных в зависимости от прогноза их использования, системы и вовсе отсутствуют.

Рассмотренная ранее Condor-G управляет параллельными приложениями независимо от среды обмена сообщениями (и для PVM, и для MPI). Она может работать с разными локальными системами управления заданиями. При помощи Condor-G пользователь устанавливает приложение в очередь на выполнение. Condor-G размещает приложение в очередь в соответствии с его приоритетами, требованиями к ресурсам, временным ограничениям. Далее система находит при помощи сервисов Grid необходимую систему/системы и отправляет приложение на выполнение на локальную машину в Grid. Пользователь может посмотреть текущий статус своего задания. Но свойства Condor-G ограничены исключительно управлением заданиями без относительно месторасположения данных.

Пакет Nimrod-G [9, 17] занимается управлением и планированием вычислений в целом на Grid, начиная от распределения файлов на удаленных системах, вычислений и сбора результатов. Nimrod-G связывает данные и вычисления, что актуально для высокопроизводительных приложений. Он может работать как напрямую с сервисами Grid, так и с системами планирования заданий внешнего уровня типа Condor-G.

Каждая из описанных систем управления заданиями, и Nimrod-G, и Condor-G, имеет свой интерфейс для запуска приложений. Однако, если говорить об организации точки доступа к ресурсам Grid, то хотелось бы иметь более развитый графический интерфейс для комфортной работы пользователя. Нам хотелось бы воспользоваться для этих целей Migrating Desktop [18], который предоставляет доступ к ресурсам, мониторинг выполнения приложения и перемещение данных по запросу пользователя в дружественной манере.

В результате мы сформировали набор программного обеспечения, который должен обеспечить функциональность комбинированной модели Grid. В дальнейшем мы планируем провести эксперименты, оценивающие эффективность использования для высокопроизводительных приложений Grid сред обмена сообщений PVM и MPI. Мы не знаем ни одного проекта, в котором бы все выбранные пакеты использовались совместно, поэтому настройка их совместной работы на практике в Grid высокопроизводительных приложений будет также отдельным исследованием.

6. Заключение

В данном исследовании представлены теоретические основы построения инфраструктуры Grid – среды значительно географически распределенных компьютерных ресурсов.

Рассмотрены существующие и применяемые на практике модели Grid: Grid данных и вычислительный Grid, а также предложена комбинированная модель Grid, которая направлена на приложения, одинаково требовательные к вычислительным и информационным ресурсам.

Для реализации комбинированной модели Grid строится российский Grid для высокопроизводительных приложений на базе ИВВИС (Санкт-Петербург), ИТЭС ОИВТ РАН (Москва), ИВМиМГ СО РАН (Новосибирск).

Рассмотрен комплекс ПО уровня Grid для обеспечения выполнения интенсивных вычислений с активным обменом большими объемами данных.

В дальнейшем исследования будут связаны с практической реализацией функций комбинированной модели Grid при помощи выбранного ПО. Возможно, что ПО потребуются дополнить другими пакетами или модифицировать в соответствии с требованиями приложения. Также в будущем мы хотели бы реализовать Grid высокопроизводительных вычислений на основе сервисов Grid в понимании OGSA/WSRF.

Благодарность

Работа поддержана грантом NWO-RFBS-047.016.007 «High Performance Simulation on the Grid».

Литература

1. I. Foster, C. Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, // Int. Jou. of High Performance Computing Applications, Vol. 15 (3), pp. 200-222, 2001.
2. Шошмина И.В. Архитектура протоколов Grid. / Курс лекций, выполненных Институтом высокопроизводительных вычислений и информационных систем в рамках совместной российско-белорусской программы «СКИФ», 2003. http://www.csa.ru/skif/kurs_2/gg/Grid_Architecture_lec_v9.html
3. www.globus.org
4. Foster, I., Kesselman, C., Nick, J. M., Tuecke, S. The Physiology of the Grid . / Grid Computing Making the Global Infrastructure a Reality / Ed/ F.Berman, G. Fox, T.Hey, John Wiley & Sons, GB, 2003. P. 217–249.
5. «OGSA/GT3 evaluation activity at CERN», ACAT03, 2003. http://lcg.web.cern.ch/LCG/peb/GTA/LCG_GTA_Presentations.htm#Presentations%20and%20reports%20on%20OGSA/GT3%20activity
6. Czajkowski K., Ferguson D., Foster I., Frey J., Graham S., Maguire T., Snelling D., Tuecke S. From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution. 2004. <http://www.globus.org/research/papers.html>
7. EGEE Middleware Architecture and Planning (Release 1), EU Deliverable DJRA1.1, 2004, <https://edms.cern.ch/document/476451/>
8. Johnston W. E., The NASA IPG Engineering Team, and The DOE Science Grid Team «Implementing production Grids» / Grid Computing Making the Global Infrastructure a Reality. Ed. F.Berman, G. Fox, T.Hey, John Wiley & Sons, GB, 2003. P. 117–167.
9. Шошмина И.В., Богданов А.В., Абрамсон Д. Куда идет Grid? // Публикации международной конференции «Распределенные вычисления и Grid-технологии в науке и образовании», Дубна, 2004.

10. Lee C., Talia D. Grid Programming Models: Current Tools, Issues and Directions // Grid Computing Making the Global Infrastructure a Reality / Ed/ F.Berman, G. Fox, T.Hey, John Wiley & Sons, GB, 2003. P. 555–578.
11. lhc-new-homepage.web.cern.ch/lhc-new-homepage/
12. LCFGng Cluster Installation Guide Version 2.0 (LCG-2), <http://cgi.di.uoa.gr/~xgrid/archive.htm>
13. Karonis N., Toonen, B., Foster I. MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface // Journal of Parallel and Distributed Computing (JPDC). V. 63. №. 5, 2003, P. 551–563.
14. Frey J., Tannenbaum T., Foster I., Livny M., Tuecke S. Condor-G: A Computation Management Agent for Multi-Institutional Grids. // Cluster Computing, V. 5, 2002, P. 237–246.
15. Goux J.-P, Kulkarni S., Linderoth J. T., Yoder M. E. Master-Worker: An Enabling Framework for Applications on the Computational Grid. //, Cluster Computing, V. 4, 2001, P. 63–70.
16. Iskra K.A., van der Linden F., Hendrikse Z.W., Overeinder B.J., van Albada G.D., Sloot P.M.A. The implementation of Dynamite – an environment for migrating PVM tasks // Operating Systems Review, V. 34, № 3, P. 40–55. Association for Computing Machinery, Special Interest Group on Operating Systems, 2000.
17. Buyya, R., Abramson, D., Giddy, J. Nimrod/G: An Architecture of a Resource Management and Scheduling System in a Global Computational Grid // HPC Asia 2000, 2000, P. 283–289.
18. Kupczyk M., Lichwala R., Meyer N., Palak B., Plociennik M., Wolniewicz P. Applications on demand as the exploitation of the Migrating Desktop // Future Generation Computer Systems, V. 21, №.1, 2005, P. 37–44.

КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ СИСТЕМЫ УСЛОВНОГО ДОСТУПА

М.В. Костин

Научный руководитель – д.т.н., профессор Т.И. Алиев

В работе предлагается концептуальная модель системы условного доступа к информационным сервисам, передаваемым в широкополосном спутниковом или эфирном цифровом канале связи. Произведена детальная декомпозиция системы, определены основные показатели эффективности системы и основные параметры и характеристики выделенных подсистем.

Введение

Концептуальная модель является удобным инструментом для проектирования, разработки и повышения эффективности функционирования реальных систем условного доступа (СУД) [1, 2]. Разработка концептуальной модели является необходимым первоначальным этапом в проектировании и детальном исследовании систем условного доступа. Модель должна максимально полно и адекватно описывать функциональную структуру, процессы и события, свойственные системам условного доступа, охватывать все параметры и характеристики, а также обеспечивать изменение параметров для наблюдения за изменениями характеристик, что позволит установить основные закономерности функционирования системы. Актуальность темы исследования обусловлена высокими темпами развития цифровых форматов спутникового и эфирного теле- и радиовещания, а также очевидной тенденцией к внедрению на территории Российской Федерации и стран Европейского Союза телекоммуникационных систем нового поколения на базе стандарта DVB (Digital Video Broadcasting) [5–7]. Научная новизна заключается в отсутствии на текущий момент опубликованных российских исследований в этом направлении и в закрытости и недоступности зарубежных разработок, носящих ярко выраженный коммерческий характер [2].

1. Эффективность системы условного доступа

Основная задача СУД – защита от несанкционированного доступа к информационным сервисам, требующим различного качества обслуживания и степени секретности. Таким образом, в качестве основного показателя эффективности должна выступать вероятность несанкционированного доступа (НСД) к информационному сервису в течение определенного промежутка времени. Однако для достижения минимального значения этого показателя в широкополосном канале связи должен передаваться неприемлемо большой объем служебной информации, а стоимость передающей и принимающих подсистем будет превышать разумное значение. Следовательно, при моделировании системы условного доступа необходимо учитывать несколько показателей эффективности, отражающих стоимостные аспекты СУД и эффективность использования канала связи:

- 1) Вероятность НСД к информационному сервису в течении определенного промежутка времени.
- 2) Объем сервисной информации, генерируемой СУД и передаваемой в канале связи, максимальное число поддерживаемых информационных сервисов, максимальное число легальных пользователей информационного сервиса.
- 3) Стоимость установки системы, затраты на эксплуатацию системы, затраты на восстановление системы.

Структура системы условного доступа

Взаимодействие системы условного доступа с внешними сущностями можно представить в виде контекстной диаграммы (рис. 1).



Рис 1. Контекстная диаграмма

В состав системы условного доступа входят следующие взаимосвязанные подсистемы, выделенные по функциональному признаку:

1. подсистема управления пользователями и сервисами;
2. подсистема удаленного управления декодерами;
3. подсистема скремблирования и шифрования информационных сервисов;
4. подсистема генерации и управления ключами;
5. подсистема приема и декодирования информационных сервисов.

Основные взаимосвязи между этими подсистемами представлены на рис. 2.

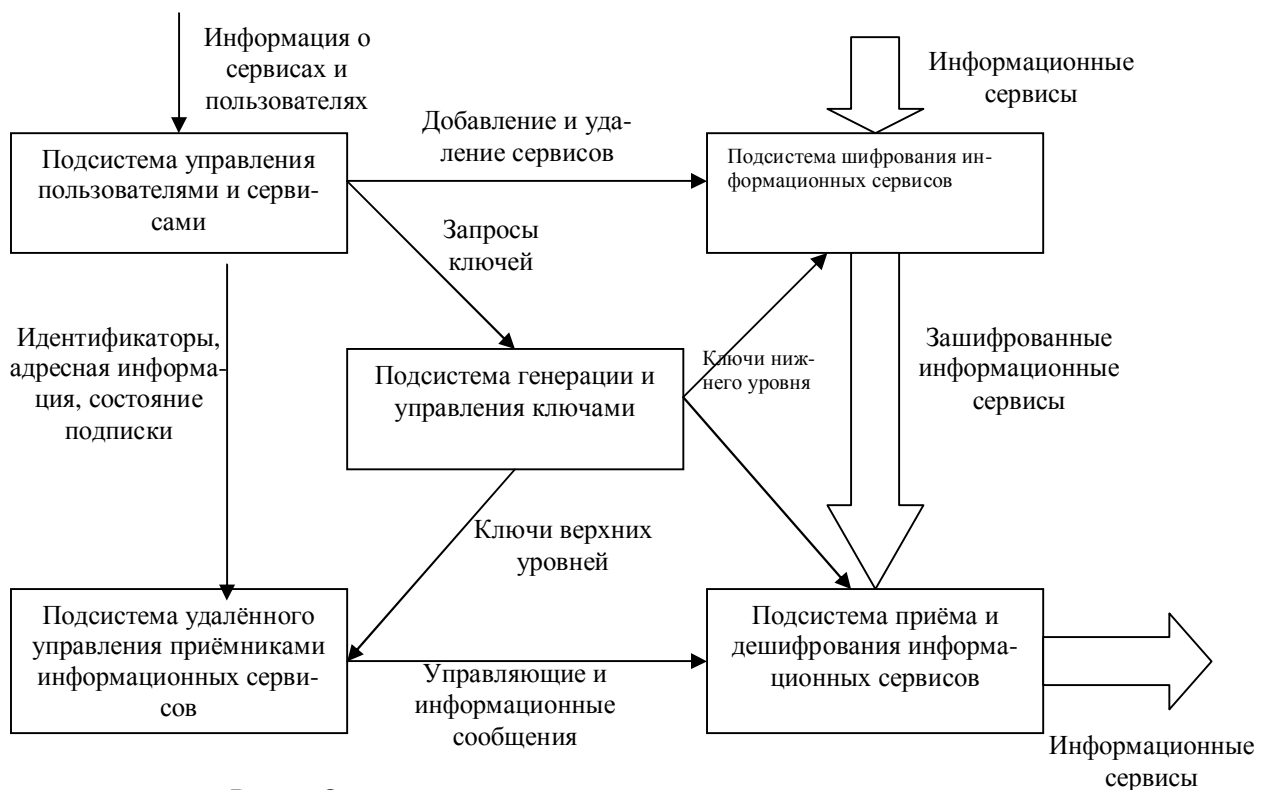


Рис. 2. Структурная схема системы условного доступа

Подсистема управления пользователями и сервисами

Эта подсистема должна обеспечивать выполнение следующих функций:

1. регистрация и удаление пользователей и информационных сервисов;
2. присвоение пользователям и сервисам уникальных идентификаторов;
3. поддержка групп пользователей и сервисов;
4. передача и регистрация секретных ключей, получаемых по запросу от системы генерации ключей, легальным пользователям;
5. определение источника утечки секретной информации на основе механизмов трассировки ключей и способов адресации конечных пользователей;
6. оперативное отключение групп и индивидуальных пользователей от доступа к определенным сервисам;
7. формирование и передача необходимой информации в систему удаленного управления декодерами.

В общем случае данную подсистему можно представить в виде матрицы прав доступа:

$$M = \begin{array}{c|cccc} & s_1 & s_2 & \cdot & s_n \\ \hline u_1 & 1 & 0 & \cdot & 0 \\ u_2 & 0 & 1 & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ u_m & 1 & 0 & \cdot & 1 \end{array},$$

где $U = \{u_1, u_2, \dots, u_m\}$ – множество пользователей системы; $S = \{s_1, s_2, \dots, s_n\}$ – множество информационных сервисов.

Для уменьшения числа столбцов матрицы M можно объединить некоторые сервисы в классы, а для уменьшения числа строк в таблице – объединить пользователей в группы. Для большей информативности матрицы M предлагается заменить 0 и 1 в узлах матрицы на временной интервал T_{ij} , отражающий время начала доступа к сервису и время окончания доступа пользователя i к сервису j .

Подсистема генерации и управления ключами

В основе функционирования данной подсистемы лежит иерархическая организация криптографических алгоритмов и ключей (рис. 3).

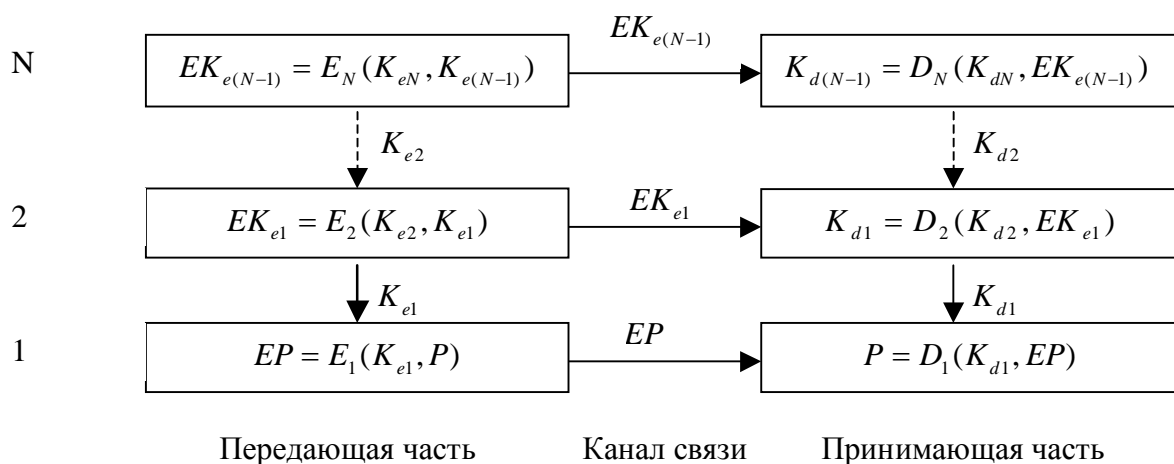


Рис. 3. Иерархическая организация криптографических алгоритмов и ключей

На самом нижнем уровне передающей части этой иерархии располагаются алгоритм шифрования (E_1) и ключ шифрования (K_{e1}) непосредственно транспортного потока (P). Соответственно, на нижнем уровне принимающей части располагаются алгоритм дешифрования (D_1) и ключ дешифрования (K_{d1}) защищенного транспортного потока (EP), передаваемого в широкополосном канале связи. Так как большинство передаваемых информационных сервисов требует обслуживания в реальном масштабе времени, то одним из основных показателей эффективности алгоритма нижнего уровня является скорость шифрования и дешифрования транспортного потока, поэтому на данном уровне предполагается использовать быстрые поточные или комбинированные блочно-поточные симметричные алгоритмы шифрования с относительно небольшой длиной ключа [3].

На верхних уровнях иерархии располагаются алгоритмы и ключи для шифрования (E_i, K_{ei}) и дешифрования (D_i, K_{di}) секретной информации, участвующей в процессе защиты транспортного потока. Требования к этим алгоритмам и ключам отличаются от требований, предъявляемых к алгоритмам и ключам нижнего уровня.

Источник	Событие	Реакция
Подсистема управления пользователями и сервисами	Добавление нового сервиса в транспортный поток	<ol style="list-style-type: none"> 1. Генерация ключа 1 уровня. 2. Пересылка ключа 1 уровня подсистеме шифрования транспортного потока. 3. Шифрование ключа 1 уровня алгоритмом E_2 и ключом K_{e2} и пересылка его подсистеме декодирования информационных сервисов. 4. Запуск таймера генерации ключа 1 уровня.
Подсистема управления пользователями и сервисами	Удаление сервиса из транспортного потока	Удаление соответствующего таймера генерации ключа 1 уровня.
Подсистема управления пользователями и сервисами	Запрос нового ключа i -го уровня.	<ol style="list-style-type: none"> 1. Генерация ключа i-го уровня. 2. Шифрование ключа i-го уровня алгоритмом E_{i+1} и ключом K_{ei+1} и пересылка его подсистеме управления пользователями и сервисами.
Подсистема управления удаленными декодерами	Запрос нового ключа i -го уровня.	<ol style="list-style-type: none"> 1. Генерация ключа i-го уровня. 2. Шифрование ключа i-го уровня алгоритмом E_{i+1} и ключом K_{ei+1} и пересылка его подсистеме управления удаленными декодерами.
Таймер	Запрос нового ключа 1 уровня для соответствующего информационного сервиса	<ol style="list-style-type: none"> 1. Генерация ключа 1 уровня. 2. Пересылка ключа 1 уровня подсистеме шифрования транспортного потока. 3. Шифрование ключа 1 уровня алгоритмом E_2 и ключом K_{e2} и пересылка его подсистеме декодирования информационных сервисов.

Таблица. Матрица событий подсистемы генерации и управления ключами

Основным показателем эффективности этих алгоритмов является криптоустойчивость, выражающаяся в вероятности получения несанкционированного доступа к секретной информации группой неавторизованных пользователей. В частности, большое значение для алгоритмов верхних уровней иерархии имеет длина ключа, которая должна быть достаточной для предотвращения атак методом полного перебора. Обратной стороной большой длины ключей является увеличение доли служебной информации в транспортном потоке при смене этих ключей. Однако чем выше уровень иерархии, тем реже должны меняться ключи. В идеальном случае ключ, находящийся на верхней ступени иерархии, вообще не должен никогда меняться, а должен находиться в максимально защищенной памяти дешифратора и использоваться только в экстренных случаях, например, для восстановления системы условного доступа или для обновления программного обеспечения принимающей части.

Кроме безопасного распространения ключей, данная подсистема должна выполнять также функцию генерации ключевой информации. Основной характеристикой подсистемы генерации ключей является функция распределения случайных чисел на выходе генератора ключей. Анализ этой характеристики позволяет сделать вывод о качестве используемого генератора случайных чисел. Другие важные характеристики это скорость выработки случайной величины и стоимость генератора.

Функционирование подсистемы генерации и управления ключами можно представить в виде матрицы событий (табл.).

Параметры:

1. число уровней иерархии алгоритмов и ключей, N ;
2. число информационных сервисов в одном транспортном потоке, K ;
3. скорость работы алгоритмов шифрования, $VE_i, 2 \leq i \leq N$ (Б/с) (алгоритм E_1 применяется только подсистемой шифрования транспортного потока);
4. скорость работы алгоритмов дешифрования, $VD_i, 2 \leq i \leq N$ (Б/с) (алгоритм D_1 применяется только подсистемой шифрования транспортного потока);
5. длины ключей шифрования, $LK_{ei}, 1 \leq i \leq N$ (Б);
6. длины ключей дешифрования, $LK_{di}, 1 \leq i \leq N$ (Б);
7. период смены ключа 1 уровня, I_1 (с);
8. скорость генерации случайных чисел, VR (Б/с).

Характеристики:

1. вероятность НСД к информации i -го уровня, $Ph_i, 2 \leq i \leq N, 0 \leq Ph_i \leq 1$ (вероятность НСД к информационным сервисам, т.е. к уровню 1, определяется подсистемой шифрования транспортного потока);
2. скорость служебной информации, передаваемой в канале, S_{si} (Б/с);
3. математическое ожидание времени нахождения заявки на генерацию ключа в очереди на обслуживание, Mw (с);
4. среднее квадратичное отклонение времени нахождения заявки на генерацию ключа в очереди на обслуживание, Dw (с);
5. время обработки заявки на генерацию ключа i -го уровня, $tp_i, 1 \leq i \leq N$ (с);
6. математическое ожидание полного времени нахождения заявки на генерацию ключа i -го уровня в подсистеме, $MQ_i, 1 \leq i \leq N$ (с);
7. производительность подсистемы, P (операций/с);
8. объем используемой оперативной памяти, R (байт).

Внешнее воздействие:

1. распределение интенсивности и тип потока поступления заявок на генерацию ключей i -го уровня.

Подсистема шифрования информационных сервисов

Основной функцией подсистемы шифрования информационных сервисов (рис. 4) является применение криптографического алгоритма 1 уровня (E_1) только к информационным сервисам, определенным подсистемой управления пользователями и сервисами. Каждому сервису должен соответствовать отдельный ключ шифрования (K_{e1}), предоставляемый подсистемой генерации и управления ключами и обновляющийся с определенным периодом (I_1).

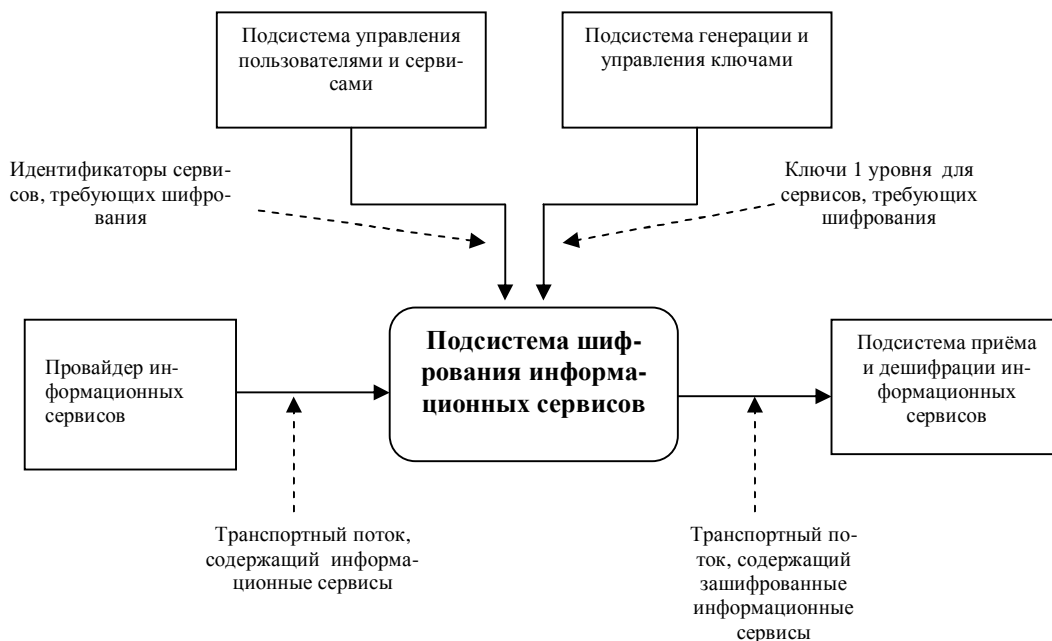


Рис. 4. Контекстная диаграмма подсистемы шифрования информационных сервисов

Функционирование подсистемы шифрования информационных сервисов можно представить в виде диаграммы состояний (рис. 5).



Рис. 5. Диаграмма состояний подсистемы шифрования информационных сервисов

Все идентификаторы информационных сервисов (для DVB-совместимых систем это значения PID [4, 6]), требующих шифрования, а также соответствующие этим идентификаторам актуальные ключи 1 уровня хранятся в специализированной таблице. Добавление нового сервиса выглядит как добавление новой строки в эту таблицу. Обновление ключа заключается в изменении информации в соответствующем поле. При принятии нового транспортного пакета идентификатор, содержащийся в заголовке пакета, последовательно сравнивается со всеми значениями PID в таблице. Если находится строка с таким же идентификатором, то из нее выбирается ключ 1 уровня, который используется для шифрования полезной нагрузки транспортного пакета. Если строка не найдена, то пакет передается в канал связи без шифрования.

Параметры:

1. скорость работы алгоритма шифрования, VE_1 (Б/с);
2. скорость работы алгоритма дешифрования, VD_1 (Б/с);
3. время обработки сообщения от подсистемы управления пользователями и сервисами, $T_{process1}$ (с);
4. время обработки сообщения от подсистемы генерации ключей, $T_{process2}$ (с);
5. длина ключа шифрования, LK_{e1} (Б);
6. длина ключа дешифрования, LK_{d1} (Б);
7. число различных информационных сервисов, требующих шифрования в одном транспортном потоке, K ;
8. размер полезной нагрузки транспортного пакета, LP_l (Б);
9. размер блока данных для алгоритма шифрования, LB (Б);
10. период смены ключа 1 уровня, I_1 (с).

Характеристики:

1. вероятность НСД к информации 1-го уровня, т.е. непосредственно к информационному сервису, $Ph_1, 0 \leq Ph_1 \leq 1$;
2. математическое ожидание полного времени нахождения транспортного пакета, требующего шифрования на генерацию ключа в подсистеме, MTP (с);
3. производительность подсистемы, P (операций/с);
4. объем используемой оперативной памяти, R (байт).

Внешнее воздействие:

1. распределение интенсивности поступления и тип потока транспортных пакетов, требующих шифрования, на вход подсистемы;
2. распределение интенсивности поступления и тип потока сообщений от подсистемы управления пользователями и сервисами;
3. распределение интенсивности поступления и тип потока сообщений от подсистемы генерации ключей.

Подсистема приема и дешифрации информационных сервисов

Одним из основных компонентов технических средств является декодер системы условного доступа, который может представлять собой программную эмуляцию, аппаратную реализацию, программную эмуляцию с поддержкой смарт-карт, дополнительный декодер, устанавливаемый в специальный разъем интегрированного приемника.

В зависимости от реализации декодер может выполнять алгоритмы дешифрования либо аппаратными, либо программными средствами, может содержать в своей памяти всю информацию, необходимую для декодирования сервисов, включая секретные ключи, либо содержать только алгоритм дескремблирования и протокол обмена со смарт-картой, в памяти которой находится секретная информация.

Схема процесса декодирования, общая для всех реализаций, приведена на рис. 6.

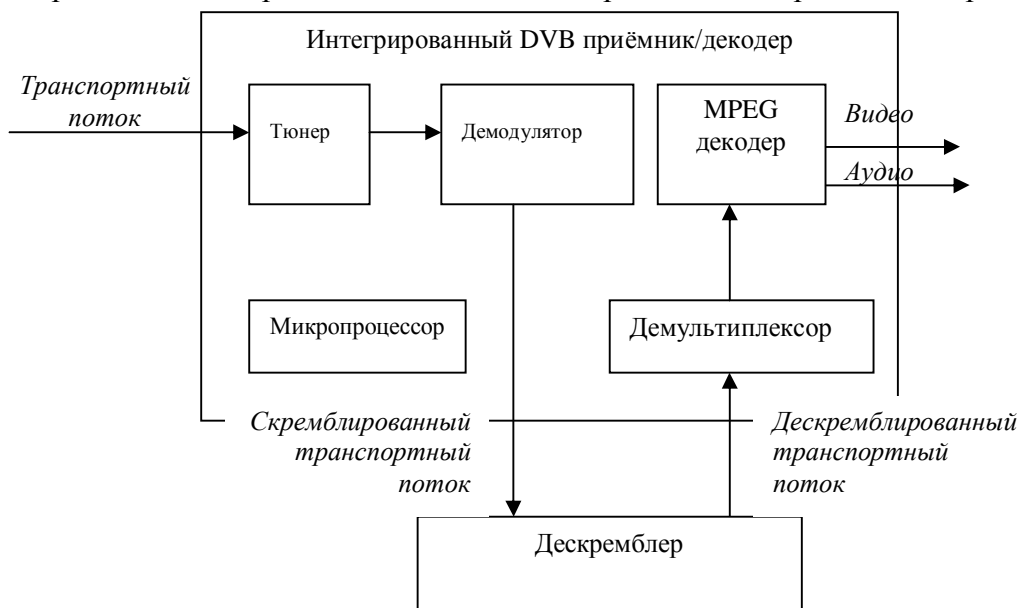


Рис. 6. Схема процесса декодирования, общая для всех реализаций

Каждый из вариантов имеет специфические особенности, позволяющие наиболее эффективно и гибко построить декодирующую часть системы условного доступа. Так, аппаратная реализация является наиболее быстрым и надежным способом декодирования, однако приемник с аппаратным декодером будет специализирован под конкретный алгоритм кодирования, что существенно уменьшает функциональность приемника и привязывает его к нескольким провайдерам, использующим эту кодировку. В случае взлома такой системы процедура обновления или восстановления декодирующей части системы условного доступа будет заключаться в замене аппаратных составляющих, что в большинстве случаев либо невозможно, либо связано с большими финансовыми затратами.

Наиболее гибкой и надежной является комбинированная программно-аппаратная реализация, основанная, например, на COMMON INTERFACE (CI) – технологии, широко поддерживаемой разработчиками DVB оборудования. Доступ к кодированным сервисам осуществляется с использованием дополнительного декодера, устанавливаемого в предусмотренный для этой цели в приемнике разъем [1].

Параметры:

1. число уровней иерархии алгоритмов и ключей, N ;
2. число команд удаленного управления приемником, K ;
3. скорость работы алгоритмов дешифрования, $VD_i, 1 \leq i \leq N$ (Б/с);
4. длины ключей дешифрования, $LK_{di}, 1 \leq i \leq N$ (Б);
5. период смены ключа i -го уровня, $I_i, 1 \leq i \leq N$ (с).

Характеристики:

1. математическое ожидание полного времени нахождения транспортного пакета, требующего расшифрования в подсистеме, MTP (с);
2. математическое ожидание полного времени обработки команды удаленного управления, $MTK_i, 1 \leq i \leq K$ (с);
3. производительность подсистемы, P (операций/с);
4. объем используемой оперативной памяти, R (байт).

Внешнее воздействие:

1. распределение интенсивности поступления и тип потока транспортных пакетов, требующих дешифрования, на вход подсистемы;
2. распределение интенсивности поступления и тип потока сообщений от подсистемы удаленного управления приемниками;
3. распределение интенсивности поступления и тип потока сообщений от подсистемы генерации ключей.

Заключение

Результатом работы является концептуальная модель СУД, содержащая ряд взаимосвязанных подсистем. В работе определены принципы функционирования СУД, выявлены основные взаимосвязи между подсистемами, сформулированы параметры и характеристики подсистем, определены основные показатели эффективности СУД. Предложенная концептуальная модель может быть основой для проектирования реальных систем, а также удобным инструментом для проведения детального исследования СУД.

Литература

1. Костин М.В. Системы условного доступа // Телеспутник. 2004. №12, С. 62–64.
2. McCormac J. European Scrambling Systems. Waterford University Press. 1996.
3. Schneier B. Applied Cryptography, John Wiley & Sons, 1995.
4. ISO/IEC 13818-1:1994 Information technology – Generic coding of moving pictures and associated audio information - Part 1 Systems, ISO, 1994.
5. ETSI EN 300 421 v1.1.2:1997 Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services, ETSI, 1997.
6. ETSI EN 300 468 v1.4.1:2000 Digital Video Broadcasting (DVB); Specification for Services Information (SI) in DVB systems, ETSI, 2000.
7. ETR 289:1996 Digital Video Broadcasting (DVB); Support for use of scrambling and Conditional Access (CA), ETSI, 1996.

КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ СИСТЕМЫ ПЕРЕДАЧИ И ПРИЕМА ДАННЫХ

А.В. Костина

Научный руководитель – д.т.н., профессор Т.И. Алиев

В статье рассматривается концептуальная модель системы передачи и приема сервисной информации, передаваемой в спутниковом широкополосном канале связи, кроме того, производится ее декомпозиция, определяются параметры и характеристики каждой части.

Введение

Основной целью данной работы является повышение эффективности использования цифрового канала связи за счет увеличения числа дополнительных сервисов, передаваемых в транспортном потоке стандарта DVB. На настоящий момент пропускная способность подавляющего большинства спутниковых каналов связи не используется полностью, и в результате часть полосы пропускания, которая могла бы быть использована для передачи данных, теряется. Очевидно, что сумма скоростей всех элементарных потоков не должна превышать скорости транспортного потока, так как существует вероятность потери информации, что приводит к невозможности декодирования базовых сервисов на приемной стороне. В большинстве случаев размер свободной части полосы пропускания не позволяет использовать ее для передачи видеоданных, однако его может быть вполне достаточно для передачи различных сервисных данных, служебной информации провайдеров, операторов систем условного доступа или производителей прямо-передающего оборудования.

Таким образом, система приема и передачи данных предназначена для предоставления конечным пользователям разнородной информации, инкапсулируемой в реальный транспортный поток, без негативного влияния на передачу базовых сервисов. Функционирование системы должно отвечать следующим требованиям:

- обеспечивать высокую эффективность использования пропускной способности цифрового канала связи;
- гарантировать всем базовым сервисам удовлетворение требований к полосе пропускания и качеству обслуживания;
- обеспечивать оптимальную реализацию передачи дополнительных сервисов различных типов, удовлетворяющих заданным ограничениям.

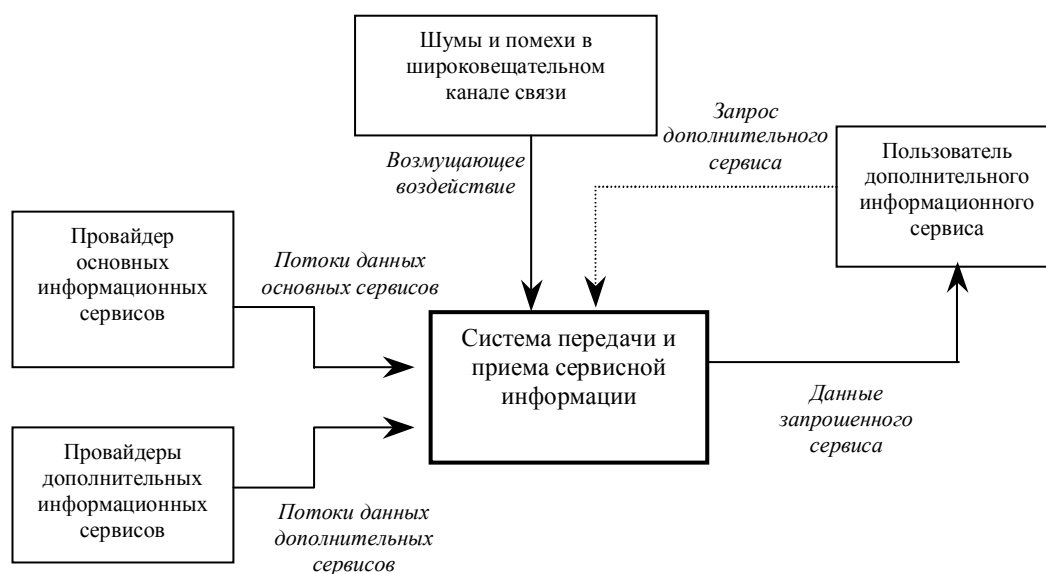


Рис. 1. Контекстная диаграмма системы передачи и приема сервисной информации

Контекстная диаграмма (рис. 1) обеспечивает наиболее общее описание системы передачи и приема сервисной информации. На диаграмме отображена связь системы с внешней средой. По результатам разработки контекстной диаграммы выполняется процесс декомпозиции модели.

Таким образом, можно выделить следующие совокупные компоненты системы передачи и приема данных (рис. 2):

- передающая часть;
- цифровой широкополосный однонаправленный канал связи;
- принимающая часть (интегрированный приемник-декодер).

Каждая компонента отражает характерные особенности системы, принципы и зависимости, с помощью которых описывается поведение системы в целом.

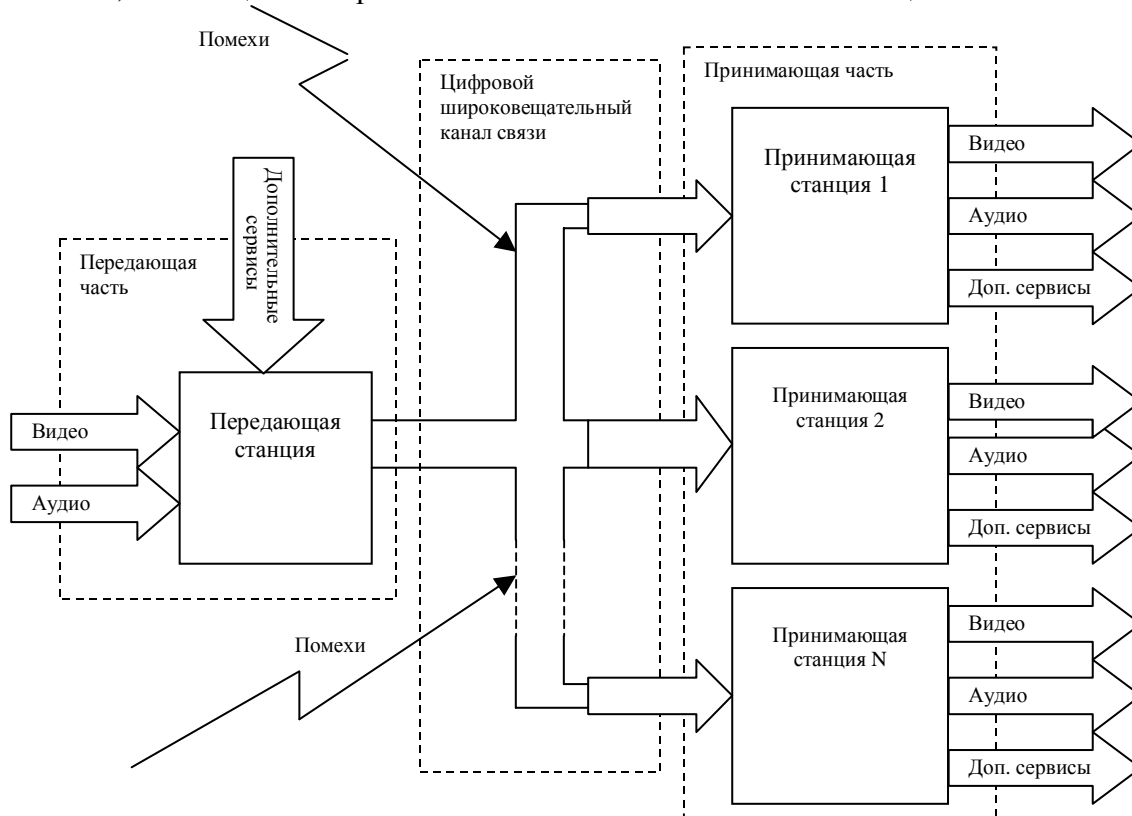


Рис. 2. Компоненты системы приема и передачи сервисной информации

Передаточная часть

Одной из основных задач передающей части является задача вычисления свободной части ширины полосы пропускания спутникового канала связи, так как именно наличие и размер свободной части полосы пропускания являются определяющим моментом при выборе спутникового канала для передачи разнородной информации. Необходимо отметить, что метод вычисления размера свободной части во многом будет определяться способом разделения цифрового спутникового канала связи между различными сервисами, из которых следует выделить:

- статическое мультиплексирование с разделением времени, используется в случаях, когда скорости всех элементарных потоков, формирующих транспортный поток, являются постоянными величинами и их значения известны до начала процесса мультиплексирования;
- динамическое (статистическое) мультиплексирование с разделением времени, отличительная черта которого заключается в прогнозировании и предсказании загрузки канала связи в режиме реального времени на основе собираемой статистической

информации и динамического анализа свойств элементарных потоков при условии достижения требований к обеспечению заданного уровня качества.

Очевидно, что наиболее эффективно полоса пропускания цифрового канала связи используется при динамическом мультиплексировании. Кроме того, необходимо решить задачу, связанную с управлением и контролем дополнительных сервисов, так как разнородность сервисов обуславливает различные требования к механизму их передачи в едином транспортном потоке.

На входы передающей системы (рис. 3) от внешних источников с различной интенсивностью поступают элементарные потоки данных, например, аудио- и видеоинформация, кодированная в соответствии со стандартом MPEG-2 [2]. Поскольку в каждый момент времени на разные входы системы могут одновременно поступать несколько элементарных потоков, а обрабатываться в мультиплексоре в каждый момент времени может только один пакет данных, принадлежащий определенному элементарному потоку, то на входе мультиплексора могут образовываться очереди из данных, требующих обработки. Число очередей в предельном случае может быть равно числу входов мультиплексора. Размеры очередей ограничиваются размером буферной памяти. Алгоритм функционирования мультиплексора заключается в последовательном опросе всех входных буферов через определенный интервал, зависящий от производительности мультиплексора и скорости выходного транспортного потока. Если в опрашиваемом в текущий момент буфере находятся данные, то из них выделяется блок, размер которого соответствует размеру минимальной единицы оперирования мультиплексора (для DVB-совместимых систем это размер полезной нагрузки транспортного пакета – 184 байта [2]). К этому блоку добавляется служебная информация – заголовок и концевик, а также вычисляется контрольная сумма блока или CRC, которые добавляются к формируемому пакету. Полученный транспортный пакет передается на дальнейшую обработку в модулятор, где он подвергается различной обработке с целью увеличения помехозащищенности, в частности, могут использоваться алгоритмы внутреннего помехоустойчивого кодирования (Viterby, FEC) или внешнего (коды Рида-Соломона). В результате обработки пакет содержит избыточные данные, необходимые для восстановления информации в случае ее искажения в канале связи. Таким образом, первоначальный размер пакета может быть существенно увеличен. Следующая стадия – это модуляция и трансляция пакета в широковещательный канал связи. Если во время очередного опроса входных буферов мультиплексор не обнаружил ни одного блока данных, то он должен сформировать пустой пакет, отличающийся определенным идентификатором и отсутствием каких-либо данных в поле полезной нагрузки. Система передачи сервисной информации должна организовать управление дополнительными информационными сервисами, которые в зависимости от типа могут поступать на один из входов мультиплексора в режиме реального времени или находиться в статической базе данных или файловой системе.



Рис. 3. Функциональная структура передающей части

Параметрами передающей части являются:

- 1) количество входящих элементарных потоков, N ;
- 2) средние скорости входящих элементарных потоков, λ_i (байт/с), $1 \leq i \leq N$;
- 3) скорость выходящего транспортного потока, λ_T (байт/с) ;
- 4) размер буферной памяти, S (байт);
- 5) количество файлов дополнительных информационных сервисов, M ;
- 6) размер файлов дополнительных информационных сервисов, s_j (байт), $1 \leq j \leq M$;
- 7) размер секций файлов дополнительных информационных сервисов, c_j (байт), $1 \leq j \leq M$;
- 8) множество возможных приоритетов дополнительных информационных сервисов, $P, P = \{1, 2, 3, \dots\}$;
- 9) приоритеты файлов дополнительных информационных сервисов, $p_j, 1 \leq j \leq M, p_j \in P$;
- 10) объем сервисной информации, добавляемой к одной секции файла (заголовки, концевики, служебные поля, контрольные суммы и др.), C_H (байт);
- 11) коэффициент избыточной информации, необходимой для достижения необходимой помехоустойчивости, FEC.

Характеристики передающей части:

- 1) размер части транспортного потока, не используемой базовыми сервисами (видео, аудио), F (байт/с);
- 2) объем дополнительных информационных сервисов, требующих передачи S_{as} (байт), $S_{as} = \sum_{j=1}^M s_j$;
- 3) математическое ожидание объема информации, требующей передачи, во входных буферах памяти, s_{mi} (байт), $1 \leq i \leq N$;
- 4) среднеквадратическое отклонение объема информации, требующей передачи, во входных буферах памяти, s_{di} (байт), $1 \leq i \leq N$;
- 5) математическое ожидание времени передачи одной секции файлов дополнительных информационных сервисов, $t_{mj}(c)$, $1 \leq j \leq M$;
- 6) среднеквадратическое отклонение времени передачи одной секции файлов дополнительных информационных сервисов, $t_{dj}(c)$, $1 \leq j \leq M$;
- 7) математическое. ожидание скорости передачи файлов дополнительных информационных сервисов, v_{mj} (байт/с), $1 \leq j \leq M$;
- 8) среднеквадратическое отклонение скорости передачи файлов дополнительных информационных сервисов, v_{dj} (байт/с), $1 \leq j \leq M$;
- 9) коэффициент полезной загрузки канала связи, K .

Цифровой широкополосный однонаправленный канал связи

Одной из важных характеристик канала связи является пропускная способность канала, которая по теореме Шеннона ограничивается заданной полосой пропускания S и соотношением сигнал/шум P_s / P_n .

В реальных каналах действуют шумы, искажения сигналов, помехи, интерференция, которые проявляются в виде ошибок в передаваемой информации и, следовательно, влияют на достоверность приема данных. Одной из наиболее часто используемых характеристик, которой принято оценивать качество канала связи, является коэффициент двоичных ошибок (BER), который представляет собой отношение между ошибочно

принятыми битами к общему количеству переданных бит. Так как транспортный поток стандарта DVB ориентирован на пакетную передачу, то в качестве характеристики, оценивающей качество передачи пакетов, будем использовать вероятность приема пакета с ошибками.

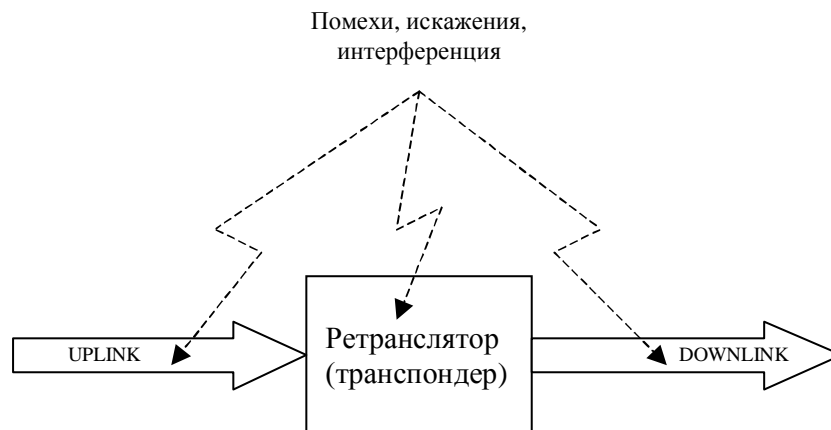


Рис. 4. Спутниковый канал связи

Параметры канала связи:

- 1) ширина полосы пропускания, F (Гц);
- 2) мощность сигнала, P_s (дБ);
- 3) мощность шума, P_n (дБ).

Характеристики канала связи:

- 1) пропускная способность, C (бит/с);
- 2) вероятность искажения одного бита передаваемой информации, вследствие влияния внешних воздействий (погодные условия, интерференция, и др.), $BERR, 0 \leq BERR \leq 1$;
- 3) вероятность приема пакета с ошибками.

Принимающая часть представляет собою интегрированный приемник-декодер (рис. 5).

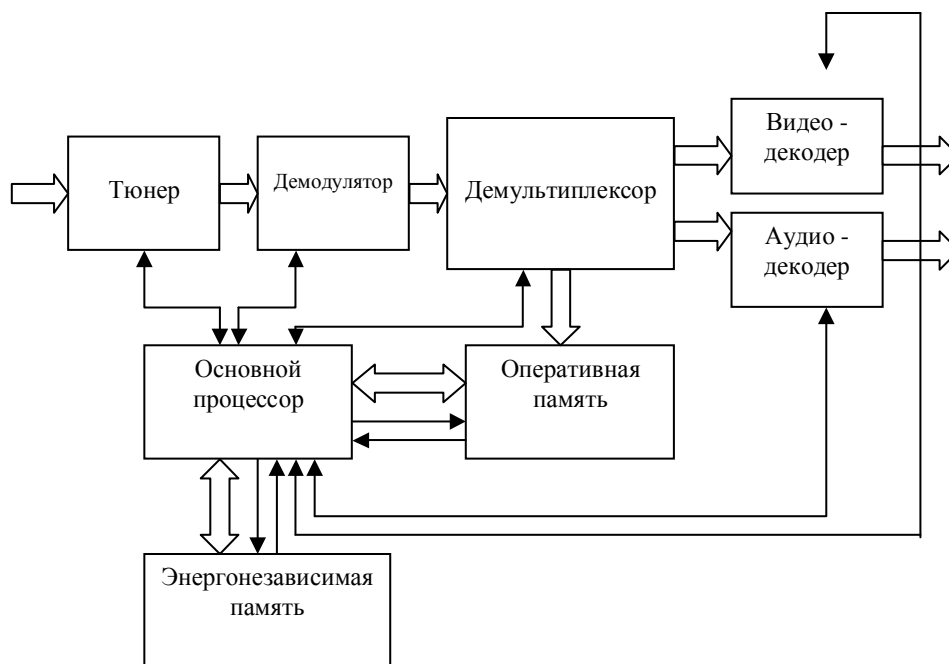


Рис. 5. Функциональная структура принимающей части

Алгоритм, выполняющийся на принимающей стороне, обеспечивает фильтрацию входящей информации и ассемблирование ее во внутренних буферах с целью последующей обработки (рис. 5). Необходимо обратить внимание на аппаратные особенности принимающего терминала, одной из которых является производительность основного процессора принимающего терминала (F , инструкций/с), определяющая время обработки принятой секции ($T_{process}$, с). Это время не должно превышать интервал между передачей секций (T_{int} , с: $T_{int} > T_{process}$), иначе произойдет существенное увеличение общего времени приема целого файла.

В общем случае функционирование процесса приема сервисной информации абонентским терминалом можно представить в виде диаграммы состояний (рис. 6).

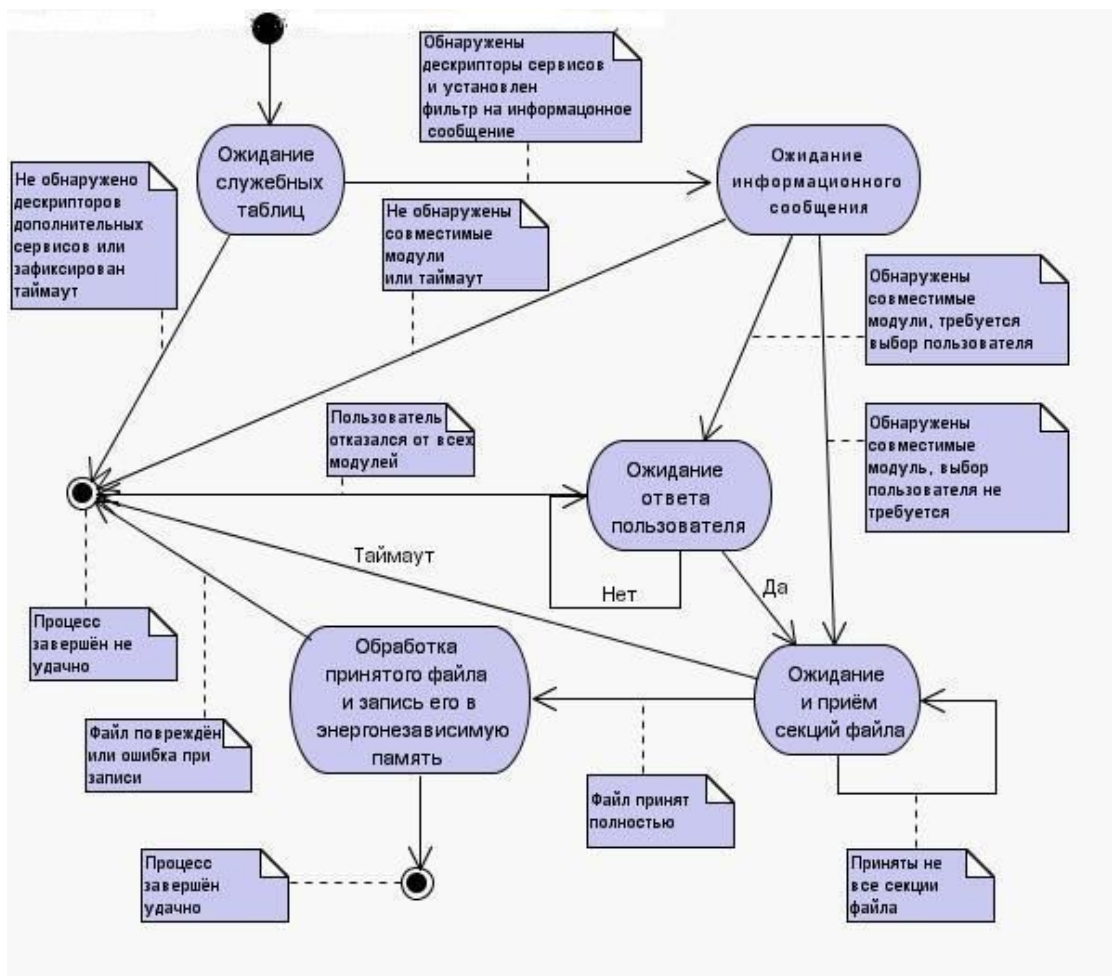


Рис. 6. Диаграмма состояний принимающей части

Параметры принимающей части:

- 1) аппаратные возможности принимающего терминала;
- 2) математическое ожидание времени передачи одной секции файла дополнительных информационных сервисов $t_{mj}(c), 1 \leq j \leq M$;
- 3) среднее квадратическое отклонение времени передачи одной секции файлов дополнительных информационных сервисов $t_{dj}(c), 1 \leq j \leq M$;
- 4) размер файлов дополнительных информационных сервисов s_j (байт), $1 \leq j \leq M$;
- 5) размер секций файлов дополнительных информационных сервисов c_j (байт), $1 \leq j \leq M$;

Характеристики:

- 1) математическое ожидание полного времени приема одного файла фиксированного размера (T_T, c);
- 2) среднеквадратическое отклонение полного времени приема одного файла фиксированного размера (T_T, c);
- 3) мат. ожидание времени ожидания начала передачи файла принимающей станцией (T_w, c);
- 4) среднеквадратическое отклонение времени ожидания начала передачи файла принимающей станцией (T_w, c);
- 5) вероятность потери секции файла (P_l , где ($0 \leq P_l \leq 1$)).

Показатели эффективности системы приема и передачи сервисной информации:

- коэффициент загрузки канала связи;
- полное время приема одного файла фиксированного размера абонентским терминалом с минимальными аппаратными возможностями.

Заключение

Таким образом, в данной статье представлена концептуальная модель системы передачи и приема информации. Проведена декомпозиция, определены основные функции каждой части. Для каждой совокупной компоненты системы рассмотрены ее параметры и характеристики.

Литература

1. Haskell B. G., Puri A., Netravali A. N. Digital Video. An Introduction Video. An Introduction to MPEG-2. Chapman and Hall, 1997.
2. 13818-1 Coding of Moving Pictures and Associated Audio – Part 1: Systems (MPEG –2). ISO, 1996.
3. Стивенсон Д. Спутниковое ТВ. М.: ДМК Пресс, 2001.
4. ETSI EN 300 468 v1.4.1:2000 Digital Video Broadcasting (DVB); Specification for Services Information (SI) in DVB systems. ETSI, 2000.

АНАЛИЗ ПРОЦЕССОВ В ОПЕРАЦИОННОЙ СИСТЕМЕ РЕАЛЬНОГО ВРЕМЕНИ

Чэнь Жуймин

Научный руководитель – к.т.н, доцент А.О. Ключев

В работе обсуждаются проблемы, которые возникают при анализах процессов в операционной системе реального времени, и их симуляция на языке С.

Введение

Системой реального времени является такая система, корректность функционирования которой определяется не только корректностью выполнения вычислений, но и временем, в которое получен требуемый результат. Если требования по времени не выполняются, то считается, что произошел отказ системы [1]. Чтобы система могла удовлетворить требованиям, предъявляемым к системам реального времени, аппаратные, программные средства и алгоритмы работы системы должны гарантировать заданные временные параметры реакции системы. Время реакции не обязательно должно быть очень маленьким, но оно должно быть гарантированным (и отвечающим поставленным требованиям). В области исследования систем реального времени много задач. Одной из них является планирование процессов в операционных системах реального времени.

Планирование периодических задач на одном процессоре – классическая проблема планирования в системах реального времени. Имеются два альтернативных подхода к решению этой проблемы, основанные на фиксированном или динамическом назначении приоритета каждой задачи. В первом приоритет каждой задачи вычислен однажды и остается неизменным в течение жизни задачи. В динамическом подходе (также называемом «управление крайними сроками») приоритет вычисляется динамически и может быть изменен во время выполнения. Эти подходы привели к развитию разнообразных вытесняющих политик планирования (вытесняющая политика подразумевает возможность прерывания обработки задачи, если поступил запрос задачи с более высоким приоритетом). Они включают «монотонизирующую норму» (RM), «самый ранний крайний срок сначала» (earliest deadline first, EDF), «слабая задача вперед» (Least laxity first, LLF), «максимальная безотлагательность сначала» (maximum-urgency-first, MUF).

Исследование алгоритмов

Алгоритм планирования RM изобретен в 1973 г. Liu и Leyland [2]. Он основан на принципе «короткий период – вперед». RM считается стабильным. Максимальная загрузка (U) в пределе достигает 69%. Аналитическая модель RM в настоящее время расширена для анализа зависимых и аperiodических задач. Для обеспечения стабильности работы системы должно быть выполнено соотношение

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \dots + \frac{C_n}{T_n} \leq n * (2^{1/n} - 1) . \quad (1)$$

Результаты загрузки представлены в табл. 1.

n	U
2	0.82
3	0.75
4	0.72
∞	0.69

Таблица 1. Зависимость от соотношения (1)

Алгоритм планирования EDF основан на принципе «самый ранний крайний срок – вперед» [3]. Алгоритм является динамическим, так как имеется необходимость постоянного вычисления времени до наступления крайнего срока. Достоинство алгоритма – в минимальном количестве переключений контекстов. Формальным условием выполнения крайних сроков является $U < 100\%$. Алгоритм считается нестабильным. Теория алгоритма EDF отражен на рис. 1.

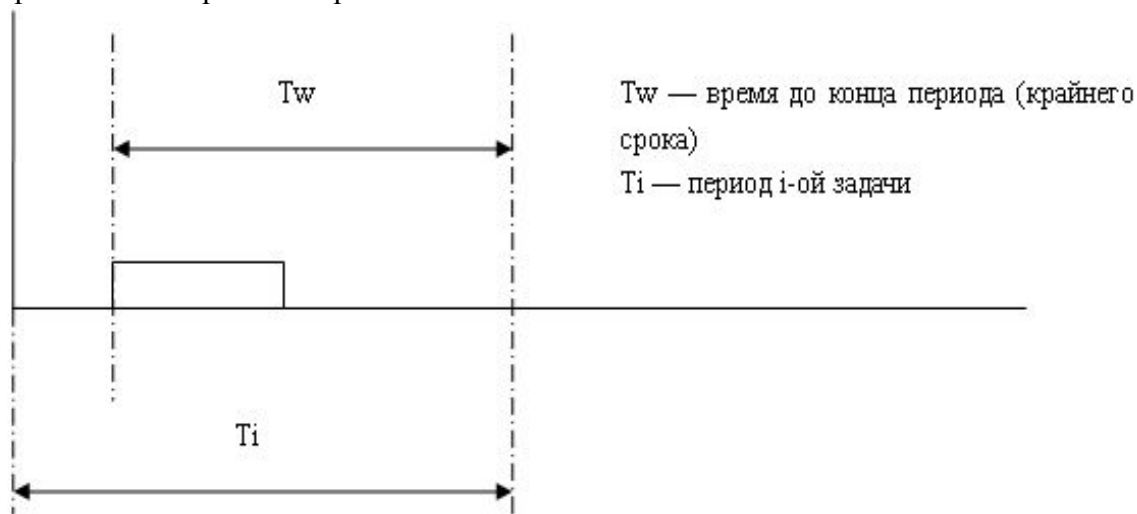


Рис. 1. Теория алгоритма EDF

Алгоритм LLF является динамическим [3], так как есть необходимость постоянно вычисления времени до наступления крайнего срока. Достоинство алгоритма – в минимальном количестве переключений контекстов. Формальным условием выполнения крайних сроков является $U < 100\%$. Алгоритм считается нестабильным. Теория алгоритма LLF отражен на рис. 2.

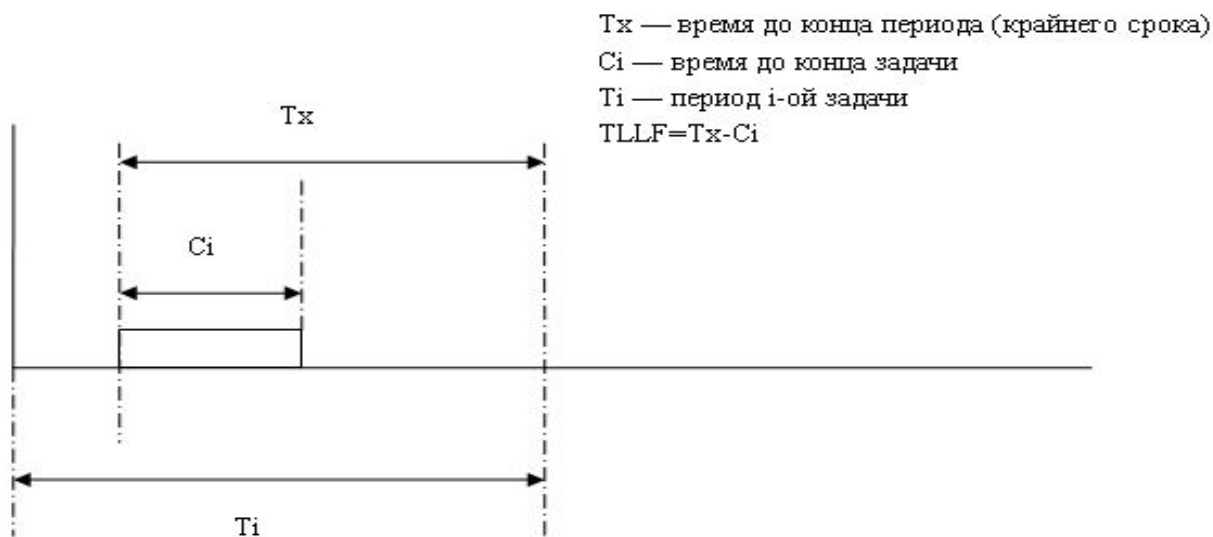


Рис. 2. Теория алгоритма LLF

Алгоритм MUF разработан в университете Carnegie Mellon, USA [4]. Это алгоритм основан на принципе «максимальная безотлагательность – сначала». Пользователь на основе экспертных оценок определяет важность задачи, разбивая задачи на две категории – важные и обычные. В процессе работы алгоритма производится сортировка задач по периоду. Первые n критичных задач объединяются во множество с загрузкой менее 100%. Первой всегда выполняется задача с минимальной слабостью и макси-

мальной важностью. Если никакой «важной» задачи нет, то выполняются обычные задачи по принципу LLF. Алгоритм является гибридным, статическим и динамическим одновременно. Период задачи может меняться, если C_i / T_i останется неизменным. Алгоритм предсказуем. Формальным условием выполнения крайних сроков является $U < 100\%$. Алгоритм считается стабильным.

Симуляторы алгоритмов планирования

В этих симуляторах у каждого процесса есть несколько соглашений [3]:

- 1) процесс выполняется периодически;
- 2) процесс должен закончиться до наступления конца периода;
- 3) процесс не должна синхронизироваться с другими задачами;
- 4) процесс не приостанавливается самостоятельно;
- 5) процесс всегда готов к исполнению.

Пример конструкции планирования симуляторов RM приведен ниже.

<pre>int add_process(char * name, int Ti, int Ci) name—имя процесса Ti—период процесса Ci—длительность выполнения процесса Добавляет процесс с именем name в таблицу процессов.</pre>
<pre>void type_process(void) распечатывает таблицу процессов</pre>
<pre>Void RM(int n) Симулятор rm. n—число квантов времени симуляции.</pre>
<pre>int main (void) { add_process("P0",6,2); add_process("P1",8,2); add_process("P2",12,3); type_process(); rm(30); }</pre>

Схема планирования задач, использующая четыре алгоритма

На этой схеме рассмотрено планирование с использованием трех процессов по описанным выше алгоритмам. Процессы выполняются целиком в каждом периоде. На схеме видно, что через каждые 24 числа квантов времени симуляции планирование повторяется. Параметры процессов определены в табл. 2 и табл. 3. Результаты алгоритмов представлены на рис. 3–6.

Имя	Период T_i	Длительность выполнения C_i	C_i/T_i
А	6	2	33.3%
В	8	2	25.0%
С	12	3	25.0%

Таблица 2. Параметры периода и длительности выполнения процессов

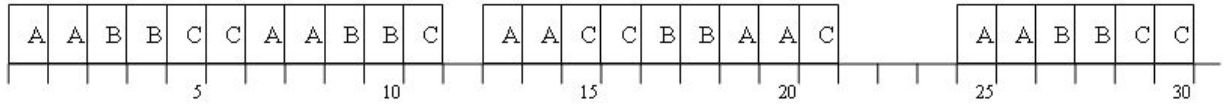


Рис. 3. Результат алгоритм RM

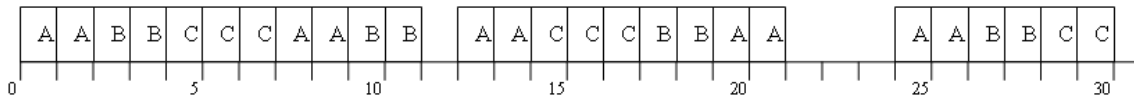


Рис. 4. Результат алгоритма EDF

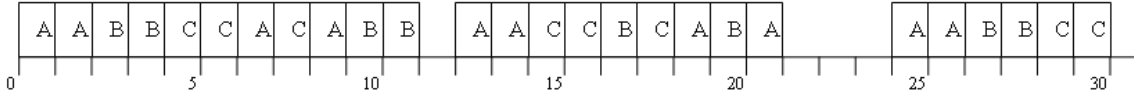


Рис. 5. Результат алгоритма LLF

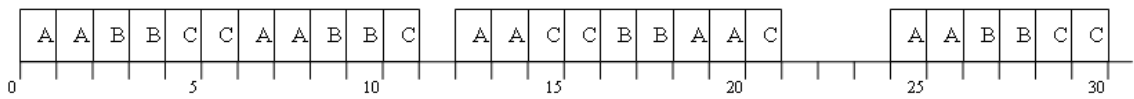


Рис. 6. Результат алгоритма MUF

Имя	Безотлагательность
А	высокая
В	высокая
С	низкая

Таблица 3. Параметры безотлагательности процессов

Заключение

В этой работе рассмотрены несколько технических решений в области реализации алгоритмов планирования, наиболее часто используемых в настоящее время, и описаны четыре примера симуляции. Выбор алгоритма планирования процессов зависит от большего количества факторов.

Литература

1. Панзиери, Даволи. Системы реального времени. / Техническое описание UBLCS-93-22. Октябрь, 1993. http://embedded.ifmo.ru/embedded_old/ETC/REFERAT/1998_1/RTSYST/abi3.htm
2. С.Liu J.Layland. Scheduling Algorithms for Multiprogramming in Hard Real Time Environment. // JACM, 20(1), 40-61 (1973).
3. Alberto Daniel Ferrari, Real-Time Scheduling Algorithms Achieving predictability for critical applications // Dr. Dobb's Journal, 1994.
4. David B. Stewart and Pradeep K. Khosla, Real-Time Scheduling of Sensor-Based Control Systems. // The Eighth IEEE Workshop on Real-Time Operatings Systems and Software, in conjunction with 17th IFAC/IFIP Workshop on Real-Time Programming, Atlanta, GA, pp. 144-150, May 1991.

ОБЗОР ОТЕЧЕСТВЕННЫХ СИСТЕМ МОДЕЛИРОВАНИЯ УПРАВЛЕНИЯ ПРЕДПРИЯТИЯМИ

А.Е. Лашкевич

Научный руководитель – к.т.н., доцент Т.А. Павловская

В статье приводится обзор современных отечественных систем моделирования управления предприятиями. Описывается класс программных продуктов ORGWARE, к которому они относятся, его назначение, место, занимаемое данным классом среди существующих средств автоматизации, взаимодействие с программными продуктами смежных классов, таких как workflow и ERP. Описываются функциональные возможности, особенности и преимущества конкретных реализаций систем моделирования, созданных отечественными производителями.

Введение

В современных условиях управление предприятием осуществляется на основе организационной структуры, которая определяется его руководством самостоятельно. Структуры управления и их подразделения определяются типом предприятия, его масштабом и другими характеристиками. Однако сложность этих структур не всегда напрямую зависит от размера предприятия, так как большое влияние оказывают история развития, используемые технологии, оборудование, средства автоматизации и т.д., поэтому структуры управления могут отличаться даже на однотипных предприятиях.

Согласно ИСО/ОМПС 9000:2000, «организация» определяется как «группа работников и необходимых средств с распределением ответственности, полномочий и взаимоотношений». В теории существует много разновидностей организационных структур. Но каким бы ни было управление, оно должно быть структурированным и регламентированным. Если управление на предприятии не организовано, это означает, что работа на предприятии нерациональна. В такой ситуации одна и та же задача может решаться многократно, а решения, принятые наугад, влекут за собой ненужные затраты временных и прочих ресурсов.

Организация – понятие многомерное. Для грамотной координации необходимо правильно представлять ее текущее состояние. Управленческие решения принимаются на основе множества факторов. Следовательно, для поддержки принятия управленческих решений эти факторы необходимо структурировать, а также задать между ними ассоциации, или связи. Если представлять организацию в виде набора взаимосвязанных структур, то, управляя такими структурами, мы управляем самой организацией.

Современный бизнес становится более динамичным. Потребитель диктует, что и в каком виде он хочет получить и по какой цене. Конкуренция на рынке в виду его глобализации резко обостряется. Потребности потребителей и ситуация на рынке меняются с все возрастающей скоростью. В результате руководство предприятия рано или поздно вынуждено радикально пересматривать свой бизнес. Компании вынуждены становиться более гибкими, постоянно реагировать на изменения внешней среды, выработать и реализовывать локальные и глобальные проекты реструктуризации своей деятельности, которые должны протекать без потери контроля над организацией [1].

Для решения задачи построения (реструктуризации) бизнеса необходимы:

- команда собственных сотрудников, обладающих всей необходимой информацией;
- внешние консультанты, имеющие свои практические наработки, навыки и умения;
- инструмент для структуризации отдельных областей управления и модели всего предприятия в целом.

ORGWARE – новый класс продуктов моделирования организационных структур управления предприятиями

Существует целый класс программных продуктов (так называемые «организационные продукты»), которые ускоряют процессы упорядочения деятельности, поддер-

живают основные процедуры построения организационных и прочих структур и создания внутрифирменных регламентных документов и положений. В России принято относить подобные продукты к классу ORGWARE [3].

Впервые подобные средства, способные помочь решению задач, связанных с организационными проблемами управления предприятием, появились на Западе в начале 90-х гг. прошлого века. Первоначально это были специализированные модули в составе мощных систем корпоративного управления (SAP/R3, SOCAP, BAAN), задачей которых было формальное описание предприятий, подлежащих автоматизации. Собственно, название этого класса происходит от названия модуля ERP-системы BAAN – BAAN-Orgware, задачей которого было формальное описание предприятия, подлежащего автоматизации с помощью системы BAAN4. Информация, полученная в результате применения этих средств, служила связующей основой для дальнейшей работы всех функциональных приложений корпоративной информационной системы.

Новые программы были ориентированы на решение задач систематизации, хранения и обработки «неколичественной» информации об организации бизнеса, которые раньше не имели адекватной компьютерной поддержки [1].

Сегодня программные продукты для организационного моделирования являются самостоятельными программами и поставляются вне рамок дорогих решений по автоматизации. Они применяются не только для решения задач проектирования систем управления, но и для организации всего предприятия в целом.

Близкими родственниками ORGWARE также являются так называемые CASE-средства – средства компьютерного проектирования программных систем (или просто систем) – Computer Added Software (System) Engineering. Но в отличие от CASE-средств, ORGWARE ориентирован не только на специалистов, применяющих эти программы в рамках проектов по внедрению ERP-систем и связанных с этим реинжинирингом бизнес-процессов¹ компании, но, прежде всего, на конечных пользователей – менеджеров компаний, использующих организационный продукт как повседневный инструмент управления [1].

ORGWARE являются средствами поддержки именно «организационного менеджмента» или организации деятельности, чем они радикально отличаются от других программных компонентов корпоративных информационных систем, нацеленных на автоматизацию оперативного управления ресурсами предприятия – финансами, материалами, персоналом, управленческим и бухгалтерским учетом. ORGWARE – программные продукты для управления организацией деятельности предприятия (формализация целей и стратегических направлений развития, реструктуризация управления, совершенствование и реинжиниринг бизнес-процессов и т.п.). Продукты основаны на технологии бизнес-инжиниринга² – технологии управления, опирающейся на использование информационных моделей предприятия. В процессе построения бизнес-модели компании происходит ее последовательное процессно-целевое описание [4]. Это позволяет, в конце концов, получить взаимосвязанные ответы на основные вопросы управления «зачем» – «что» – «где» – «кто» – «как» – «когда» – «кому» – «сколько»... На этапе организационного проектирования проводится «вертикальное» системно-целевое описание компании. При этом процессы, протекающие в компании, в свернутом виде (как

¹ Бизнес-процесс – деятельность фирмы в рамках производства одного вида продукции или оказания одного вида услуг, т.е. каким образом фирма получает прибыль в рамках одного вида деятельности. Например, ответы на вопросы: «кому продал?», «у кого купил?», «кто занимается этим на фирме?», «чем он руководствуется при принятии решений относительно цены?», «где берет для этого информацию?», «кто для него ее представляет?», – дают представление о внутреннем устройстве фирмы [2].

² Бизнес-инжиниринг – технологии управления, основанные на информационных моделях предприятия: организационно-функциональной структуре, бизнес-процессах, жизненном цикле продукта, а также моделях внешней среды.

функции) идентифицируются, классифицируются и закрепляются в иерархии управления [2].

На российском рынке продукты класса ORGWARE представлены следующими проектами: «Биг-Мастер», выпускаемый компанией «БИГ-СПб», а также «1С-Инталев: Бизнес-архитектор» – совместный проект компаний «1С» и «Инталев».

Программный продукт «БИГ-Мастер»

Самый первый российский ORGWARE был разработан компанией «БИГ-СПб». «БИГ-Мастер» близок по своим возможностям к средствам моделирования, входящим в состав зарубежных комплексных систем управления предприятиями, основанных на методологиях продуктов класса ERP. Они существенно облегчают формальное описание деятельности предприятия. Однако, в отличие от них, БИГ-Мастер является самостоятельным модулем, интегрируемым с другими программными средствами.

К аналогам «БИГ-Мастера» относятся специализированные CASE-средства анализа и моделирования информационных систем (такие продукты, как BPwin фирмы Platinum Technologies или Rational Rose фирмы Rational Software). Особое место в перечне такого рода продуктов принадлежит пакету Aris Toolset компании IDS Prof. Scheer. Но в CASE-средствах отсутствует генерация документов-регламентов, ориентированных на менеджеров компании. Между тем в «БИГ-Мастере» это является одной из главных задач.

Еще один класс программных продуктов, имеющих отношение к организации деятельности предприятия, относительно которых позиционируется «БИГ-Мастер», – это традиционные программы для управления персоналом («системы учета кадров»). До сих пор они автоматизируют, прежде всего, учет данных о персонале (Ф.И.О., пол, возраст, должность и т.п.), а также расчет определенных показателей, характеризующих его деятельность (заработной платы, рабочего времени и т.п.).

И, наконец, наиболее естественно сближение «БИГ-Мастера» с активно развиваемыми сейчас системами Workflow³. В большинстве имеющихся на рынке продуктов этого класса имеется встроенный графический редактор маршрутов (или карт) деловых процессов, позволяющий визуализировать происходящие на предприятии бизнес-процессы. С помощью этого редактора можно создавать и видоизменять сценарии выполнения той или иной операции.

Возможны два варианта взаимодействия «БИГ-Мастера» с такими системами. Первый – это организация экспорта создаваемых в нем описаний бизнес-процессов в готовые системы Workflow с использованием стандартных протоколов обмена данными. Другой – это создание собственных механизмов управления бизнес-процессами, что и реализуется в новой версии продукта, ориентированного на Интранет-технологии.

В состав комплекса «БИГ-Мастер» входят «конструктор бизнес-моделей» – его исполнительное ядро («Оргмастер») и набор функционально-ориентированных модулей, которые можно подключать по желанию.

«Оргмастер» реализует построение базовой модели, состоящей из иерархически упорядоченных классификаторов функций-процессов, организационных процессов, ресурсов, документов и отношений между ними. Это позволяет провести первичный управленческий анализ организационных аспектов деятельности предприятия. На следующем этапе ядро реализует описание процессов внутри компании.

В результате работы «Оргмастера» формируется первичный набор базовых организационных документов – Положения об организационной структуре, Положения о

³ Workflow – класс продуктов, полностью или частично автоматизирующий бизнес-процессы предприятия, при этом документы, информация или задания передаются для выполнения необходимых действий от одного участка к другому в соответствии с набором определенных процедурных правил [2].

контурах управления и бизнес-процессах компании, Положения о подразделениях (службах, отделах), а также спецификации выделенных процессов компании. Таким образом, программа обеспечивает реализацию начального этапа постановки регулярно-го менеджмента в компании.

Затем, используя необходимые модули, можно получить:

- 1) более детальную (по сравнению с общей) организационно-функциональную модель соответствующей предметной области;
- 2) специальные модели, характерные для данной области (например, финансовые модели, рыночные маркетинговые модели, стратегические модели SWOT-факторов и целеполагания и т.п.);
- 3) референтные модели предприятий, полученные в ходе выполнения реальных проектов;
- 4) библиотечные классификаторы, содержащие типовые наборы специализированных функций и других объектов описания предметной области;
- 5) специальные внешние приложения, подключаемые из «БИГ-Мастера» в среде MS Office.

Входных документов как таковых не существует. Данные заносятся в классификаторы группой, обычно сформированной из представителей заказчика и специалистов по системе. Выходными документами служат уже упоминавшиеся отчеты. Одной из интересных особенностей отчетов является возможность раскрытия их на нужную глубину. Если пользователя интересует информация по подразделениям, простым щелчком мышки он может исключить из рассмотрения более мелкие подуровни или, наоборот, детально рассмотреть конкретное подразделение.

Отчеты также можно выгружать в комплект офисных программ Microsoft Office и обрабатывать средствами его приложений.

Программный продукт «1С-Инталев: Бизнес-архитектор»

Программный продукт «1С-Инталев: Бизнес-архитектор» разработан компанией «Инталев» и распространяется фирмой «1С» в рамках проектов «1С:Консалтинг» и «Совместные решения».

Данный продукт рекомендован к применению в проекте «1С: Консалтинг» как инструмент топ-менеджера, аналитика или консультанта в сфере управленческих технологий, предназначенный для проектирования и оптимизации структур управления предприятием. Продукт помогает повышать управленческую прозрачность предприятия, распределять ответственность между управленцами, устранять сбои при взаимодействии различных подразделений и выстраивать эффективные взаимосвязи элементов внутри структуры компании.

Продукт позволяет формализовать, т.е. перевести в управляемое состояние, различные области управления, такие как система менеджмента качества, бюджетирование, целеполагание и стратегический менеджмент, организационное проектирование. Результатом применения программы является четко описанная структура управления предприятием, а также ряд сформированных регламентных положений и инструкций, являющихся основой для взаимодействия и деятельности компании.

Одна из основных функций программы – представление данных в виде матрицы распределения. Эта возможность позволяет в удобной форме представить и распределить любые связанные данные между собой. Организованные и структурированные таким образом данные можно выгружать в функциональные модули, позволяющие решать прикладные задачи. Структурированные в «Бизнес-архитекторе» данные могут формировать информационную основу всего программного обеспечения предприятия.

Примером подобного использования «Бизнес-архитектора» может быть кадровый учет. Если на предприятии был сформирован состав-классификатор подразделений, эти

данные могут быть использованы в информационном модуле учета кадров предприятия, в то время как другая часть той же модели предприятия, связанная с бюджетированием – например, управленческий план счетов и финансовая структура – могут использоваться в информационном модуле финансового планирования и учета. Но информационную основу для обеих модулей представляет единая модель данных в «Бизнес-архитекторе» [3].

Приведенные примеры далеко не исчерпывают список возможных областей применения. Можно использовать программу при структуризации и решении управленческих задач в таких областях, как маркетинг, производство, система качества и многие другие.

Преимущества данного продукта:

- 1) возможность применения на предприятиях различных областей;
- 2) наглядность;
- 3) дружественный интерфейс;
- 4) простота внедрения и эксплуатации;
- 5) наличие референтных моделей – методические образцы проектирования бизнеса;
- 6) интеграция с «1С: Предприятие 7.7».

Описание моделей происходит при помощи классификаторов.

Классификация – это способ систематизации. С помощью системы иерархических классификаторов можно построить систематизированное формализованное описание бизнеса. После создания классификаторов задаются связи между их элементами. К примеру, если определено, что бизнес-функцию «Закупка канцелярии» выполняет оргзвено «Отдел снабжения», в дальнейшем никаких разногласий между отделами об ответственности быть не может.

Проекция – это логическое объединение группы классификаторов и связей между ними в единое целое. В программе реализована удобная работа с многомерными матрицами неограниченной размерности.

После создания классификаторов и проекций можно задавать типовые отчеты, отражающие внутрифирменные регламентные документы и положения. Эти отчеты будут формироваться прямо в MS Word.

Отредактировав классификатор и проекции, можно получить новые регламентные документы в нужном формате. Также в программе возможно сформировать оперативные отчеты, управленческие отчеты и др.

Каждый классификатор и измерение проекции, а также их заголовки могут быть получены в заданном пользователем представлении.

Для улучшения дружественности интерфейса все достаточно сложные операции осуществляются с помощью мастеров (Wizards), что особенно полезно на первых этапах работы.

Разработчиками создан общий «Мастер Бизнес-архитектора», обучающий работе с программой, последовательно, шаг за шагом проходя через все этапы создания структуры области управления предприятия, а также позволяющий через него обратиться к любой функции программы.

Набор функциональных возможностей (меню пользователя) определяется позиционированием на конкретном объекте системы, что значительно облегчает работу с системой.

Модель предприятия может содержать несколько областей управления, каждая из которых характеризуется определенным набором классификаторов, проекций и отчетов.

В программе возможна коллективная работа пользователей. Гибкая настройка безопасности позволяет задать каждому пользователю на каждую область управления

свои права доступа. Существует архив моделей, используя который, всегда можно в случае неудачного изменения вернуться к предыдущему варианту.

В программе предусмотрен обмен моделями в современном формате XML. Консультанты могут пользоваться результатами своего труда многократно, а предприятия, например холдинговые структуры, могут обмениваться своими управленческими работками.

Пользователю программы с помощью удобного мастера импорта данных доступен импорт данных из любой конфигурации «1С: Предприятие».

Путем последовательного выполнения простых операций импорт имеющихся данных в проекции и классификаторы системы «Бизнес-архитектор» выполняется наглядно и удобно.

В программе предусмотрена возможность сохранения как классификаторов, так и проекций в формате txt и XML. Любой пронумерованный список во внешнем файле легко может быть импортирован в «Бизнес-архитектор» в качестве классификатора.

Заключение

Таким образом, современные отечественные разработки систем моделирования предприятий могут использоваться как инструмент топ-менеджера, аналитика или консультанта в сфере управленческих технологий, предназначенный для проектирования и оптимизации структур управления предприятием. Данные продукты помогают повышать управленческую прозрачность предприятия, распределять ответственность между управленцами, устранять сбои при взаимодействии различных подразделений и выстраивать эффективные взаимосвязи элементов внутри структуры компании. Они позволяют формализовать, т.е. перевести в управляемое состояние, различные области управления, например, бюджетирование, целеполагание и стратегический менеджмент и т.д. Они применяются не только для решения задач проектирования систем управления, но и для организации всего предприятия в целом. Эти программные продукты ориентированы на решение задач систематизации, хранения и обработки «неколичественной» информации об организации бизнеса, которые раньше не имели адекватной компьютерной поддержки. Они имеют поддержку взаимодействия и обмена данными с другими типами программного обеспечения, необходимыми для функционирования организации. Сегодня программные продукты для организационного моделирования являются самостоятельными программами и поставляются вне рамок дорогих решений по автоматизации.

Литература

1. «Orgware» – новый класс программ для управления организацией. http://www.big.spb.ru/bigmaster/basic_concepts/orgware_orgmanagement.shtml
2. Горелик С. Бизнес-инжиниринг и организационное проектирование. http://www.big.spb.ru/publications/bigspb/be_orgproject.shtml
3. Романов Ю. Структуризация бизнеса с помощью продуктов ORGWARE. <http://www.intalev.ru/publications/?uin=28>
4. Романов Ю., Иванова Т. Внедрение автоматизированной системы процессного управления – на шаг впереди конкурентов (системы класса workflow). <http://www.intalev.ru/publications/?uin=30>

КАНАЛЫ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА К ИНФОРМАЦИИ И СРЕДСТВА ИХ ЗАЩИТЫ В КОМПЛЕКСАХ УПРАВЛЕНИЯ ПОДВИЖНЫМИ ОБЪЕКТАМИ

С.В. Егоров

Научный руководитель – д.т.н., профессор Ю.А. Гатчин

В работе выявлены возможные каналы несанкционированного доступа к информации в комплексах управления подвижными объектами. Проведена классификация этих каналов и предложены средства защиты их от несанкционированного доступа.

Введение

Проблема безопасности информации продолжает оставаться актуальной. Постоянная интенсификация информационных процессов в обществе порождает ряд попутных и достаточно серьезных проблем, без решения которых нельзя говорить об эффективности информатизации [1, 4, 7].

Решение названной проблемы осложняется еще и тем, что в применяемой сегодня практике отсутствует привязка к типовым объектам автоматизации [4]. В данной работе ставится задача выявить каналы несанкционированного доступа к информации в комплексах управления подвижными объектами (КУПО) и предложить средства защиты этих каналов.

1. Объект исследования и возможные каналы несанкционированного доступа

В качестве объекта исследования выступает комплекс управления подвижными объектами. Этот комплекс представляет собой два территориально разделенных объекта, осуществляющих между собой связь по локальной вычислительной системе и специальным каналам связи. Связь с объектами управления осуществляется через радиоканалы. Схема этого комплекса изображена на рис. 1.

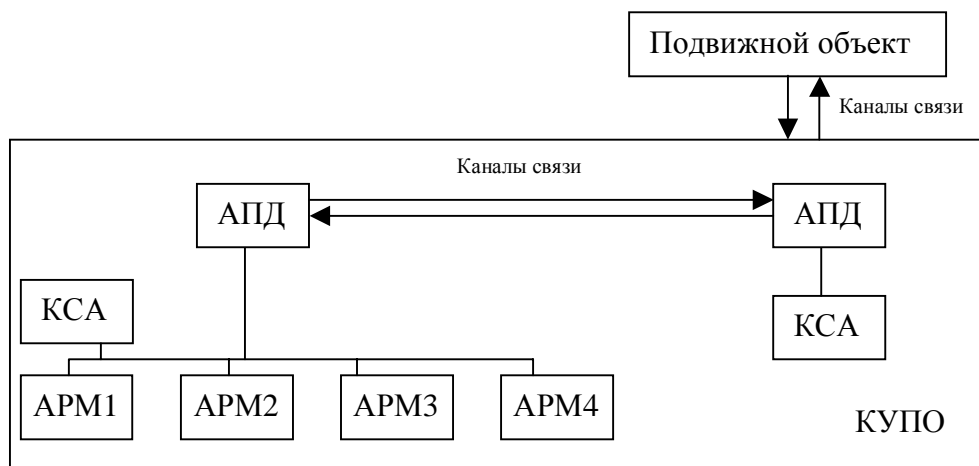


Рис. 1. Схема комплекса управления подвижными объектами: АПД – аппаратура передачи данных, КСА – комплекс средств автоматизации, АРМ – автоматизированное рабочее место

Для комплекса такого рода характерны следующие штатные(законные) каналы доступа к информации:

- устройства ввода (вывода) информации;
- средства отображения информации;
- средства документирования информации;
- средства загрузки программного обеспечения;
- машинные носители информации;

- бумажные носители информации;
- внешние каналы связи.

Имея в виду, что при отсутствии защиты нарушитель может воспользоваться как штатными, так и нештатными каналами доступа, можно назвать возможными каналами несанкционированного доступа (ВК НСД) следующие каналы:

- все перечисленные выше штатные средства при их использовании законными пользователями не по назначению или за пределами своих полномочий или при их использовании посторонними лицами;
- технологические пульта и средства управления;
- внутренний монтаж аппаратуры;
- линии связи между аппаратными средствами данной вычислительной системы;
- побочное электромагнитное излучение с аппаратуры технических средств; побочные наводки по сети электропитания и шинам заземления аппаратуры; побочные наводки на вспомогательных и посторонних коммуникациях, сервисном оборудовании;
- отходы в виде бумажных и машинных носителей информации, брошенных в мусорную корзину, а также ремонтируемые запоминающие устройства, содержащие остаточную информацию.

Особое внимание следует уделить внешним каналам связи как наиболее уязвимым и трудно контролируемым каналам. Особую привлекательность для нарушителя эти каналы представляют потому, что в них передается, как правило, готовая (конечная) информация по одному протоколу, в то время как внутри КУПО информация циркулирует по различным протоколам и в необработанном виде. В то же время особенность информации, передаваемой к самим подвижным объектам, состоит в том, что время жизни самой информации очень мало, поэтому в равной степени с защитой содержимого канала связи стоит задача сохранения целостности самого канала связи.

2. Классификация возможных каналов НСД

При меньшей номенклатуре и количестве технических средств КСА, а также выборе модели поведения менее квалифицированного нарушителя количество ВК НСД к информации уменьшается, уменьшаются, соответственно, количество и прочность применяемых средств защиты. Поэтому используем классификацию каналов НСД по классам нарушителей. Приведем следующую классификацию [2]:

- высококвалифицированный нарушитель-профессионал (специалист по вычислительной технике, специально подготовленный к несанкционированному доступу к информации);
- квалифицированный нарушитель-непрофессионал (специалист по вычислительной технике, не подготовленный к несанкционированному доступу к информации);
- неквалифицированный нарушитель-непрофессионал;
- прочие нарушители.

Исходя из классификации нарушителей, приведем классификацию каналов несанкционированного доступа [4]:

I класс – все ВК НСД, известные на момент разработки систем;

II класс – все ВК НСД, кроме побочного электромагнитного излучения (ПЭМИН), анализа трафика и идентификаторов абонентов;

III класс – все ВК НСД, кроме ремонтируемой аппаратуры, носителей информации, перемещаемых в пределах контролируемой зоны, остатков информации в запоминающих устройствах, аппаратуры передачи данных и внутренних линий связи, анализа трафика и идентификаторов абонентов, дублирования, переупорядочивания и задержки сообщений.

IV класс – предлагается не вводить на основании того, что исследуемые КУПО слишком дороги, чтобы в них были созданы условия безопасной обработки информации ни-

же III класса, так как динамика современной жизни часто меняет отношение к той или иной информации по различным причинам.

3. Применение средств защиты информации от НСД в КУПО

Исходя из возможных каналов НСД и возможных нарушителей, классификация которых приведена выше, рассмотрим распределение известных средств защиты по возможным каналам НСД в КУПО. Это распределение представлено в табл. 1.

№ п/п	Наименование ВК НСД	Класс защиты			Средства защиты
		I	II	III	
1	2	3	4	5	6
1	Устройства ввода (вывода) информации	+	+	+	Система контроля и разграничения доступа в помещения. Программно-аппаратный комплекс контроля входа в систему. Программа контроля и разграничения доступа к ПО и информации КСА.
2	Аппаратура отображения и документирования информации	+	+	+	Система разграничения и контроля доступа в помещения
3	Ремонтируемая и профилактируемая аппаратура	+	+	-	Система разграничения и контроля доступа в помещения. Система контроля ввода (вывода) аппаратуры в (из) рабочий контур обмена информацией. Средства стирания остатков информации. Средства наложения на остатки информации случайной последовательности символов и чисел. Средства уничтожения носителей секретной информации
4	Машинные носители информации	+	+	+	Учет и разграничение доступа к носителям. Электронная идентификация носителей. Шифрование информации. Резервирование информации с охраной ее копии
5	Документы	+	+	+	Учет, регистрация и разграничение доступа к документам
6	Носители программного обеспечения	+	+	+	Учет, регистрация и разграничение доступа к носителям ПО. Верификация и контроль целостности ПО. Резервирование ПО с контролем доступа к его копии

7	Машинные носители с остатками информации	+	+	+	+	Учет, регистрация и разграничение доступа. Средства стирания информации. Наложение случайной последовательности символов и чисел. Средства уничтожения носителей
8	Бумажные носители с остатками информации	+	+	-	-	Средства уничтожения носителей
9	Средства загрузки ПО	+	+	+	+	Средства контроля и разграничения доступа в помещения. Средства контроля и блокировки доступа к загрузке ПО.
10	Технологические пульты и органы управления, внутренний монтаж аппаратуры	+	+	-	-	Средства контроля и разграничения доступа в помещения. Система контроля вскрытия аппаратуры
11	Внутренние линии связи между аппаратными средствами КСА	+	+	-	-	Средства контроля доступа на территорию КУПО. Средства контроля вскрытия аппаратуры. Скрытая прокладка линий связи. Шифрование передаваемой информации
12	Внешние каналы связи КСА	+	+	+	+	Программа контроля и разграничения доступа к информации КСА. Шифрование передаваемой информации Организационные и технические средства защиты целостности канала связи с подвижным объектом
13	Побочное электромагнитное излучение и наводки информации	+	-	-	-	Средства снижения или зашумления уровня излучения и наводок информации на границе контролируемой зоны объекта автоматизации
Примечание: Знак «+» – наличие средства защиты; знак «-» – отсутствие средства защиты.						

Таблица 1. Распределение известных средств защиты по возможным каналам НСД в КУПО

Перечисленные в таблице средства могут быть применены, если они удовлетворяют требованиям к системе защиты информации и заданной прочности защитной оболочки того или иного класса. Однако этот расчет прочности защиты необходимо уточнить после объединения этих средств в единую систему, т.е. после создания средств централизованного контроля и управления.

Особое внимание следует уделить защите информации в каналах связи, так как реализовать защиту от НСД к таким каналам представляется крайне сложной задачей.

Минимальной частью внутренней информации в канале связи является кадр кодограммы. Кадр кодограммы включает в себя служебные признаки, адресную и содержательную части. При отсутствии защиты информации нарушитель может делать с ней почти все, что пожелает. Перечень его возможных несанкционированных действий хотя и обширен, но ограничен и хорошо известен [1, 3, 7]. В этот перечень входят:

- изменение, удаление, задержка, переупорядочение, дублирование регулярных и посылка ложных сообщений;
- воспрепятствование передаче сообщений;
- осуществление ложных соединений;
- чтение содержания сообщений;
- анализ трафика и идентификаторов абонентов сети;
- чтение и использование пароля законного пользователя;
- нарушение своих полномочий законным пользователем путем обращений к запрещенной для него информации.

К ним добавляются возможные несанкционированные действия особого рода законных пользователей-нарушителей: отказ отправителя от собственного сообщения и получателя – от принятого, а также массированные повторы сообщений и передача различного рода информационного «мусора».

Средства защиты информации для каналов связи представлены в табл. 2 [5, 6].

№ п/п	Возможные несанкционированные действия	Средства защиты информации КУПО	Средства защиты информации СПД
	Нарушение целостности открытого канала связи с подвижными объектами	<p>Технические средства:</p> <ul style="list-style-type: none"> - программа выбора диаграмм направленности передачи сигналов в канал связи; - программа оперативного изменения параметров радиосигналов с целью приближения несущего сигнала к шумоподобному; - программа оперативного изменения способа информации в передаваемом радиосигнале при сохранении синхронизации абонентов канала связи – введение второго уровня шумоподобности сигнала <p>Организационные средства:</p> <ul style="list-style-type: none"> - средства предельного сокращения времени информационного обмена в канале связи; - средства учета пространственного разнеса подвижного объекта управления и субъекта НСД; - средства увода диаграммы направленности сигналов канала связи от направления на субъект НСД (выполняется совместно с маневрированием подвижного объекта) 	

2	Чтение содержания сообщения	Абонентское шифрование	Линейное шифрование
3	Дублирование сообщения в целях навязывания ложной информации	Средства шифрования даты \ времени отправления сообщения вместе с его содержательной частью	
4	Переупорядочение сообщения	Средства шифрования даты и времени отправления сообщения вместе с содержательной частью	Шифрование и цифровая подпись служебных частей кодограммы
5	Задержка или удаление сообщения	Квитанция о получении сообщения. Введение контрольного интервала времени ответа	Дублирование соединения или маршрута
6	Переадресация сообщения	Цифровая подпись адреса получателя одновременно с содержательной частью сообщения	Шифрование и цифровая подпись служебных частей кодограммы
7	Посылка ложного сообщения	Цифровая подпись содержательной части сообщения	
8	Анализ трафика	Специальные средства заполнения потока	Линейное шифрование
9	Анализ идентификаторов получателей		Линейное шифрование
10	Чтение и подбор паролей отправителей сообщений	Межсетевой экран КСА КУПО Средства идентификации и аутентификации	Средства идентификации и аутентификации
11	Нарушение законным пользователем своих полномочий	Межсетевой экран КСА КУПО Средства разграничения и контроля доступа КСА КУПО	Средства разграничения и контроля доступа УК СПД
12	Отказ отправителя от переданного, получателя – от принятого сообщения	Центр контроля и управления безопасностью информации в КУПО	Центр контроля и управления безопасностью информации в СПД

Таблица 2. Средства защиты информации для каналов связи

Полнота перекрытия перечисленных несанкционированных действий соответствующими средствами будет определять степень замыкания защитных оболочек. При этом средства защиты от отказов законных пользователей от принадлежащих им документов в оболочку входить не должны по причине выполнения ими другой задачи – регистрации сообщения в центре управления и контроля, результаты решения которой должны проходить отдельным показателем. Вполне возможно, что показатели средств защиты ресурсов сети от повторов сообщений и информационного «мусора» также должны быть отдельными.

Расчет прочности защитной оболочки КУПО при использовании указанных средств по методике, описанной в [4], дает результат $P_{об} = 0,82$, где $P_{об}$ – прочность защитной оболочки КУПО, что говорит о том, что событие преодоления защитной оболочки малове-

роятно [4]. Если такая преграда окажется недостаточной, то может быть заменена новой (на основе новых технических средств) или продублирована второй преградой.

Заключение

В работе были изучены каналы несанкционированного доступа к информации для конкретного типа автоматизированной системы – комплекса управления подвижными объектами. Также была проведена классификация этих каналов и предложены средства защиты каналов от несанкционированного доступа.

Среди всех возможных каналов доступа к информации наиболее уязвимым представляется канал связи с внешними объектами, поэтому в работе уделено особое внимание средствам защиты этого канала. Основным средством является шифрование как содержательной части кодограммы, так и ее служебной части.

Работа служит основой для построения системы защиты информации в комплексах управления подвижными объектами.

Литература

1. Анин Б.Ю. Защита компьютерной информации. СПб: БХВ-Петербург, 2000.
2. Временное положение по организации разработки, изготовления и эксплуатации программных и технических средств защиты информации от несанкционированного доступа в автоматизированных системах и средствах вычислительной техники. М.: Гостехкомиссия РФ. М.: Воениздат, 1992.
3. Корнеев И.Р., Беляев А.В. Информационная безопасность предприятия. СПб: БХВ-Петербург, 2003. 752 с.
4. Мельников В.В. Безопасность информации в автоматизированных системах. М.: Финансы и статистика, 2003. 368 с.
5. Романец, Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. / 2-е изд. М.: Радио и связь, 2001.
6. Соколов А.В., Степанюк О.М. Защита от компьютерного терроризма. Справочное пособие. СПб: БХВ-Петербург; Арлит, 2002. 496 с.
7. Щеглов А.Ю. Защита компьютерной информации от несанкционированного доступа. СПб: Наука и техника, 2004. 384 с.

ОБЗОР ОСНОВНЫХ ПРОТОКОЛОВ ЭЛЕКТРОННОЙ ПОДПИСИ

С.Ю. Каменева, О.В. Михайличенко

Научный руководитель – д.т.н., профессор А.Г. Коробейников

В статье рассмотрены проблемы аутентификации данных. Приведен теоретический принцип действия электронной цифровой подписи (ЭЦП), рассмотрены широко используемые протоколы, а также приведена схема комплексной защиты сообщений с использованием ЭЦП и комбинированного шифрования.

Введение

Аутентификация сообщений защищает две обменивающиеся сообщения стороны от любой третьей, но не обеспечивает защиту каждой из сторон от другой. Здесь имеется ряд возможностей для возникновения конфликтов.

В ситуациях, когда нет полного доверия между отправителем и получателем, требуется нечто большее, чем простая аутентификация. Наиболее привлекательным решением этой проблемы оказывается цифровая подпись. Она должна обеспечить следующие возможности [1]:

- установить автора, а также дату и время подписи;
- установить достоверность содержимого сообщения на время подписи;
- проверки подписи третьей стороной на случай возникновения спора.

На основе этих свойств можно сформулировать следующие требования, выдвигаемые к цифровой подписи.

- Подпись должна быть двоичным кодом, который зависит от подписываемого сообщения.
- Подпись должна использовать некоторую информацию, уникальную для отправителя, чтобы предотвратить возможность как фальсификации, так и отрицания авторства.
- Цифровую подпись должно быть относительно просто распознать и проверить.
- С точки зрения вычислений должно быть нереально фальсифицировать цифровую подпись ни с помощью создания нового сообщения для имеющейся цифровой подписи, ни с помощью создания фальшивой подписи для имеющегося сообщения.
- Экземпляр цифровой подписи должен быть удобно хранить в запоминающем устройстве.

Протоколы электронной подписи

Протоколы (схемы) электронной подписи являются основными криптографическим средством обеспечения целостности информации [2].

Схема Эль Гамала

Пусть обоим участникам протокола известны некоторое простое число p некоторой порождающей g группы Z_p^* и некоторая хэш-функция h .

Подписывающий выбирает секретный ключ $x \in_R Z_{p-1}^*$ и вычисляет открытый ключ $y = g^x \pmod p$. Пространством сообщений в данной схеме является Z_{p-1} .

Для генерации подписи нужно сначала выбрать $u \in_R Z_{p-1}$. Если $u \notin_R Z_{p-1}^*$ (что проверяется эффективно), то необходимо выбрать новое u . Если же $u \in_R Z_{p-1}^*$, то искомой подписью для сообщения m является пара (r, s) , где $r = g^u \pmod p$, $s = u^{-1}(h(m) + xr) \pmod{(p-1)}$. Параметр u должен быть секретным и может быть уничтожен после генерации подписи.

Для проверки подписи (r, s) для сообщения m необходимо сначала проверить условия $r \in Z_p^*$ и $s \in Z_{p-1}$. Если хотя бы одно из них ложно, то подпись отвергается. В противном случае подпись принимается только тогда, когда $g^{h(m)} \equiv y^r r^s \pmod p$.

Вера в стойкость схемы Эль Гамала основана на (гипотетической) сложности задачи дискретного логарифмирования по основанию g .

Схема Фиата – Шамира

Для ее обеспечения центр обеспечения безопасности должен выбрать псевдослучайную функцию f , криптографическую хэш-функцию h , а также выбрать различные большие простые числа p , q и вычислить $n = pq$. Число n и функции f и h являются общедоступными и публикуются центром, а числа p и q должны быть секретными. Кроме того, схема использует два натуральных параметра l и t .

Для каждого пользователя центр обеспечения безопасности генерирует идентификационную информацию I , содержащую, например, имя пользователя, его адрес, идентификационный номер и т. п., и для каждого $j = 1, \dots, l$ вычисляет $y_j = f(I, j)$, отбирает среди них квадратичные вычеты по модулю n (изменив обозначения, мы считаем, что y_j для всех $j = 1 \dots l$ являются квадратичными вычетами по модулю n) и вычисляет x_j – наименьший квадратичный корень по модулю n из $y_j^{-1} \pmod n$. Числа y_j играют роль открытого ключа, а x_j – секретного.

Так как эти ключи вычисляются с использованием I , схема Фиата – Шамира относится к схемам, основанным на идентификационной информации (identity based). В другом варианте схемы Фиата – Шамира сразу выбираются (псевдослучайным образом) по параметру y_j . На практике идентификационная информация I и/или открытый ключ (y_1, \dots, y_l) каждого пользователя помещаются в некоторый справочник, доступный всем пользователям для чтения, но не доступный для записи. Для обеспечения аутентичности данные в этом справочнике заверяются подписью центра обеспечения безопасности. Секретный ключ (x_1, \dots, x_l) и идентификационная информация I могут быть помещены на интеллектуальную карточку пользователя.

Для генерации подписи для обеспечения t подписывающих:

1. выбирает $u_i \in_{\mathbb{R}} \mathbb{Z}_n$ (каждое u_i – независимо друг от друга) и вычисляет $r_i = u_i^2 \pmod n$ для $i = 1, \dots, t$;
2. вычисляет $h(m, r_1, \dots, r_t)$ и полагает биты e_{ij} ($i = 1, \dots, t, j = 1, \dots, l$) равными первым lt битам $h(m, r_1, \dots, r_t)$;
3. вычисляет для $i = 1 \dots t$ $v_i = u_i \prod_{j=1}^l x_j^{e_{ij}} \pmod n$.

Искомой подписью для сообщения m является набор $(e_{ij}, v_i \mid i = 1 \dots t, j = 1 \dots l)$

Для проверки подписи $(e_{ij}, v_i \mid i = 1, \dots, t, j = 1, \dots, l)$ для сообщения m подписывающий:

1. вычисляет $v_j = h(I, j)$ для $j = 1, \dots, l$ или берет их из общедоступного справочника и сравнивает их с имеющимися в подписи (если обнаружено несовпадение – подпись отвергается);
2. вычисляет для $i = 1 \dots t$ $z_i = v_i^2 \prod_{j=1}^l y_j^{e_{ij}} \pmod n$.

Подпись принимается тогда и только тогда, когда первые lt битов $h(m, z_1, \dots, z_t)$ равны e_{ij} .

Несомненным достоинством схемы Фиата – Шамира является отсутствие дискретного экспонентирования, что делает схему весьма эффективной. Но, с другой стороны, в этой схеме длины ключей и подписи значительно больше, чем в схемах типа Эль Гамала.

Схема стандарта электронной подписи ANSI США (DSA)

Эта схема аналогична схеме Эль Гамала, но несколько эффективнее, так как в ней порядок g меньше, чем в схеме Эль Гамала. Пусть в открытом доступе имеются некоторые простые числа p , q такие, что $q \mid p-1$, а также элемент g порядка q группы \mathbb{Z}_p^* и

хэш-функция h , действующая из пространства сообщений в Z_q^* . Параметры p, q, g и хэш-функция h могут быть выбраны центром обеспечения безопасности. Подписывающий выбирает секретный ключ $x \in_R Z_q$ и вычисляет открытый ключ $y = g^x \bmod p$. Для генерации подписи для сообщения m нужно выбрать $u \in_R Z_q^* \setminus \{1\}$ и вычислить $r = g^u \bmod p \bmod q$ и $s = u^{-1} (h(m) + xr) \bmod q$. Параметр u должен быть секретным и может быть уничтожен после вычисления r и s . Если $r = 0$ или $s = 0$, то выбираются новое значение u , и процесс генерации подписи повторяется. В противном случае (r, s) – искомая подпись для сообщения m .

Для проверки подписи (r, s) для сообщения m необходимо сначала проверить условие $0 < r < q$ и $0 < s < q$. Если хотя бы одно из них ложно, то подпись отвергается. В противном случае подпись принимается тогда и только тогда, когда:

$$g^{vh(m)} y^{vr} \bmod p \bmod q = r, \text{ где } v = s^{-1} \bmod q.$$

Схема стандарта электронной подписи ГОСТ

Пусть p, q, g, h, x, y имеют тот же смысл, что и в схеме DSA. Для генерации подписи для сообщения m нужно выбрать $u \in_R Z_q^* \setminus \{1\}$ и вычислить $r = g^u \bmod p \bmod q$ и $s = u^{-1} (h(m) + xr) \bmod q$. Параметр u должен быть секретным и может быть уничтожен после вычисления r и s . Если $r = 0$ или $s = 0$, то выбираются новое значение u и процесс генерации подписи повторяется. В противном случае (r, s) – искомая подпись для сообщения m [3].

Для проверки подписи (r, s) для сообщения m необходимо сначала проверить условие $0 < r < q$ и $0 < s < q$. Если хотя бы одно из них ложно, то подпись отвергается. В противном случае подпись принимается тогда и только тогда, когда $g^{ws} y^{-wr} \bmod p \bmod q = r$, где $w = h(m)^{-1} \bmod q$.

Схема RSA

В схеме RSA подписывающий выбирает два различных больших простых числа p и q , которые играют роль секретного ключа, и публикует открытый ключ (n, e) , где $n = pq$, а e – некоторое число, взаимно простое с $\phi(n) = (p-1)(q-1)$ (ϕ – функция Эйлера). Подписью для сообщения m является $s(m) = h(m)^d \bmod n$, где $d = e^{-1} \bmod \phi(n)$ (очевидно, что, зная p и q , можно эффективно вычислить d) и h – хэш-функция. Проверка подписи s для сообщения m состоит в проверке сравнения $s^e \equiv h(m) \pmod{n}$.

Схема RSA достаточно эффективна и широко используется на практике. Вера в стойкость схемы основана на (гипотетической) трудности задачи факторизации целых чисел.

Комплексная защита сообщений

Поскольку шифрование защищает сообщения от ознакомления, а ЭЦП – от подмены (это две основные угрозы информации в Интернете), то было бы логично для обеспечения более полной безопасности совместно применять ЭЦП и комбинированное шифрование. Для этого нужно выполнить следующее.

На подготовительном этапе двое друзей, например, создают две пары ключей: секретный и открытый для асимметричного шифрования, а также секретный и открытый ключи ЭЦП. Открытыми ключами они обмениваются, а затем один посылает другому сообщение, подписанное своим секретным ключом.

Затем первый друг генерирует случайный ключ симметричного шифрования K , которым шифрует отправляемое письмо, причем только это.

Далее, чтобы можно было сообщение расшифровать, он зашифровывает ключ K (а в открытом виде посылать ключ симметричного шифрования ни в коем случае недопустимо) на открытом ключе асимметричного шифрования своего друга и добавляет его к зашифрованному письму.

Второй друг, получив зашифрованное сообщение, расшифровывает своим секретным ключом асимметричного шифрования ключ K , которым затем расшифровывает и само письмо.

Наконец, он проверяет с помощью открытого ключа друга его ЭЦП в данном письме и убеждается, что оно пришло именно от его друга и в неизменном виде.

Здесь придаться практически не к чему – при грамотном использовании подобная система не оставит хакеру никаких шансов на успех. Правда, может показаться неудобным то, что приходится делать слишком много ключей. Для решения этой задачи предусмотрен алгоритм Диффи – Хеллмана, позволяющий, в частности, применять одну и ту же пару ключей ЭЦП как для собственно ЭЦП, так и для симметричного шифрования.

Смысл данного алгоритма заключается в следующем [4]. В стандарте ГОСТ Р34.10–94 для ЭЦП открытый ключ вычисляется из секретного:

$$K_p = a^K s \bmod p,$$

где a и p – некоторые общеизвестные большие числа (могут принимать значения до 2^{1024} , и это ужасающе большое значение). Предположим, что есть пользователи 1 и 2, сгенерировавшие свои секретные ключи и вычислившие из них открытые:

$$K_{p1} = a^{K_1} s_1 \bmod p;$$

$$K_{p2} = a^{K_2} s_2 \bmod p.$$

После обмена открытыми ключами у каждого из них появилась пара ключей: свой секретный и чужой открытый, т.е. абонент 1 имеет ключи K_{s1} и K_{p2} , а абонент 2 – K_{s2} и K_{p1} . Представим, что будет, если абоненту 2 вдруг вздумается возвести в степень своего секретного ключа открытый ключ абонента 1:

$$(K_{p1})^{K_2} s_2 = (a^{K_1} s_1)^{K_2} s_2 \bmod p = (a^{K_2} s_2)^{K_1} s_1 \bmod p = (K_{p2})^{K_1} s_1 = K_c.$$

Значит, существует ключ K_c , обычно называемый «ключом парной связи», который могут вычислить только абоненты 1 и 2 с использованием имеющихся у них ключей, поскольку у злоумышленников нет K_{s1} или K_{s2} , и поэтому они не сумеют определить K_c . Затем с помощью общего ключа K_c можно быстро симметрично зашифровать сообщения по стандарту ГОСТ 28147–89.

Заключение

В статье были рассмотрены протоколы создания электронной цифровой подписи. Для обеспечения полной безопасности при передаче информации рекомендуется применять электронную подпись и комбинированное шифрование. Вера в рассмотренные выше протоколы ЭЦП основана на гипотетической сложности вычисления (взлома) противником схемы шифрования.

Литература

1. Столлингс В. Криптография и защита сетей: принципы и практика. М.: Издательский дом «Вильямс», 2001. 669 с.
2. Баричев С., Серов С., Основы современной криптографии. М., 2001. 121 с.
3. Алгоритм криптографического преобразования данных. ГОСТ 28147-89.
4. Процессы формирования и проверки электронной цифровой подписи. ГОСТ Р 34.10-2001.

ОБЕСПЕЧЕНИЕ ЦЕЛОСТНОСТИ ДАННЫХ СИСТЕМЫ И ПРЕДОТВРАЩЕНИЕ УТЕЧКИ ИНФОРМАЦИИ

А.А. Малинин

Научный руководитель – д.т.н., профессор Ю.А. Гатчин

В статье представлен оригинальный метод формирования, передачи и обработки данных в локальных и глобальных сетях по протоколу HTTP, который позволяет избежать риска несанкционированного доступа к данным, используемых для авторизации в системах «клиент-сервер».

Введение

Рассмотрим типичную архитектуру систем, о которых пойдет речь. Пользователь, используя веб-браузер или какое-либо другое клиентское программное обеспечение, посредством Интернета или Интранета взаимодействует с сервером системы, который обращается к хранилищу данных, выбирая или модифицируя находящуюся там информацию. Примерами таких систем могут быть веб-сервер, использующий базы данных, а также программный комплекс планирования и управления учебным процессом кафедры высшего учебного заведения.

Можно сразу же выделить объекты, которые могут стать целью атак:

1. клиентское программное обеспечение;
2. сервер системы;
3. хранилище данных.

К этим вполне понятным объектам нужно добавить еще один – назовем его «линии связи», подразумевая под этим путь, который проходят все запросы цепочки «клиент–сервер–хранилище данных». Рассмотрим это на примере. Допустим, у крупной организации есть корпоративный веб-сайт, работающий на веб-сервере Apache, использующий серверный язык Perl и хранящий информацию в базе данных MySQL. Через небольшие промежутки времени работники этой организации добавляют на этот сайт новость следующим образом: используя веб-браузер, заходят в определенный раздел сайта, авторизуются, набирают нужный текст, подтверждают опубликование новости.

В данной ситуации как минимум один раз пользователь вводит ключевую информацию (как правило, пару логин–пароль), чтобы получить определенные привилегии в системе, которые не доступны по умолчанию при гостевом доступе. Соответственно, данные формы с этой ключевой информацией отправляются на сервер методом GET или POST в открытом виде. Эти данные могут быть получены на прокси-сервере организации или провайдера, а также сторонними лицами, имеющими возможность сканирования трафика в сетях, через которые проходят эти GET/POST запросы. Чтобы наглядно увидеть ключевые точки маршрута запроса к сайту, можно набрать в консоли команду `tracert www.ifmo.ru`.

Содержание

В результате потенциальными жертвами атак злоумышленников с целью компрометирования информации являются четыре объекта с их окружением:

1. клиентское программное обеспечение;
2. сервер системы;
3. хранилище данных;
4. линии связи.

Под окружением объекта подразумеваются все средства программного обеспечения конкретного компьютера и его операционной системы, позволяющие получить привилегии пользователя с расширенными правами или пользователя System (Root). Например, на сервере баз данных, используемых в системе, также работает служба

удаленного рабочего стола (Remote desktop). Определенный пользователь может не иметь прав на доступ к серверу баз данных на этом компьютере, но иметь доступ к папке, в которой хранятся файлы баз данных, что позволит ему просмотреть или модифицировать их. Таким образом, доступ к файлам баз данных через службу удаленного рабочего стола является потенциальной угрозой для рассматриваемой системы.

Рассмотрим классификацию уязвимостей с относительной оценкой важности (табл. 1).

Доступ/результат	Отказ в обслуживании	Чтение	Модификация
Локальный	VI (низкая)	IV	III
Удаленный	V	II	I (высочайшая)

Таблица 1. Классификация уязвимостей с оценкой важности

Относительность здесь вполне объяснима – ведь результат будет достигнут. Уязвимость к атакам «Отказ в обслуживании» (D.o.S – Denial of Service) в рамках данной статьи не будет рассмотрена, поскольку в первом приближении не является причиной доступа к информации системы.

Локальный доступ означает возможность непосредственного взаимодействия с компьютером от имени пользователя, имеющего на это привилегии. Удаленный доступ подразумевает возможность взаимодействия с компьютером только через те сервисы, которые доступны извне.

В современных системах пользователю не требуется подтверждать свои права с каждым обращением к серверу. При доступе к сайту, как правило, используется механизм сессии, когда браузер при каждом запросе отправляет веб-серверу уникальный идентификатор сессии, хранящийся в куках или в скрытых полях форм. Схема действий такова:

1. пользователь получает приглашение ко входу в систему с просьбой ввести логин и пароль;
2. эти данные по протоколу HTTP отправляются серверу;
3. сервер «запоминает» идентификатор текущей сессии;
4. при каждом запросе пользователя сервер проверяет, совпадает ли присланный идентификатор с «запомненным» ранее.

К сожалению, в этом случае введенные пароли пересылаются в открытом, текстовом виде. Также зачастую операционная система сохраняет их для удобства пользователя. В случае передачи таких данных форм методом GET ссылка вида «<http://site.com?user=ivanov&password=123&sessid=0123456789abcdefgh>» содержит в явном виде логин и пароль, к тому же эта ссылка останется в логах прокси-серверов, в истории веб-браузера пользователя. Данные, пересылаемые браузером по протоколу HTTP, могут быть легко перехвачены сканерами в локальных сетях. В результате злоумышленники могут пользоваться полученными данными до тех пор, пока не будет изменен пароль или логин скомпрометированного пользователя на сервере.

В данной статье не будут описаны уязвимости протоколов или программного обеспечения и методы их использования. Сегодня для обеспечения защиты от подобного перехвата информации зачастую используются дополнительные средства на стороне сервера и клиента (например, Ява-апплет) по шифрованию/дешифрованию данных, доступ по протоколу HTTPS.

Применение безопасного протокола с шифрованием данных требует установки дополнительного программного обеспечения на сервере, дополнительных навыков в программировании, получения сертификатов безопасности узла.

Разработка дополнительных модулей, таких как Ява-апплеты, требует знания специфических языков программирования и значительно снижает гибкость работы с системой, поскольку в некоторых организациях, Интернет-кафе, клубах и т.д. отключены те или иные возможности веб-браузеров из соображений безопасности (Java, Javascript, VBScript, ActiveX, загрузка модулей). К тому же часто клиентскому браузеру приходится отдельно загружать платформы или плагины для возможности запуска апплетов или ActiveX-приложений.

В данной статье представлен оригинальный метод использования ключевой информации и передачи на базе локальных и глобальных сетей протоколом HTTP, потенциально не подверженных риску при перехвате отправляемых данных. Суть в том, что для авторизации используется переменный код, а не статические величины:

$$P = f(T, D, IP, X_1, \dots, X_n),$$

где P – ключевой код, f – функция от $n + 3$ параметров, T – текущее время, D – текущая дата, IP – IP-адрес компьютера клиента, X_1 – первый параметр, X_n – n -ный параметр.

В этом случае процесс авторизации на веб-сервере может выглядеть следующим образом.

1. Пользователь вводит переменный код.
2. Переменный код отправляется методом POST на веб-сервер.
3. Веб-сервер по определенному алгоритму вычисляет соответствие кода установленному алгоритму и вычленяет кодовую фразу.
4. Веб-сервер сравнивает хэш-функцию кодовой фразы с данными, хранящимися в хранилище данных.
5. Веб-сервер генерирует и «запоминает» идентификатор сессии в зависимости от признаков этого идентификатора (IP-адрес машины, время истечения действия или количество разрешенных действий).
6. При каждом запросе пользователя сервер проверяет соответствие присланных параметров и идентификатора сессии с «запомненным» ранее.

Также веб-серверу необходимо сгенерировать идентификатор сессии для пользователя:

$$Sessid = f(T, D, IP, X_1, \dots, X_n),$$

где Sessid – идентификатор сессии, f – функция от $n + 3$ параметров, T – текущее время, D – текущая дата, IP – IP-адрес компьютера клиента, X_n – n -ный параметр.

Приведем пример. Пусть переменный код вычисляется по следующему алгоритму:

$$P = D * M + N . \text{pass} . Dw,$$

где P – переменный код, D – текущий день месяца, M – текущий месяц, N – текущий час, pass – кодовая фраза пользователя, Dw – номер текущего дня недели (например, для среды – это число 3).

Допустим, сейчас 16:02, вторник, 11 января 2005 года, а кодовой фразой является «СуперПаРоЛЬ». Тогда код будет формироваться так:

$$P = 11 * 1 + 16 . \text{«СуперПаРоЛЬ»} . 2$$

$$P = \text{«27СуперПаРоЛЬ2»}.$$

Сервер по этому же алгоритму вычисляет числовые части кода, вычленяет кодовую фразу, предположим, по алгоритму вычисляет значение функции MD5 от этой фразы (в данном случае – «8c8731f81316f9baf949fc3c5c995fc1») и сверяет с существующими записями хэш-функций кодовых фраз пользователей. После нахождения совпадения пользователь идентифицирован.

Рассмотрим былые уязвимости.

Компьютер пользователя и линии связи: пароль или кодовая фраза пользователя в явном виде отсутствует, т.е. не может быть перехвачена. В этом случае может быть узнан идентификатор сессии, что позволит воспользоваться уже установленной сессией с этого компьютера (при условии, что пользователь не выполнил операцию выхода из

системы, количество разрешенных действий пользователя на сервере не истекло, время действия сессии не истекло).

Хранилище данных: ключевая фраза в явном виде не содержится, есть только хэш-функция. При попытке изменения данных, касающихся времени действия сессии, IP-адреса и т.п., возникнет несоответствие данных, зашифрованных в идентификаторе сессии, что не позволит воспользоваться идентификатором со стороннего компьютера или с компьютера пользователя в другое время.

Веб-сервер: локальный доступ по-прежнему опасен, поскольку может открыть злоумышленнику алгоритмы работы системы, данные для доступа к базе данных, функции проверки ключевого кода пользователя и генерации идентификатора сессии. Кроме того, любой доступ к серверу с определенными привилегиями может позволить изменить алгоритм работы системы и удалить нужные файлы. Поэтому повлиять на эту уязвимость средствами рассматриваемой системы не является возможным в принципе.

Рассмотренный выше метод обработки данных, используемых для авторизации в системе, имеет следующие преимущества перед традиционными статичными парами «логин-пароль» с отправкой методами GET/POST по протоколу HTTP.

1. Пользователь вводит данные единожды, в одно поле, а не в два.
2. Доступ к базе данных не позволяет узнать ключевые слова пользователей.
3. Доступ к базе данных не позволяет продлить действия определенной сессии.
4. Локальный доступ к компьютеру пользователя не дает возможности узнать ключевой фразы.
5. Локальный доступ к компьютеру пользователя не дает возможности использовать известный идентификатор сессии при корректном выходе из системы или достижении установленных ограничений на количество операций за одну сессию и времени действия сессии.
6. Сканирование отправляемых данных с клиентского компьютера не позволяет узнать ключевое слово пользователя или конкретного пользователя из имеющегося списка.
7. Не требуется знания дополнительных языков программирования от создателей системы.
8. Не требуется установки дополнительных модулей и платформ, а также не требуется специальной настройки браузера пользователя.

В это же время описанный метод имеет недостатки:

1. увеличенные временные затраты по проектированию и разработке системы;
2. невозможность восстановления забытого ключевого слова пользователя – только установка нового.

Заключение

В статье рассмотрены вопросы угрозы утечки информации и модификации данных в системах, использующих хранилища данных и предоставляющих пользователям возможность удаленного доступа. Авторами предложен метод предотвращения несанкционированного чтения или изменения данных и структуры данных без применения дополнительных технических средств и дополнительного программного обеспечения, работающего на стороне клиента.

ОРГАНИЗАЦИЯ ПЕРСОНАЛЬНОГО ДОКУМЕНТООБОРОТА С ПОДДЕРЖКОЙ АССОЦИАТИВНЫХ СВЯЗЕЙ ПОЛЬЗОВАТЕЛЯ

В.А. Петров, В.В. Пovyшев, И.В. Шилов

Научный руководитель – к.т.н., доцент Н.Ф. Гусарова

В работе обсуждаются возможности внесения в формально параметризуемый информационный поток неформальной компоненты. Приводится пример программной реализации.

Практичность (usability) и эргономичность современного программного продукта (ПП) во многом лимитируются формализованностью взаимодействия пользователя и информационной системы [1].

Одно из направлений преодоления этого барьера сегодня связывается с построением обобщенной модели пользователя и ее последующей диверсификацией (возможно, параметрической). Например, для поисковых систем в качестве такого параметра предлагается использовать текущий контекст поискового запроса и/или положительно оцениваемых результатов поиска [2, 3]. Однако, как показывают исследования (см., например, [4, 5]), пертинентность поисковых систем – характеристика степени соответствия документа или данных, найденных в результате поиска, информационной потребности пользователя, выраженной в его запросе [6] – зависит от содержимого индексируемых данных косвенно, напрямую же она зависит от личных оценок пользователя, которые недоступны для индексации этими системами.

Еще одним шагом в этом направлении можно считать поддержку написания и выполнения сценариев. Скриптовые расширения присутствуют во многих ПП, например, Adobe Photoshop, 3Dmax, 1С–Предприятие. Однако возможности описания нестандартных ситуаций, не предусмотренных в исходной поставке ПП, здесь весьма ограничены.

Имеются попытки объединения рассмотренных подходов. Например, технология документооборота Magic Objects [7] предполагает дополнение традиционных реляционных структур данных, хранящихся в физических таблицах, уровнем специальных семантических объектов. Для каждого типа рабочих мест специфицируется информация, с которой он работает, и доступные ему операции над данными. При входе в систему пользователь авторизуется, и в соответствии с настройками его рабочего места ядро системы загружает в АРМ-навигатор доступные для него объекты с операциями. Потребности рабочих мест выявляются в процессе внедрения информационной системы через общий словарь в виде слоя семантических объектов. Однако на сегодняшний день возможности технологии ограничиваются разрешением синонимических коллизий (например, термины «счет», «дата поставки» и т.п. в различных подразделениях предприятия могут толковаться по-разному) в рамках стандартного тезауруса.

Организация персонального документооборота на произвольном тезаурусе связывается, в основном, с технологиями рокет РС [8]. Кроме поддерживаемых традиционных файлов форматов .doc и .xls, здесь сформировалось понятие «заметок» как интегрированное сочетание текста формата ASCII, рукописных символов и звука (в перспективе к ним, видимо, добавится изображение). При этом содержание заметки представляется в едином файле собственного формата. Аналогичный вариант документооборота с формальной интеграцией контента – технология электронных писем. Однако на сегодняшний день рассмотренные технологии не поддерживают семантическое объединение исходных информационных потоков.

Одним из путей преодоления формализованности взаимодействия пользователя и информационной системы видится возможность внесения в формально параметризуемый информационный поток неформальной компоненты. В [4] предложено реше-

ние поставленной задачи посредством определения эмоционального состояния пользователя аппаратными средствами, что, очевидно, сужает круг возможных его применений. На сегодняшний день представляется более реалистичным вводить контекстную информацию в систему в виде текста для последующей его индексации встроенной поисковой системой.

Для экспериментальной проверки предложенного решения реализован графический front-end к консольной утилите cdcontrol, написанный на языке Perl/Tk под операционную систему FreeBSD и функционально являющийся проигрывателем аудио компакт-дисков.

Запуск производится в графической среде из окна терминала. По нажатию на кнопку осуществляется формирование, выполнение и отображение консольной команды в окне терминала, а также занесение ее в файл ~/.bash_history. При запуске программы командная строка освобождается (становится доступной для дальнейшего диалога), и пользователь получает возможность занесения в нее вслед за отображенной командой личных комментариев, которые также заносятся в файл ~/.bash_history. По окончании работы файл содержит информацию о дорожках, поставленных на воспроизведение, и произвольный пользовательский текст, оформленный как комментарии к ним. Таким образом, осуществляется внесение в формальный поток (набор вызовов утилиты cdcontrol) неформальной компоненты (пользовательских комментариев).

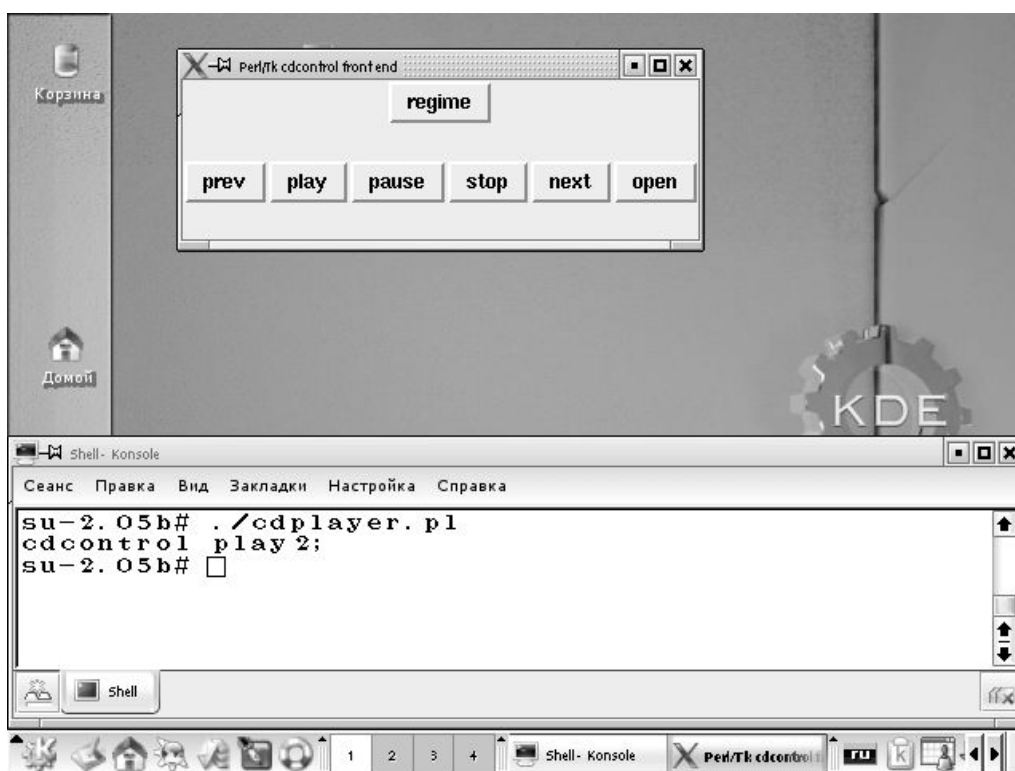


Рис. 1. Интерфейс программы

Экспериментальная проверка usability предложенного решения иллюстрируется рис. 2. Фокус-группа, репрезентирующая целевую аудиторию потенциальных пользователей, прослушивала различные дорожки компакт-диска, пользуясь GUI-интерфейсом. Посредством комментариев, записываемых на терминал, пользователи выражали свое отношение к воспроизводимым трекам диска, вследствие чего формировался файл ~/.bash_history, уникальный для каждого пользователя. Через неделю все

участники фокус-группы, пользуясь средствами утилиты `more` и базируясь на ранее введенных комментариях, осуществляли поиск в файле `~/.bash_history` и снова прослушивали те треки, которым ранее давали оценку. В результате 87% найденных по комментариям треков оправдали ожидания пользователей; предложенные принципы взаимодействия были положительно оценены 80% участников фокус-группы.

Описанное решение может найти применение при проведении маркетинговых исследований (сбор личных оценок одного и того же файла), ведении проектов по разработке ППО через Интернет, формировании совместного контекста группы пользователей внутри организации и т.д.

```
.bash_history+(~)- VIM - Shell - Konsole
Сеанс  Правка  Вид  Закладки  Настройка  Справка
mount_msdosfs /dev/ad0s3 /mnt/packages/
umount /mnt/packages/
mount_msdosfs /dev/ad0s3 /mnt/packages/
mount
cp -v /usr/home/spoiler/infospace_prj/*.bmp /mnt/packages/Unix_Windows/
cp -v /usr/home/spoiler/infospace_prj/cdplayer/cdplayer.pl /mnt/packages/Unix_Windows/
shutdown -r now
cat ~/.bash_history
cdcontrol play 2;
# моя любимая песня
cdcontrol play 3;
# неплохо
cdcontrol play 4;
# в целом понравилось
mount_msdosfs /dev/ad1s3 /mnt/usb/
ls /mnt/usb/
umount /mnt/usb/
mount_msdosfs /dev/ad1s2 /mnt/usb/
ls /mnt/usb/
```

Рис. 2. Фрагмент истории ведения команд (файл `~/.bash_history`)

Литература

1. Кудрин Р. Когнитивный инжиниринг // Компьютерра. 2004. № 539. – <http://www.computerra.ru/offline/2004/539/33273/>
2. Беляев Д.В. Ассоциативная модель смысловых контекстов и ее применение в задаче уточнения поисковых запросов // Электронный журнал Труды МАИ. 05.04.2005 – http://www.mai.ru/projects/mai_works/articles/num18/article9/article.zip
3. Прокудин А. Кто не спрятался, я не виноват // Компьютерра. 2004. № 567. – <http://www.computerra.ru/offline/2004/567/36689/>
4. Митилино С. Социальный ПК // Компьютерное Обозрение. №1-2. 17–23 января 2001 – <http://www.itc.ua/print.phtml?ID=5001>
5. Ландэ Д.В. Искать и не сдаваться // СНП. Май. 2004. – <http://www.visti.net/~dwl/art/sart/index.html>
6. Воройский Ф.С. Информатика. Новый систематизированный толковый словарь. М.: Физматлит, 2003. 760 с.
7. Костачук А.В. Технология Magic Objects 2002 – <http://www.magicobjects.i2x2.ru/index.htm>
8. Точка зрения: Проблема несовместимости представлений пользователей и возможностей КПК. <http://www.hpc.ru/lib/arts/2063>

**ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОГО ТЕСТИРОВАНИЯ
ДЛЯ ОСУЩЕСТВЛЕНИЯ ИТОГОВОГО КОНТРОЛЯ ЗНАНИЙ
СТУДЕНТОВ В ОБЛАСТИ ФИЗИКИ****Ю.В. Шлюжайте****Научный руководитель – д.ф.-м.н., профессор Ю.Л. Колесников**

Для проверки знаний студентов используется компьютерное тестирование. Оно позволяет избежать некоторых трудностей, сопутствующих традиционным методам контроля успеваемости. Была проведена работа по созданию банка тестовых заданий для осуществления итогового контроля знаний студентов II курса по физике (раздел «Оптика»).

Введение

Для контроля знаний студентов во многих высших учебных заведениях в действие постепенно вводится компьютерное тестирование по различным предметам – как естественнонаучного цикла, так и гуманитарного. Это становится возможным во многом благодаря практически повсеместному распространению компьютерных технологий, в частности, глобальной сети Интернет. Целью внедрения такого способа проверки знаний студентов является, прежде всего, облегчение нагрузки преподавателей, которые ранее вели огромную работу по организации, проведению и обработке результатов проверки знаний. Кроме того, автоматизированный контроль знаний позволяет унифицировать аттестационные требования и, как следствие, повысить объективность аттестации студентов.

Центр дистанционного обучения СПбГУ ИТМО уже не первый год практикует проведение компьютерного тестирования для обучающихся и аттестационных целей, что позволяет осуществлять контроль знаний учащихся по различным дисциплинам (в том числе и физики). В большинстве своем такой способ контроля знаний студентов использовался для текущего и входного контроля, но не для итогового, для которого предусматривались традиционные – зачет, экзамен.

Проверка знаний студентов в области физики

Впервые тестирование, правда, осуществленное без использования компьютеров, в качестве итогового контроля знаний в области физики (по разделам «Электричество и магнетизм») было проведено в период летней сессии 2003/2004 учебного года. Материал такого аттестационного зачета был основан на программе так называемого «теорминимума». Студент, сдавший успешно такой зачет, мог претендовать на оценку «удовлетворительно», а также при желании повысить оценку при сдаче традиционного экзамена своему лектору. Этот подход к осуществлению итогового контроля знаний, который был организован в качестве эксперимента, как альтернатива традиционному экзамену, во многом позволил достигнуть поставленных целей: во-первых, была облегчена работа преподавателей, а во-вторых, удалось избежать психоэмоциональной напряженности у студентов, которую они обычно испытывают при личном взаимодействии с преподавателем.

Компьютерное тестирование знаний в области физики

По перечисленным результатам было принято решение о расширении банка тестовых заданий по физике (в частности, раздела «Оптика») для их использования при

осуществлении итогового контроля знаний студентов в период зимней сессии 2004/2005 гг. уже с использованием компьютеров.

Следует отметить, что до того момента времени уже существовала и широко использовалась тестовая база с вопросами по отдельным темам раздела «Оптика» (таких как «Геометрическая оптика», «Интерференция», «Дифракция»), т.е. было возможно проведение только текущего контроля знаний студентов по упомянутым разделам оптики. Такое положение вещей во многом упростило задачу создания наиболее полного банка тестовых заданий, охватывающих все темы раздела «Оптика». Уже существующие вопросы по рассмотренным темам были подкорректированы и дополнены, а основная работа была направлена на подбор и разработку новых тестовых заданий по темам раздела «Оптика», которые ранее не были разработаны для использования при аттестации студентов путем компьютерного тестирования.

В результате в тестовую базу были добавлены вопросы по темам «Фотометрия», «Поляризация», «Дисперсия», «Тепловое излучение», «Лазеры». Всего добавлено около 200 новых тест-кадров (в основном – тестовые задания закрытой формы), что явилось вполне достаточным для формирования нужного количества вариантов, которые использовались при проведении итоговой аттестации студентов. Электронные тесты были созданы в полном соответствии с основными принципами построения пакета тестовых заданий, сформулированными Центром дистанционного обучения университета.

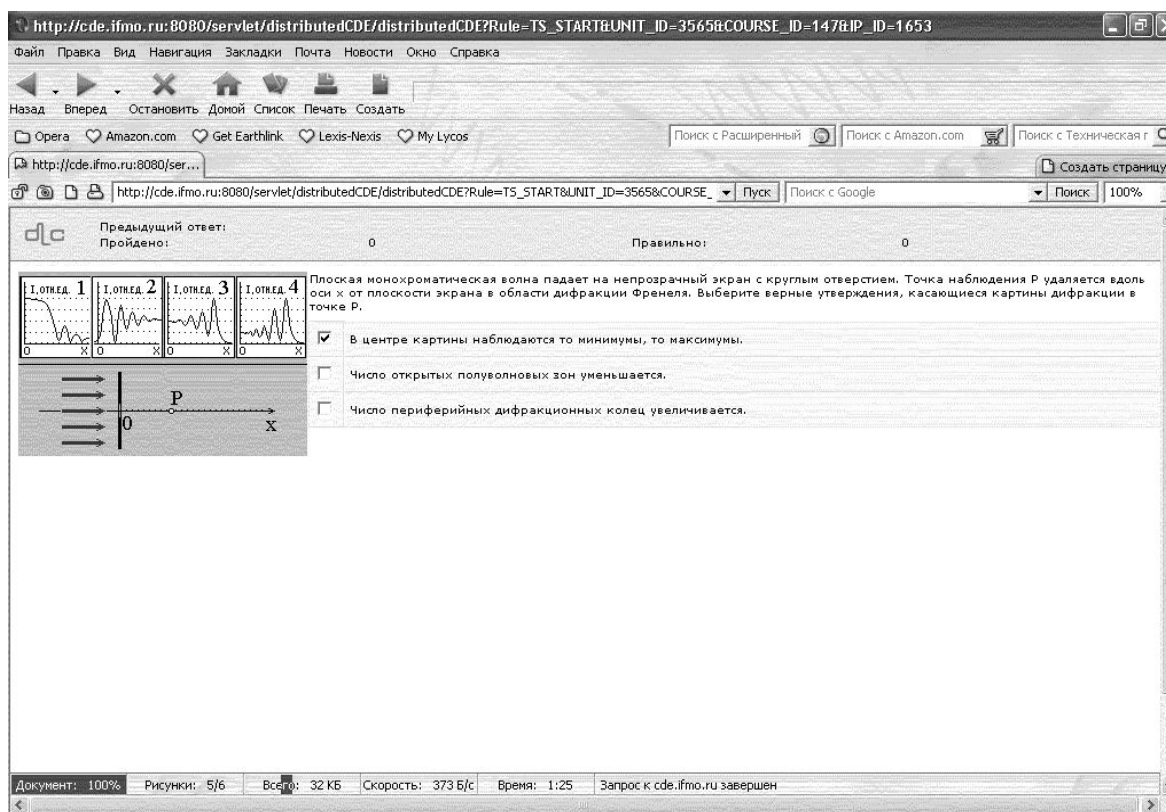


Рис. 1. Пример тестового задания по физике («Дифракция Френеля»)

Закключение

Доработанный банк тестовых заданий по физике охватил практически все темы раздела «Оптика». Использование электронных тестов по физике при осуществлении итогового контроля студентов университета в период зимней сессии 2004/2005 учебного года позволило не только наиболее адекватно оценить их знания в области физики, но также внести вклад в дальнейшее развитие системы тестирования университета.

Литература

1. Бобцов А.А., Лямин А.В., Чежин М.С. Современное состояние и перспективы развития системы тестирования знаний в системе дистанционного обучения для университета // Научно-технический вестник. 2001. №1. С. 23–27.
2. Колесников Ю.Л., Королев А.А., Стафеев С.К., Ярышев Н.А. Опыт применения компьютерных технологий в преподавании физики в Санкт-Петербургском государственном институте точной механики и оптики (техническом университете) // Физическое образование в вузах. 2001. Т.7. №3. С. 77–80.

РЕАЛИЗАЦИЯ КОНЦЕПЦИИ МНОГОУРОВНЕВОЙ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ НА БАЗЕ ИНТЕРНЕТ-ШКОЛЫ ПРОГРАММИРОВАНИЯ

М.А. Казаков

Научный руководитель – д.т.н., профессор В.Г. Парфенов

Рассмотрена концепция многоуровневой подготовки квалифицированных кадров в области информационных технологий на базе Интернет-школы программирования, реализуемая автором на технической базе кафедры «Компьютерные технологии». Кратко рассмотрены концептуальные и технические аспекты реализации.

Введение

С развитием информационных технологий широкое распространение получили системы дистанционного обучения. В настоящее время наблюдается интенсивный рост заинтересованности в таких системах как со стороны учебных заведений, так и со стороны учащихся. Причиной такого интереса является наличие автоматизации большей части учебного процесса и возможность обучаться на расстоянии, что снижает нагрузку на преподавателей и повышает доступность обучения [1, 2].

В настоящей статье рассматривается дистанционное обучение элементарной теории алгоритмов программирования [3, 4]. Основной проблемой при обучении программированию является необходимость тестирования практических навыков [5, 6]. Учебный материал должен содержать большое количество динамических иллюстраций «визуализаторов» [7].

В последнее десятилетие в нашей стране был разработан ряд дистанционных Интернет-систем, обеспечивающих информационную, методическую и учебно-тренировочную поддержку творческим конкурсам по информатике и программированию. Исторически системы дистанционной Интернет-поддержки одаренных и профильно мотивированных учащихся развивались по двум направлениям. Одно из них было связано с непосредственной поддержкой олимпиад по информатике, программированию и компьютерным технологиям. Эти системы предназначены для уже имеющих достаточно обширную базовую подготовку студентов и школьников. На соответствующих олимпиадных сайтах, как правило, представлены достаточно сложные олимпиадные задачи. Второе направление ориентировалось на учащихся, которые хотели получить начальную подготовку по базовым фундаментальным курсам компьютерных информационных технологий и в случае успешного освоения материала перейти на следующую образовательную ступень и стать пользователями олимпиадных сайтов. Сайты второго направления обычно назывались Интернет-школами или Интернет-курсами.

В настоящее время на технической базе кафедры «Компьютерные технологии» активно разрабатывается, апробируется и внедряется система дистанционного обучения под названием «Интернет-школа программирования» [8]. В первой части статьи рассматривается концепция многоуровневого обучения специалистов в области информационных технологий. Интернет-школа программирования объединяет в себе особенности обоих указанных направлений, поскольку в рамках одной системы объединяет как Интернет-школу, так и систему поддержки Интернет-соревнований. Во второй части дано краткое техническое описание реализации Интернет-школы, особенно архитектуры и используемых технологий.

Описание концепции

Подробное описание терминологии предметной области можно найти в статье [6]. Поэтому здесь мы не будем останавливаться на этом, а перейдем непосредственно к концепции и ее реализации в рамках Интернет школы программирования.

В основе многоуровневой системы подготовки специалистов лежит разделение процесса обучения на некоторое количество этапов. По окончании каждого этапа ученик выходит на качественно новый уровень подготовки. Условно каждый этап можно разделить на три стадии:

- получение новых знаний;
- освоение полученных знаний;
- контроль уровня освоенных знаний.

В рамках Интернет-школы предлагается схема, представленная на рис. 1.

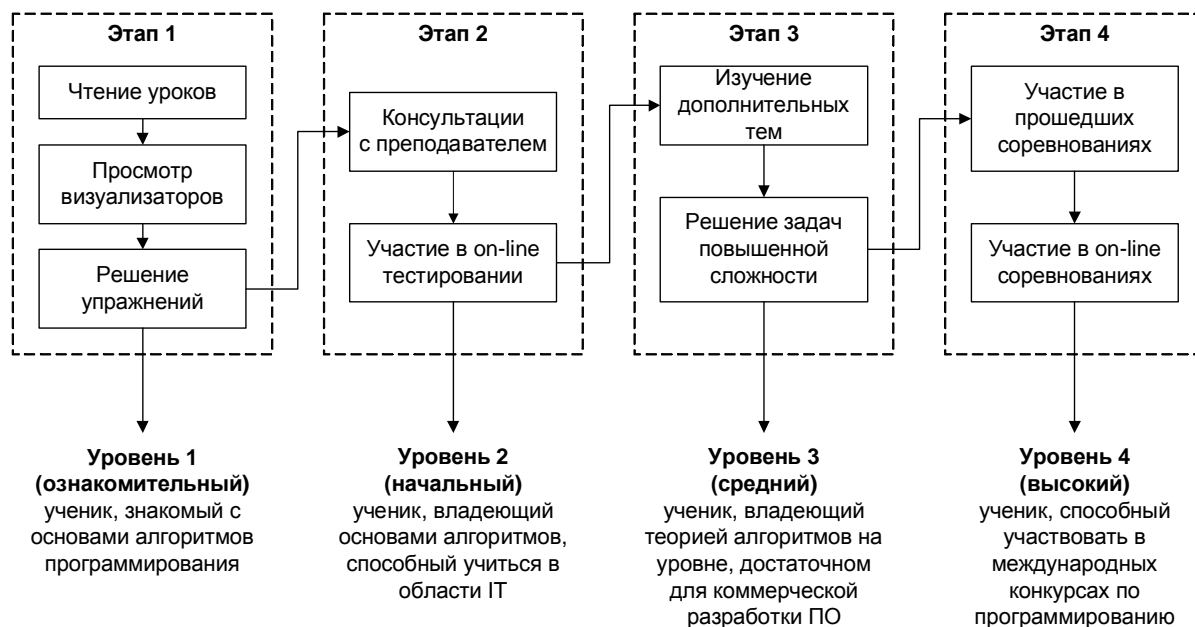


Рис. 1. Концепция 4-уровневого обучения теории алгоритмов

Рассмотрим отдельно каждый этап подготовки.

Этап 1 соответствует ознакомительному этапу подготовки. На этом этапе ученику доступны следующие возможности.

- Знакомство с учебным материалом. Материал представлен на сайте Интернет-школы программирования в форме HTML уроков, снабженных иллюстрациями и визуализаторами.
- Решение упражнений. В текстах уроков встречаются упражнения, предлагаемые для самостоятельной проработки.

Целью данного этапа является *профильная ориентация*. Среди артефактов данного этапа можно выделить:

- текст урока – словесное описание теоретических основ с примерами исходного кода, разборы упражнений, приведенных в предыдущих уроках;
- визуализаторы – динамические иллюстрации, представляющие выполнение алгоритма при определенных входных данных;
- упражнения – относительно небольшие задания для самостоятельной проработки учебного материала.

В процессе прохождения первого этапа ученик получает начальное представление об основах алгоритмов. При этом посредством самоконтроля при помощи системы автоматической проверки решений задач ученик способен оценить свои способности в области программирования.

Этап 2 характеризуется более детальной отработкой практических навыков ученика. Особенностью данного этапа является наличие более жесткого расписания обу-

чения и взаимодействие с преподавателем. На этом уровне ученик детально изучает учебные материалы с целью *научиться решать задачи*. Консультации преподавателя (тьютора) помогают ученику разобраться с учебным материалом. По окончании изучения разделов учебного курса ученик проходит on-line тестирования с целью определить уровень освоения материала. На данном этапе появляются следующие артефакты:

- обратная связь с преподавателем – реализуется в виде форума, переписки по e-mail;
- автоматическая проверка решений – технология, позволяющая проводить дистанционную проверку практических навыков посредством автоматической проверки решений задач. В данном случае Интернет-школа программирования обеспечивает техническую базу для проведения таких тестирований;
- групповые тестирования – специальные тестирования, организуемые в Интернет-школе программирования с ограниченными временными рамками. Результаты этих тестирований рассматриваются на предмет выявления абитуриентов способных к обучению в области IT.

На рассматриваемом этапе начинает играть роль концепция *персонализации* [9] ученика в системе. В результате успешного прохождения этого этапа ученик владеет основами алгоритмов и готов к поступлению в вуз по специальности в области IT.

Уровень 3 подразумевает знание расширенного набора алгоритмов и умение их применять. На этом этапе ученик переходит в другой раздел единой Интернет-системы – архив задач. В этом разделе представлено множество задач различной степени сложности, автоматическая система проверки решений проверяет правильность решения задач. Для решения многих задач требуется воспользоваться справочной системой, включающей в себя разборы задач с подробными описаниями стандартных алгоритмов и способами их применения. Целью данного этапа является *выработка практических навыков* использования элементарной теории алгоритмов. На этом этапе появляются новые артефакты:

- справочные материалы – набор решений и комментариев к различным типовым задачам;
- архив задач – структурированная система задач с различных олимпиад, снабженная набором тестов.

Следует отметить, что основой обучения на этапах 3 и 4 является обучение через решение задач.

Уровень 4 (необязательный) подразумевает *расширенные знания алгоритмов*, умение быстро и эффективно применять их. На этом этапе формируются такие навыки, как:

- быстрота реализации алгоритмов;
- умение реализовывать алгоритмы с наименьшими затратами на тестирование;
- умение быстро локализовать и устранить ошибку в готовом тексте программы.

Этот этап можно, с одной стороны, рассматривать как подготовку к участию в олимпиадах по программированию, а с другой стороны – как высший уровень подготовки специалиста IT в области теории алгоритмов. На этом этапе появляются следующие артефакты.

- Исторические соревнования – соревнования, результаты которых хранятся на сайте Интернет-школы программирования в разделе олимпиадного on-line сервера. Любой ученик может принять участие в таких виртуальных соревнованиях на правах участника, тем самым попробовать свои силы не только в задачах повышенной сложности, но и в условиях ограниченного времени.
- On-line соревнования – эти соревнования проводятся на основе автоматической проверочной системы, которая в этом случае не только проверяет решения, но и подводит итоги соревнований, как это происходит в случае on-line тестирований. Следует отметить, что в последние годы система проведения on-line соревнований

используется для проведения официальных четвертьфинальных соревнований чемпионата мира по программированию.

Контрольными испытаниями на этом этапе является участие в соревнованиях наряду с другими участниками (бывшими учениками).

Таким образом, четырехуровневая система подготовки квалифицированных кадров в области информационных технологий позволяет поэтапно, шаг за шагом, пройти все этапы – от ознакомления с предметом элементарной теории алгоритмов до высококвалифицированного специалиста в области IT. Следует отметить, что подготовка в области технологий и других областей IT выходит за рамки данной концепции.

Технические аспекты реализации

Для описания технических подробностей реализации проведем классификацию и группировку артефактов Интернет-школы программирования с целью описания реализации каждой группы артефактов в отдельности.

1. Статические материалы: уроки, упражнения, задания, справочные материалы.
2. Визуализаторы алгоритмов.
3. Задачи для проверки практических навыков, разработка тестов.
4. Архив задач.
5. Система автоматической проверки решений.
6. Система проведения соревнований и тестирований.

Поскольку все справочные материалы находятся в общем доступе на http-сервере, то они оформлены в виде статических HTML файлов.

Для формирования уроков используется паттерн *представление с преобразованием* (Transform view) [10]. При этом для увеличения скорости обработки запросов преобразование производится не динамически, а статически. Такой подход позволяет хранить исходные данные для уроков в XML, в то время как пользователю виден результат преобразований в виде HTML. Для формирования уроков используется специальный генератор уроков. Схема действия генератора представлена на рис. 2.

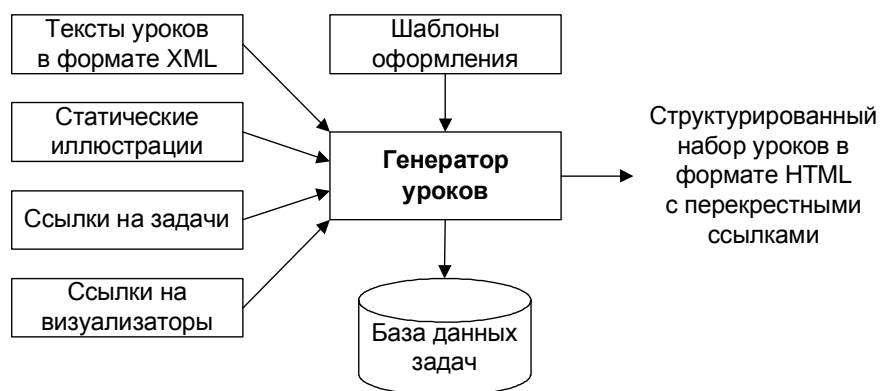


Рис. 2. Схема действия генератора уроков Интернет школы программирования

Необходимость в генераторе уроков возникла в результате опытных исследований. Основная проблема в ручном редактировании уроков была связана с тем, что авторы уроков часто вносят исправления, что приводит к огромному количеству ручной работы по поддержанию стиля форматирования. Поэтому был выбран формат XML как универсальный формат описания данных, в том числе и текстовых. В результате такой унификации все правки вносятся в XML документы, а далее все уроки, задействованные в правке, пересоздаются. Дополнительным удобством автоматической генерации является отсутствие в результирующих уроках лишних пробелов, поскольку генератор

при создании HTML документов делает их максимально компактными. При автоматической генерации уроки и задания автоматически нумеруются, а ссылки на них автоматически обновляются. Таким образом, автоматическая генерация существенно экономит время и уменьшает количество ошибок в тексте уроков.

Визуализаторы алгоритмов являются одной из особенностей Интернет-школы программирования. История развития технологии создания визуализаторов, а также наиболее современная технология их создания подробно описана в работе [11]. Здесь же кратко отметим основные аспекты построения визуализатора.

Визуализатор – это программа, иллюстрирующая выполнение алгоритма при определенных входных данных. Применительно к дискретной математике и программированию визуализаторы обычно моделируют некоторые алгоритмы, давая возможность обучающемуся при помощи интуитивно понятного интерфейса проходить алгоритм шаг за шагом от начала до конца, а при необходимости – и обратно.

Визуализатор «по-крупному» предлагается строить следующим образом:

- по программе, реализующей визуализируемый алгоритм, строится автомат логики визуализатора (контроллер – controller);
- выбираются визуализируемые переменные (модель – model);
- проектируется формирователь изображений и комментариев, который преобразует номер состояния и соответствующие значения визуализируемых переменных в «картинку» и поясняющий текст (представление – view).

Такая конструкция визуализатора соответствует одному из основных паттернов проектирования объектно-ориентированных программ, который обозначается аббревиатурой MVC (Model–View–Controller) [10]. Для реализации логики визуализатора используется формальное преобразование программы, реализующей визуализируемый алгоритм, в конечный автомат.

Разработка учебных задач для Интернет-школы программирования состоит из трех этапов.

- Разработка условия задачи.
- Разработка тестов к задаче.
- Регистрация задачи в системе автоматической проверки.

Технология разработки тестов основывается на теории тестирования программного обеспечения. Специфика разработки тестов для автоматической проверки решений задач детально рассматривается в [5]. Здесь лишь отметим, что разработка тестов состоит из следующих этапов.

1. Написание верного решения задачи.
2. Написание условно верных решений задачи.
3. Рассмотрение возможных ошибок.
4. Написание системы тестов:
 - a. тест из примера;
 - b. набор тестов на проверку граничных условий;
 - c. тесты на отсеечение ошибок;
 - d. тесты на отсеечение условно верных решений.
5. Составление проверочной программы.

При регистрации задачи в автоматической проверочной системе каждой задаче сопоставляется специальный уникальный идентификатор, по структуре схожий со структурой DNS – стандарта соответствия между IP адресами и уникальными именами компьютеров в сети Интернет. Примером такого идентификатора может служить, например, следующая строка:

```
ru.ifmo.ips.Algorithms.lesson1.problem1
```

В данном случае это задача с сайта Интернет-школы программирования, глава «Алгоритмы», урок №1, задача №1.

Прежде чем рассматривать оставшиеся части программного комплекса Интернет-школы программирования, остановимся на общей схеме PCMS2web, изображенной на рис. 3 [6]. Система PCMS2web выполняет основную часть всех функций поддерживающих учебный процесс и проведение соревнований. Система PCMS2web реализована в виде web-приложения.

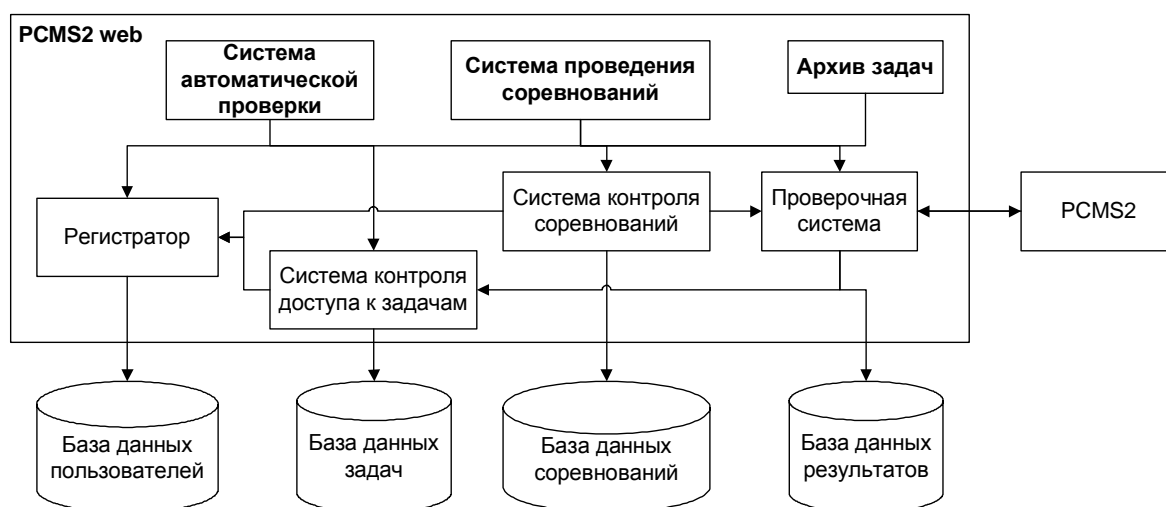


Рис. 3. Схема взаимодействия подсистем PCMS2web

Как видно из иллюстрации, PCMS2web условно можно разделить на три уровня.

- **Уровень базы данных** (база данных пользователей, база данных задач, база данных соревнований).
- **Уровень бизнес логики** (регистратор, система доступа к задачам, система контроля соревнований, проверочная система, PCMS2 – проверочное ядро).
- **Уровень представления** (система автоматической проверки, система проведения соревнований, архив задач).

Рассмотрим подробнее реализацию каждой из *компонент бизнес-логики*, которая реализована с использованием технологии сервлетов.

Регистратор реализует логику персонализации пользователя в рамках PCMS2web. К функциональности регистратора относится:

- авторизация;
- хранение в сессии web-сервера данных о пользователе;
- проверка прав пользователя на доступ к отдельным страницам;
- сохранение и отображение информации о пользователе.

Система контроля доступа к задачам отвечает за ограничение доступа к задачам и условиям задач. Условия задач находятся в структуре каталогов, не входящей в структуру web-сервера. Основная функция этой системы – это ограничение доступа к условиям задач. Условия выдаются только в случае наличия разрешения у пользователя прав на просмотр задачи. Это позволяет во время тестирований и соревнований сохранять тайну условий до начала соревнований.

Система контроля соревнований отвечает за ведение процесса тестирований и соревнований. К функциям этой системы относятся:

- хранение статистики подходов по конкретным задачам, а также статистики в рамках текущих для пользователя соревнований.
- создание/удаление/переключение сессий (участия пользователя в соревновании), а также начало и окончание тестирования.
- формирование таблицы результатов по сессии или по соревнованиям.

Проверочная система отвечает за проверку решений и запись результатов решения в базу данных. К функциям проверочной системы относятся:

- помещение решение в очередь на проверку;
- выдача решений PCMS2 kernel на проверку и получение результатов.

Проверка решения осуществляется на основе пакета, который содержит:

- идентификатор задачи;
- язык программирования;
- текст исходного кода для одного или нескольких файлов.

Компоненты уровня представления выполняют функции взаимодействия с пользователем. Пользовательский интерфейс реализован с использованием технологии JSP, т.е. реализован паттерн *представление по шаблону* (Template view)[10].

Система автоматической проверки выполняет следующие функции:

- по результатам взаимодействия с пользователем формирует пакет с заданием для проверочной системы;
- выдает пользователю статистику по проверке решений зада. Для каждого решения задачи выдается
 - название задачи,
 - время решения,
 - результат,
 - исходный код.

Архив задач выполняет функции поиска задач по соревнованиям, теме, названию.

Система проведения соревнований реализует пользовательский интерфейс к системе контроля соревнований и все ее функции.

Система реализована, функционирует и продолжает совершенствоваться на технической базе кафедры «Компьютерные технологии». Следует отметить, что к настоящему моменту Интернет-школа программирования содержит три учебных курса, количество проверенных решений – около 30000, из них около 7000 – верные решения, кроме того, система проведения соревнований использовалась для проведения 10 официальных соревнований – этапов чемпионата мира по программированию.

Заключение

В работе рассмотрена концепция многоуровневой подготовки квалифицированных кадров в области информационных технологий в применении к Интернет-школе программирования. Рассмотрены комплекс программных и технологических решений позволяет утверждать, что в рамках Интернет-школы программирования реализована четырехуровневая система подготовки в области элементарной теории алгоритмов.

Литература

1. Подготовка и проведение учебных курсов в заочно-дистанционной форме обучения. / Под редакцией профессора ЦИКИНА И.А. СПб: Изд-во СПбГТУ, 2000.
2. Агапонов С.В., Джалиашвили З.О., Кречман Д.Л., Никифоров И.С., Ченосова Е.С., Юрков А.В. Средства дистанционного обучения. Методика, технология, инструментарий. / Под ред. Джалиашвили З.О. СПб: БХВ-Петербург, 2003. 336 с.
3. Васильев В.Н., Елизаров Р.А., Парфенов В.Г., Столяр С.Е. Организация дистанционного обучения программистов. // Телематика'98. Всероссийская научно-методическая конференция. Санкт-Петербург, 8-11 июня 1998 г.: Тез. докл. С. 172–173.
4. Казаков М.А., Осипова Т.Г., Парфенов В.Г., Столяр С.Е. Интернет-школа программирования в СПбГИТМО. // Телематика'99. Всероссийская научно-методическая конференция. Санкт-Петербург, 7-10 июня 1999 г.: Тез. докл. С. 165–166.
5. Столяр С.Е., Казаков М.А. Подготовка тестов как часть технологии автоматизированного тестирования. // XXX науч.-техн. конф. проф.- преподавательского состава 25-28 января 1999 года. Тез. докл. СПб: СПбГИТМО(ТУ), 1999. С.105.

6. Казаков М.А. Система автоматического тестирования программных решений и проведения соревнований в режиме on-line. // Вестник конференции молодых ученых СПбГУ ИТМО. Сборник научных трудов 2004. СПб: СПбГУ ИТМО, 2004. С. 181–189.
7. Казаков М.А., Столяр С.Е. Визуализаторы алгоритмов как элемент технологии преподавания дискретной математики и программирования. // Телематика'2000. Международная научно-методическая конференция. Санкт-Петербург, 29 мая – 1 июня 2000 г.: Тез. докл. С.189–191.
8. Интернет-школа программирования <http://ips.ifmo.ru/>
9. Udi Manber, Ash Patel, and John Robison. Experience with Personalization on Yahoo! – Communications of the ACM, August 2000. Vol.43, No. 8. P. 35–39.
10. Фаулер М. Архитектура корпоративных программных приложений.: Пер. с англ. М.: Издательский дом «Вильямс», 2004. 544 с.
11. Казаков М.А. Использование автоматного подхода для построения визуализаторов. // Вестник конференции молодых ученых СПбГУ ИТМО. Сборник научных трудов 2004. СПб: СПбГУ ИТМО, 2004. С. 166–180.

ПРИНЦИПЫ ПЕРЕНОСА ОБРАЗОВАТЕЛЬНЫХ INTERNET-РЕСУРСОВ НА КАРМАННЫЕ КОМПЬЮТЕРЫ

Р.В. Койнов

Научный руководитель – к.т.н., доцент Л.С. Лисицына

Рассматриваются общие принципы переноса образовательных Internet-ресурсов на карманные компьютеры. Подробно рассмотрены основные этапы переноса, проиллюстрированы примеры типовых ошибок и недочетов.

Введение

При создании электронного учебного пособия (ЭУП) для карманного компьютера (КПК) можно руководствоваться общими правилами, которые были разработаны и неоднократно были опубликованы [1]. В создании компьютерных учебных материалов (в идеальном случае) должны принимать участие три человека: «программист», ответственный за их реализацию (графика, звуковые эффекты, анимация, программирование), «автор», ответственный за содержательную сторону проекта (исходные тексты, сценарий, тестирование), и «педагогический дизайнер», ответственный за практическую эффективность разработки. Разработка ЭУП для КПК «с нуля» подчиняется тем же правилам, однако этот процесс трудоемкий и достаточно длительный по времени. Создателям ресурса также необходимо будет учитывать, что, в отличие от настольных компьютеров, на рынке КПК присутствуют две основные платформы, которые делят его примерно поровну – это PalmOS и различные вариации Windows Mobile. Особенностью этого баланса является то, что эти платформы построены на совершенно различных принципах, и ПО, разработанное для одной платформы, невозможно использовать с другой. Наиболее оптимальным вариантом является использование уже готовых сетевых образовательных ресурсов, адаптированных для использования с той или иной платформой.

Когда ресурс не создается с нуля, а переносится на другую платформу, авторский коллектив имеет дело с уже готовым материалом, поэтому функции автора сводятся к минимуму, что позволяет сэкономить время и ресурсы.

Перенос образовательных ресурсов на КПК

Процесс создания электронного учебного пособия для КПК, созданного на основе Internet-ресурса, можно разделить на несколько этапов:

1. описание целей и условий обучения;
2. предварительная обработка ресурса (отбор материала, его структурирование и формализация);
3. подготовка бета-версии учебных материалов;
4. оценка учебных материалов и их доработка по результатам оценки;
5. сопровождение и развитие учебных материалов.

На каждом из выделенных этапов готовится очередная часть учебных материалов, проводятся их оценка (авторами и/или заказчиком) и необходимая корректировка.

1. Описание целей и условий обучения

Первый этап обычно разбивают на три шага:

- оценка возможности создания ЭУП для КПК на основе Internet-ресурса;
- выбор формы реализации учебного пособия;
- описание учебных задач и уточнение целей обучения (детализация, операционализация).

Перечень ключевых вопросов, направляющих работу на каждом из трех шагов, приведен в табл. 1. Рассмотрим каждый из них в отдельности.

Выполняемый шаг	Ключевой вопрос
Оценка возможности создания ЭУП для КПК на основе Internet-ресурса	Какой процент информации будет потерян или потребует значительных изменений при переносе на КПК?
Выбор формы реализации учебного пособия	В какой форме будет представлен материал на КПК? Какие еще учебные и методические материалы понадобятся?
Описание учебных задач и уточнение целей обучения (детализация, операционализация)	Какие методические цели должны быть достигнуты с помощью ЭУП созданного для КПК?

Таблица 1. Три шага, выполняемых в ходе первого этапа

Оценка возможности создания ЭУП для КПК на основе Internet-ресурса. Приступая к созданию учебного ресурса, необходимо убедиться, что информация на Internet-ресурсе, который будет служить основой для пособия, представлена в форме, имеющей аналоги на платформе КПК. Выше было рассмотрено, что не всю информацию из Internet можно корректно отображать на экране «наладонника», например, Java-апплеты не могут корректно обрабатываться ни одной из основных платформ КПК. Кроме того, следует заранее определить коррективы, которые нужно будет внести в ресурс в ходе предварительной подготовки ресурса. Вполне может оказаться, что выгоднее по затратам будет не перенос готового Internet-ресурса, а создание нового, с нуля, в основе исходных данных которого будут лежать информация с трансформируемого ресурса.

Результатом выполнения первого шага являются:

- аргументированное решение о разработке электронного учебного пособия;
- набор рекомендаций, которые нужно будет учесть в процессе обработки ресурса;
- перечень коррективов, которые необходимо будет внести в ресурс в процессе переноса на платформу КПК.

Выбор формы реализации учебного пособия. Очевидно, что существует несколько форм реализации учебного пособия для КПК, созданного с применением Internet-технологий. Авторам нужно будет решить, насколько выбор той или иной формы оправдан в конкретном случае. В результате формируется перечень показателей, влияющих на выбор:

- опыт использования подобных форм реализации материалов в данной предметной области;
- содержание изучаемого материала.

Результатом выполнения второго шага является план создания комплекта учебных материалов с описанием его состава и структуры, организационных форм реализации образовательных ресурсов.

Описание учебных задач и уточнение целей обучения. Начнем с того, что цели обучения – это то поведение, те знания, умения и навыки, которые должен продемонстрировать обучаемый, чтобы его можно было назвать «компетентным». Цель обучения можно назвать хорошо определенной, если учащийся понял то, что от него хочет автор ресурса. Самый простой и эффективный способ – это проверить, позволяет ли предложенная формулировка цели ответить на три вопроса:

- что именно сможет делать обучаемый? (исполнение),
- при каких условиях он это сможет делать? (условия),
- насколько хорошо он это сможет делать? (критерий).

Если на каждый вопрос следует явный ответ, значит, цель определена хорошо. К методическим целям, которые могут быть успешно реализованы с использованием ЭУП, можно отнести:

- индивидуализация и дифференциация траектории обучения (например, за счет возможности поэтапного продвижения к цели по линиям различной сложности);
- осуществление обучающимся самоконтроля и самокоррекции (выполнение заданий тестов, анализ результатов и ошибок);
- тренировка в усвоении учебного материала и самоподготовка обучающихся;
- компьютерная визуализация учебной информации (различные интерактивные модели имитаторы);
- создание и использование информационных баз данных, необходимых в учебной деятельности, и обеспечение доступа к информационно-справочным базам;
- оптимизация деятельности учащегося;
- формирование культуры учебной деятельности.

Результатом выполнения третьего шага является иерархия целей обучения и их операционализация.

Результатом выполнения работ первого этапа является комплект материалов анализа целей, описание условий обучения с помощью создаваемых ресурсов для КПК и выбранных средств реализации.

2. Предварительная обработка ресурса

После того, как определены цели и желаемые результаты разработки, можно приступить к предварительной обработке материала Internet-ресурса. Дальнейшие действия определяются тем, какую форму реализации учебного пособия выбрал авторский коллектив.

Конспекты. Возможны две реализации учебных продуктов этой группы.

- Текст (текстовые массивы со вставленными графическими иллюстрациями). Такая форма подойдет для ресурсов, в которых не используются часто формулы, мелкие значки, символы других алфавитов, кроме латинского и кириллицы. При создании продуктов такого вида в первую очередь стоит акцентировать внимание на тех элементах, которые будет невозможно отобразить на экране КПК или они будут значительно искажены.
- Слайд-шоу. Этот вид конспектов представляет собой набор статических слайдов-иллюстраций. Такая форма реализации позволяет корректно отображать на экране КПК формулы, математические символы, буквы «нестандартных» алфавитов.

При выборе текстовой формы реализации конспекта для КПК в первую очередь необходимо отсечь медиа-элементы страницы, которые невозможно перенести в эту форму электронного пособия для КПК (Flash, Java и т.д.). Это можно сделать с помощью любого из редакторов HTML документов, даже используя стандартные средства Microsoft Office. Далее следует убрать ненужные графические элементы оформления – банеры, декоративные «шапки», ссылки на посторонние ресурсы. Вполне может понадобиться изменить дизайн навигационной панели переносимого ресурса, так как она может некорректно «вписаться» в экран КПК.

Следует также учесть, что выбранные средства просмотра и технические возможности КПК не позволяют комфортно просматривать информацию, структурированную в таблице. При переносе Internet-ресурса, возможно, потребуется преобразовать таблицы в такую форму, которая смогла бы верно донести до пользователя необходимую информацию.

При выборе в качестве формы реализации конспекта для КПК слайд-шоу исходная информация подвергнется более значительной переработке. Как уже указывалось, основой ресурсов этой группы является статическая иллюстрация – слайд. Процесс

предварительной обработки будет заключаться в создании последовательности этих слайдов. Для этой операции можно воспользоваться различными виртуальными принтерами, которые преобразуют текстовую информацию в набор графических файлов, либо воспользоваться технологией, разработанной в ходе этой работы. При переносе следует учесть следующие особенности реализации такой формы конспектов для КПК.

- У некоторых слайдов может варьироваться длина (при этом ширина останется без изменений). Это обусловлено тем, что часть материала не сможет поместиться в рамки стандартного размера экрана (в первую очередь это касается различных графиков и таблиц). Таким образом, часть слайдов станет «длиннее». Однако использование горизонтальной (а не стандартной вертикальной) прокрутки ресурса позволит свести к минимуму возможные неудобства, возникшие в результате этого.
- Следует быть готовым, что придется создавать слайды с различными размерами. Это вызвано тем, что в настоящий момент для КПК существуют два стандартных размера экрана. При ориентации только на один из размеров неизбежны потери качества отображения информации при просмотре ресурса на КПК, экран которого не соответствует типовому. Следовательно, чтобы охватить максимально возможную аудиторию пользователей, необходимо предусмотреть наличие версий конспекта адаптированных для различных размеров экранов КПК.

Такая форма подготовки ресурсов позволяет использовать различные шрифты для отображения текста, более точно соотносить размеры формул и текста, избежать проблем, когда КПК не понимает кодировку документа (например, отсутствует русская локализация). Оптимальной такая форма отображения будет для текстов, которые содержат большое количество формул, курсивные элементы форматирования, различные шрифты.

Модели. В настоящее время в Internet основная часть интерактивных моделей представлена в виде Java-апплетов или Flash-роликов. Естественным было бы предположить возможность переноса на КПК не только текстового содержимого сайта.

Как уже рассматривалось выше, для различных платформ КПК реализован просмотр и Java-, и Flash-ресурсов, однако ряд технологических задач, которые необходимо будет решить для корректного переноса ресурсов, сделает невыгодным перенос некоторых из них. В первую очередь это касается интерактивных моделей реализованных с применением Java-технологий. Ввиду того, что для платформы Windows не существует официальной Java-машины (компания Microsoft заменила её собственной технологией .NET), невозможен перенос моделей простым копированием на КПК. Можно отметить, что, несмотря на отсутствие «фирменной» поддержки, существуют виртуальные Java-машины сторонних фирм-разработчиков для этой платформы, которые позволяют запускать программы написанные на J2ME. Для платформы Palm существует виртуальная Java-машина от Sun Microsystems, однако она также реализована в спецификации J2ME.

Диаметрально противоположным образом обстоит дело с интерактивными Flash-роликами. Существует официальный проигрыватель роликов от компании Macromedia для Windows платформ КПК. Для PalmOS существуют наработки сторонних разработчиков, которые реализованы только для старших платформ этой серии (т.е. для 5 версии PalmOS и выше). Эти реализации обладают некоторыми ограничениями на обработку action script подпрограмм.

Следует, однако, заметить одно несомненное преимущество интерактивных Flash-роликов перед Java-моделями. Это отсутствие необходимости промежуточной обработки (перекомпиляции) переносимых ресурсов. Можно просто скопировать ролики на КПК любой платформы, и их сразу же можно будет проигрывать (с некоторыми ограничениями для Flash-проигрывателей сторонних разработчиков). Еще одним достоинством этой технологии является простота взаимодействия между клипами, т.е. для до-

бавления новой информации можно просто создать клип с недостающей информацией и прицепить его к уже имеющемуся. Эта особенность Flash-технологий избавляет авторов от необходимости иметь на руках исходную программу интерактивной модели.

Результатом выполнения работ второго этапа является наличие информации, которая готова к переносу на платформу КПК без потери качества.

3. Подготовка бета-версии учебных материалов

После второго этапа мы будем иметь на руках материалы, подготовленные для переноса на платформу КПК. При подготовке бета-версии учебного пособия для КПК, созданного на основе Internet-ресурса, следует в первую очередь обратить внимание на тщательную проработку следующих параметров:

- подготовка иллюстраций (фотографий, рисунков, скетчей, анимационных эффектов и пр.). В основном это касается учебных пособий, созданных на основе текстовых конспектов. Необходимо учитывать, что иллюстрации, которые выглядели на экране монитора компьютера великолепно, на экране КПК могут выглядеть совершенно иначе. Маленькие текстовые комментарии, неконтрастные линии и т.д. могут просто потеряться при отображении;
- выбор кодировки текста. Кириллические символы алфавита – одни из немногих в Internet, которые имеют несколько кодировок для отображения. При подготовке и последующем просмотре Internet-ресурсов на настольном компьютере проблем обычно не возникает, так как на нем предусмотрена возможность просмотра документов всех видов кодировок. Ограниченность ресурсов КПК не поддерживает подобный подход к работе с документами с различными кодировками – слишком «затратно».

Главная задача разработчиков на этом этапе — убедиться, что электронные учебные материалы реализуются в полном соответствии с поставленными целями. По мере того, как ресурс будет компоноваться, разработчики могут заметить отдельные недочеты или просчеты, допущенные ими на предыдущем этапе. Иногда эти недочеты можно исправить на этапе создания бета-версии, оперативно внося соответствующие изменения.

Результатом выполнения третьего этапа является бета-версия электронных учебных материалов.

4. Оценка учебных материалов и их доработка по результатам оценки

Бета-версия электронных учебных материалов завершена. Прежде чем передать подготовленный материал обучаемым, надо провести его оценку, дабы убедиться, что учебное средство выполняет свое назначение.

Один из способов такой оценки – экспертиза. Экспертиза сравнительно дешева, ее достаточно просто организовать, она не занимает продолжительного времени. Однако качество экспертной оценки существенно зависит от квалификации и качества работы приглашенных экспертов.

Более надежный путь – педагогический эксперимент, в ходе которого электронные учебные материалы используют для подготовки группы обучаемых. Если эксперимент достаточно полно воспроизводит условия будущего использования электронных учебных материалов, он может дать полную и достоверную информацию о сильных и слабых сторонах разработанного программного продукта.

Оценка разработанных учебных материалов с рекомендациями по ее доработке и/или особенностям использования служит основанием для подготовки рабочей (итоговой) версии продукта.

Результатом выполнения четвертого этапа является рабочая версия электронных учебных материалов.

5. Сопровождение и развитие учебных материалов

Разработка электронных учебных материалов для КПК успешно завершена. Окончательная рабочая версия передана потребителю, который использует ее по назначению. Теперь встал вопрос оценки того, насколько практична и полезна выполненная разработка. Если учебные материалы непрактичны и не выполняют связанных с ними ожиданий, о них достаточно скоро забудут, и их не придется сопровождать. Однако если было создано действительно полезное учебное средство, авторскому коллективу придется не только время от времени решать мелкие технические вопросы, возникающие по ходу их эксплуатации, но неоднократно возвращаться к их содержанию, выпускать необходимые дополнения, специальные редакции и т.д.

Результат выполнения работ пятого этапа – новые рабочие версии электронных учебных материалов.

Заключение

В результате проведенных работ были разработаны принципы и методики, которые позволяют перенести на КПК образовательные Internet-ресурсы, что позволит расширить сферу применения «наладонников», не ограничивая возможность их использования только в качестве органайзера.

Литература

1. Уваров А.Ю. Создание учебных материалов с учетом принципов эффективного учения. // Информатика. 2003. №30. С. 30–52.

СТРУКТУРА ФЕДЕРАЛЬНОГО ЗАКОНОДАТЕЛЬСТВА РФ В ОБЛАСТИ ЗАЩИТЫ ИНФОРМАЦИИ

Е.А. Проценко

Научный руководитель – д.т.н., профессор Л.Г. Осовецкий

На современном этапе развития информационного общества и его теснейшего взаимодействия с интересами личности и государства необходимо обеспечить комплексную правовую защиту информации на государственном уровне. Первые шаги на этом пути уже сделаны.

Основной вектор развития нашего законодательства на сегодняшний день определен не только на государственном, но и на международном уровне. В частности, 22 июля 2000 года в Окинаве была подписана Хартия глобального информационного общества [1], которая прежде всего призвана обеспечить как на государственном, так и на частном уровне ликвидацию международного разрыва в области информации и знаний. Всемирная встреча на высшем уровне по вопросам информационного общества, проходившая с 10 по 12 декабря 2003 года в Женеве, определила, что «... к 2005 году все страны должны разработать всеобъемлющие, перспективные и устойчивые национальные электронные стратегии. Ведущая роль в этом процессе должна принадлежать органам государственного управления. Частный сектор и гражданское общество, в диалоге с органами государственного управления, должны сыграть важную консультативную роль в формировании национальных электронных стратегий» [2]. Аналогичная встреча планируется уже в этом 2005 году в Тунисе.

Развивая положения Окинавской хартии, была принята Доктрина информационной безопасности от 9 сентября 2000 г. № Пр-1895[3], на «стержне» которой выстроился целый каркас нормативно-правых актов, в определенной степени обеспечивающих правовую защиту информации, соподчиненность которых показана на рис. 1.

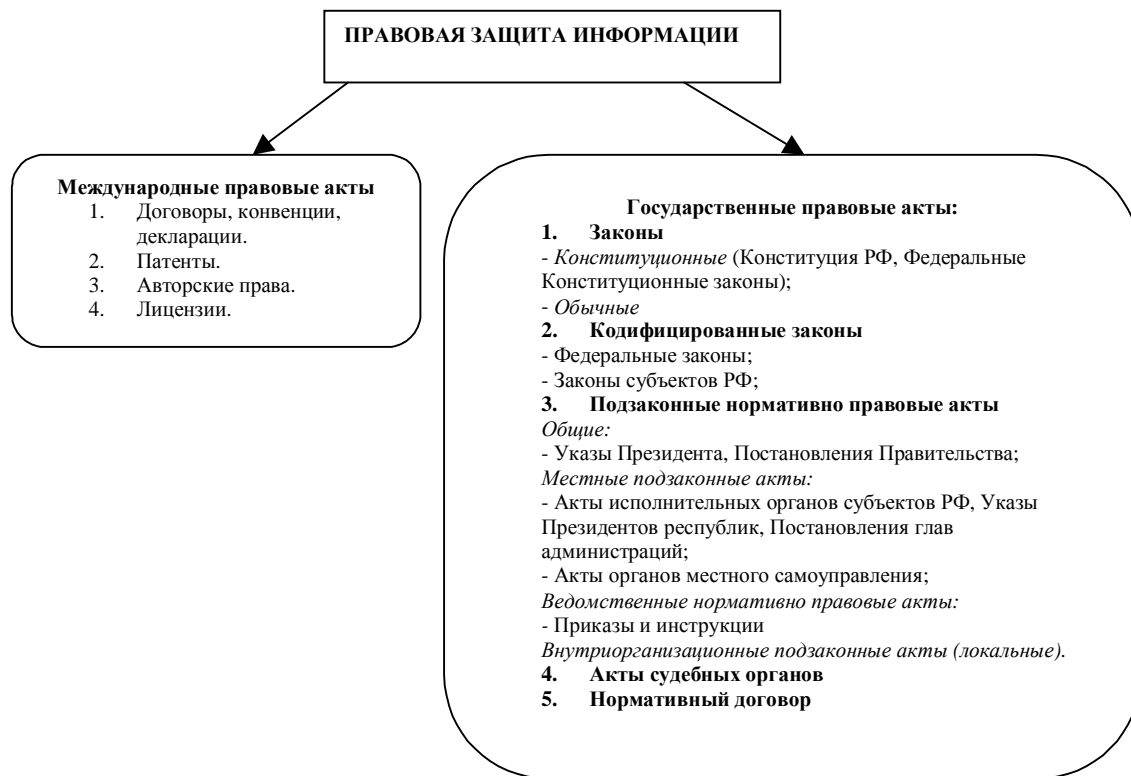


Рис.1. Правовая защита информации

Как отмечает В.И. Ярочкин [4], «... современные условия требуют и определяют необходимость комплексного подхода к формированию законодательства по защите информации, его состава и содержания, соотнесения его со всей системой законов и правовых актов Российской Федерации». С этим мнением нельзя не согласиться, тем более что на сегодняшний день оно постепенно воплощается в жизнь путем разработки и закрепления норм, регулирующих сферу информационных правоотношений, которые отражены в статьях целого ряда важнейших для России законов, указов Президента и постановлений Правительства:

1. Кодекс Российской Федерации об административных правонарушениях от 30 декабря 2001 г. № 195-ФЗ ст.ст. 5.13, 5.39, 8.5, 13.11, 13.13, 13.14, 20.23, 20.24, 20.44, 20.45;
2. Гражданский Кодекс Российской Федерации. Части 1, 2 и 3. Ст.ст. 51, 67, 128, 139, 150, 152, 727, 771, 1123, 1125, 1126;
3. Трудовой кодекс Российской Федерации от 30 декабря 2001 г. № 197-ФЗ ст.ст. 85-90, 351;
4. Уголовный кодекс РФ от 13 июня 1996 г. № 63-ФЗ ст.ст. 137, 138, 140, 144, 155, 183, 283, 284;
5. Закон РФ от 21 июля 1993 г. № 5485-1 «О государственной тайне»;
6. Федеральный закон от 29 июля 2004 г. № 98-ФЗ «О коммерческой тайне»;
7. Федеральный закон от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании»;
8. Федеральный закон от 4 июля 1996 г. № 85-ФЗ «Об участии в международном информационном обмене»;
9. Федеральный закон от 20 февраля 1995 г. № 24-ФЗ «Об информации, информатизации и защите информации»;
10. УКАЗ Президента РФ от 06.03.1997 г. № 188 «Об утверждении перечня сведений конфиденциального характера»;
11. УКАЗ Президента РФ от 30.11.1995 г. № 1203 (ред. от 29.05.2002) «Об утверждении перечня сведений, отнесенных к государственной тайне»;
12. УКАЗ Президента РФ от 31.12.1993 г. № 2334 (ред. от 01.09.2000) «О дополнительных гарантиях прав граждан на информацию»;
13. УКАЗ Президента РФ от 20.01.1994 г. № 170 (ред. от 09.07.1997) «Об основах государственной политики в сфере информатизации»;
14. УКАЗ Президента от 15.08.1995 г. № 808 «О президентских программах правовой информатизации»;
15. Постановление Правительства РФ от 28 января 2002 г. № 65 «О Федеральной целевой программе «Электронная Россия (2002-2010 годы)»;
16. Постановление Правительства РФ от 04.09.1995 г. № 870 «Об утверждении правил отнесения сведений, составляющих государственную тайну, к различным степеням секретности».

В некоторые из перечисленных нормативно-правовых актов были внесены изменения, что, несомненно, благотворно отразилось на правовом обеспечении информационной безопасности России. Но в тоже время, как отмечает Е.К. Волчинская [5], «...остались невыполненными большинство мер по совершенствованию нормативно-правового обеспечения информационной безопасности в области конституционных прав и свобод человека и гражданина, реализуемых в информационной сфере, в области информационного обеспечения государственной политики Российской Федерации, в области защиты информационных ресурсов от несанкционированного доступа, в области функционирования системы обеспечения информационной безопасности Российской Федерации».

При этом современный этап развития федерального законодательства характеризуется большим массивом действующих нормативно-правовых актов информационной сфере. Растет количество несистематизированных разобщенных правовых актов, многие из которых фактически утратили силу, дублируют или подменяют и нередко противоречат друг другу. Этот факт позволяет сделать вывод о том, что действующее законодательство еще не в полной мере соответствует общественным отношениям, складывающимся в сфере информации, информатизации и защиты информации.

В связи с этим требуется произвести систематизацию действующего законодательства путем инкорпорации как средства внешнего упорядочения нормативно-правовых актов в хронологическом или тематическом порядке, без переработки самих норм права, с созданием подраздела «Информационное право» в классификаторе правовых актов.

От государства требуется в числе приоритетных законопроектов рассматривать проекты, предложенные в соответствии с Концепцией развития информационного законодательства и Окинавской хартией глобального информационного общества, подписанной Президентом Российской Федерации 22 июля 2000 года:

- проект Федерального закона «О праве на информацию»;
- проект Федерального закона «Об обеспечении доступа к информации о деятельности государственных органов и органов местного самоуправления»;
- проект Федерального закона «О защите информации в информационно-телекоммуникационных системах»;
- проект Федерального закона «О персональных данных»;
- проект Федерального закона «О профессиональной тайне».

Следует признать, что концепция «информационного общества» и ФЦП «Электронная Россия (2002–2010 годы)», в частности, полностью не исполняются. Одна из причин этого – использование выделяемых ресурсов не по назначению. Существующая на государственном уровне межведомственная разобщенность и нескоординированность действий ведут к дублированию выполняемых работ. Информационно-правовое обеспечение базы персональных данных в системах МВД, МЧС, Пенсионного фонда, ГАС «Выборы», «регистр населения» и других создается различными ведомствами за государственный счет, но хранят дублирующую информацию. Указанная проблема существует не только на федеральном уровне. Дублирующие системы создаются и региональными, и местными органами власти. Но взаимная интеграция этих систем (с ограниченным доступом для каждого из субъектов) позволила бы существенно сэкономить государственный бюджет и направить его средства на решение неотложных задач.

Еще одна сложность состоит в том, что в законодательстве необходимо решить вопрос о системе взаимоотношений многочисленных субъектов в процессе реализации государственной политики в сфере формирования информационных ресурсов и информатизации, так как информационные ресурсы могут быть государственными и негосударственными, могут находиться в собственности органов государственной власти, местного самоуправления, коммерческих организаций, общественных организаций и граждан.

Также необходимо сделать вывод о том, что субъекты РФ остро нуждаются в разработке концепции совершенствования законодательства в области обеспечения информационной безопасности и основ федеральной политики в зоне правового обеспечения информационной безопасности в целях координации законотворческой деятельности субъектов РФ и совершенствования законодательства.

В заключение следует отметить, что для успешной трансформации нашего общества в информационное общество законодателям необходимо оперативно анализировать, учитывать, нормативно закреплять процессы, происходящие в информационной сфере, и желательно согласовать свою нормативно-правовую базу с международной

практикой, чтобы избежать осложнений при интеграции в мировое правовое пространство. Решение указанных задач обеспечит создание условий для ускорения процессов внедрения современных информационных технологий во все сферы общественной жизни, что позволит России сократить отставание от развитых стран мира, поможет избежать информационной изоляции от мирового сообщества и укрепит безопасность страны.

Литература

1. Окинавская хартия глобального информационного общества от 22.07.2000 г. № б/н // Дипломатический вестник. № 8, август 2000 года.
2. <http://www.infoforum.ru/detail.php?pagedetail=888>
3. «Доктрина информационной безопасности Российской Федерации» (утв. Президентом РФ 09.09.2000 N Пр-1895) / «Российская газета». № 187. 28.09.2000.
4. Ярочкин В.И. Информационная безопасность: Учебник для студентов вузов. М.: Академический проект; Гаудеамус, 2004. 544 с.
5. <http://www.infoforum.ru/detail.php?pagedetail=888>

ОБЗОР ПРОБЛЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ИНТЕЛЛЕКТУАЛЬНЫХ СЕТЕЙ

К.А. Коптяев

Научный руководитель – д.т.н., профессор Л.Г. Осовецкий

В статье рассматривается концепция интеллектуальной сети, ее основные узлы, приведен обзор основных проблем информационной безопасности и мероприятий по безопасности, которые должны быть проведены для минимизации угроз.

Введение

Бурный рост телекоммуникационного рынка в последние годы напрямую связан с развитием телефонных сетей и появлением в них множества новых функций помимо традиционных разговорных. Расширение функций стало возможно благодаря появлению концепции интеллектуальной сети (Intellectual Network – IN [1]). IN представляет собой совокупность узлов телефонной сети, позволяющих расширить традиционный набор услуг, например, такими услугами, как бесплатный вызов, телеголосование, виртуальная частная сеть и другими. Таким образом, IN является надстройкой над сетью, дополнением к телефонным станциям и коммутаторам.

IN позволяет операторам связи не только самим предоставлять услуги, но и привлекать сторонние компании для предоставления так называемого «контента» (content – содержание) пользователям сети. Примерами могут служить мелодии, картинки, тематическая информация.

Бурно растущий рынок дополнительных услуг и обострение конкурентной борьбы между поставщиками услуг и между операторами приводят к повышению требований по информационной безопасности систем и узлов IN, защите данных в сетях и обеспечению безотказного обслуживания абонентов и предоставления услуг.

Архитектура IN

IN состоит из узлов, выполняющих определенные функции в процессе предоставления услуги (см. рис. 1). Как правило, узлы IN представляют собой отдельные сетевые устройства. При небольшом количестве возможно объединение некоторых узлов.

В данном списке приведены основные узлы IN и описание их базовых функций:

- SSP (Service Switching Point – узел коммутации услуг). Используется для маршрутизации запросов на выполнение дополнительных услуг от традиционных узлов сети к узлам IN.
- SCP (Service Control Point – узел выполнения услуг). Задача SCP – выполнение логики услуги. Как правило, SCP получает какие-либо данные, выполняет определенные действия, описанные в логике выполнения услуги, и отправляет результаты выполнения услуги. В ходе выполнения логики возможно обращение SCP к внешним службам, например, биллингу.
- SDP (Service Data Point – узел данных). Данный узел предназначен для хранения данных, необходимых для выполнения логик услуг, при выполнении логики услуги SCP имеет доступ к данным SDP.
- SCEP (Service Creation Environment Point – узел создания услуг). Данный узел предоставляет инструментарий для создания и редактирования логик услуг.
- SMP (Service Management Point – узел управления услугами). Это единый центр управления интеллектуальной сетью, позволяющий конфигурировать узлы, добавлять и удалять услуги, маршруты, осуществлять мониторинг сети.

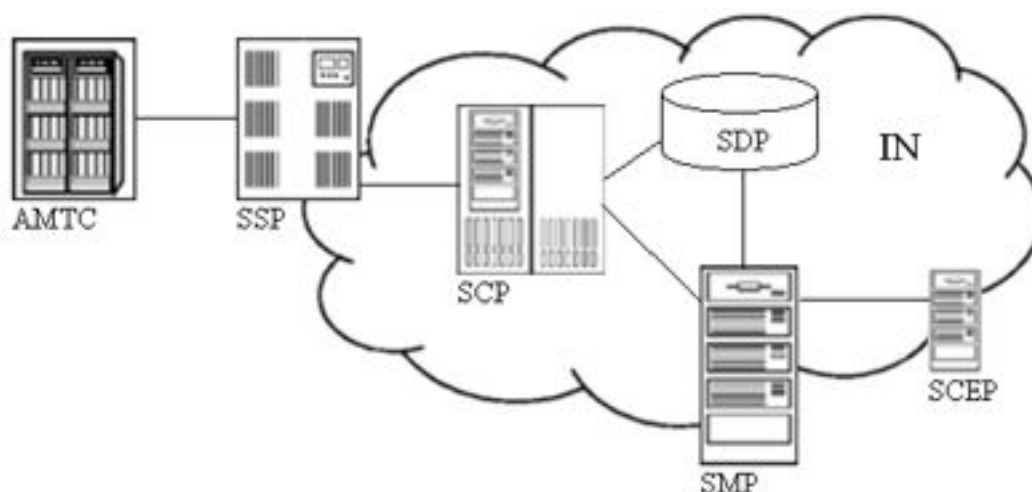


Рис. 1. Архитектура IN

Анализ уязвимостей узлов IN

IN является, как правило, распределенной системой, с множеством сетевых узлов, поэтому одной из основных проблем является возможность получения несанкционированного доступа к узлам сети и к конфиденциальной информации, управлению или логикам услуг.

Примером конфиденциальной информации могут служить, например, данные об абонентах сети, данные документов абонентов, состояние балансов лицевого счета абонентов. При получении доступа к управлению каким-либо узлом возможно выведение его из строя или изменение его конфигурации в пользу злоумышленника. Если получен доступ к логикам услуг, существует опасность уничтожения этих услуг, отказов абонентам со стороны оператора либо модификация услуг.

Учитывая многокомпонентность систем, существует опасность подмены оригинальных компонентов или логик услуг другими компонентами, использующими те же интерфейсы, но при этом применяющими алгоритмы работы, приводящие к нежелательным для абонентов или оператора последствиям.

Не менее важной проблемой является обеспечение безотказности работы всей сети IN в целом и отдельных узлов, в частности. При выходе какого-либо узла из строя некоторые (а иногда все) логики услуг не смогут корректно выполняться, и услуга будет предоставляться не в полном объеме либо не будет предоставляться вовсе. Поэтому выход из строя узла сети является серьезной проблемой. Причиной выхода узла из строя может стать резко возросшая нагрузка сети (например, в новогоднюю ночь) или сбой какого-либо компонента узла при определенном стечении обстоятельств.

Защита узлов сети

Уже в настоящее IN, как правило, строится на основе универсальной платформы, которая является неким общим для всех узлов фундаментом, однако нынешнее поколение платформ призвано решать в основном такие задачи, как настройка узлов, их конфигурирование и аудит активности узлов.

Для решения проблем информационной безопасности необходимо разработка и внедрение нового поколения платформ IN, отвечающих самым современным требованиям и стандартам по информационной безопасности. Перечислим некоторые требования к этим платформам.

Необходимо обязательное разграничение прав пользователей, которые могут иметь доступ к управлению узлами с четким описанием их ролей. Например, администратор SCP должен иметь доступ к конфигурированию SCP, а администратор услуг

должен иметь только возможность конфигурации и модификации логик услуг, т.е. иметь доступ к узлу SCEP. Таким образом, для выполнения серьезных изменений в динамике работы IN потребуются действия (подтверждения) нескольких пользователей. Например, изменять логику и принимать ее на исполнение должны разные пользователи.

Платформа сети должна отслеживать действия пользователей и оценивать возможные нарушения безопасности, например, блокируя при этом потенциального нарушителя.

Для исключения возможности получения несанкционированного доступа к данным об абонентах необходимо уделить особое внимание построению системы защиты узла хранения данных SDP. Необходимо обеспечить, чтобы на любую операцию на любом объекте данных распространялась какая-либо политика управления доступом.

Проблема безотказности работы сети должна решаться постоянным аудитом узлов, отслеживанием наличия связей между узлами и откликов от узлов в процессе работы. Необходимо построение системы мониторинга не только для каждого узла в отдельности, как зачастую реализуется в настоящее время, но и общесетевого средства мониторинга. Данное средство должно обеспечивать возможность не только противостоять угрозам, но и обеспечивать живучесть сети даже в том случае, если она подверглась какой-либо атаке. Необходимо построить платформу таким образом, чтобы она могла сама блокировать и перезапускать узлы при их потенциально опасных, некорректных или подозрительных действиях, переключая при этом нагрузку на резервные каналы и сервера.

Особое внимание необходимо уделить мониторингу внешних связей сети, для обеспечения высокого уровня безопасности IN необходимо гарантировать, что все внешние связи установлены именно с теми, с кем это нужно для функционирования узлов.

Заключение

Растущая востребованность дополнительных услуг приводит к усложнению архитектуры IN, увеличению количества узлов, повышению нагрузок и увеличению объемов обрабатываемой информации. При этом возрастают и угрозы информационной безопасности сети. Нынешнее поколение платформ IN зачастую не справляется с возросшим количеством угроз.

Некоторые аспекты проблем безопасности IN и требования к построению нового поколения платформы IN были рассмотрены в этой статье.

Литература

1. Гольдштейн Б.С., Ехриель И.М., Рерле Р.Д. Интеллектуальные сети. М.: Радио и связь, 2000.
2. Лихтциндер Б.Я., Кузякин М.А., Росляков А.В., Фомичев С.М. Интеллектуальные сети связи. М.: Эко-трендз, 2002.

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ ДЛЯ СОЗДАНИЯ ВИРТУАЛЬНЫХ ЭКСПОЗИЦИЙ, ПОСВЯЩЕННЫХ ЗНАМЕНАТЕЛЬНЫМ ДАТАМ УНИВЕРСИТЕТА

Т.В. Шеламова

Научный руководитель – д.т.н., профессор Ю.Л. Колесников

Последние два десятилетия XX века проходят под эгидой стремительного развития информационных технологий. Количество информации, связанной с деятельностью человечества, ежегодно увеличивается в геометрической прогрессии. Анализируя темпы развития информационных технологий, можно сделать вывод, что в XXI веке одним из главных ресурсов будет информация.

Развитие глобальных информационных сетей является ответом на потребности человечества в формировании единого информационного пространства и возможности оперативно находить необходимую информацию, передавать ее.

Сегодня в системе образования поднимается вопрос о широком использовании информационных технологий с целью повышения эффективности управления учебным процессом. Рациональное использование информационных ресурсов позволяет применять новые, более эффективные способы передачи студентам учебной информации, автоматизировать некоторые процедуры управления процессом профессиональной подготовки будущих специалистов.

В последние годы в мире наблюдается значительный рост пользователей сети Интернет, а также разнообразных образовательных и информационных ресурсов, технологии создания и содержание которых широко обсуждаются на многих представительных конференциях и в литературе [1,2]. В числе таких ресурсов отметим и разработки виртуальных музеев. Были созданы виртуальные музеи Саратовского государственного университета, Московского государственного университета и ряда других учебных заведений. С другой стороны, созданы и развиваются музеи отраслей науки и техники; например, Виртуальный компьютерный музей, Музей истории Интернета в России, Ангарский музей часов, Интернет-музей телевидения и радио и ряд других.

Задача сохранения истории своего высшего учебного заведения и отрасли своей деятельности является, с одной стороны, безусловно, актуальной, но, с другой стороны, не является самоцелью. Реализация поставленной задачи позволяет осуществлять, в том числе, и воспитательную функцию по отношению к новым поколениям студентов, которые в процессе обучения должны получить не только необходимую совокупность профессиональных знаний и умений, но и узнать историю своего вуза, историю своей отрасли, биографии ученых и педагогов, работавших в области их будущей профессиональной деятельности [3].

На протяжении нескольких лет в Санкт-Петербургском государственном университете информационных технологий, механики и оптики сотрудниками и выпускниками ИТМО проводились работы по сбору материалов к созданию музея. Разработка виртуального музея началась в 2000 году. Было открыто несколько разделов, основанных на коллекции подлинных исторических документов: статей, интервью, воспоминаний, документов и др. Исторические документы могут быть представлены в текстовых и графических форматах, содержат фотографии исторических событий и персоналий, копии исторических документов. При этом широко используются возможности гиперпространства, включающие не только внутренние перекрестные ссылки, но и широко используемые ссылки на внешние Интернет-ресурсы, содержащие интересующую нас информацию [4–8].

На новом этапе развития виртуального музея разрабатывается концепция создания виртуальных выставок, входящих в новый раздел музея – «Тематические выставки» (рис. 1). Так, к 100-летию первого выпуска специалистов Механико-оптическим и ча-

совым отделением Ремесленного училища цесаревича Николая в Университете была открыта выставка, посвященная этому событию (рис. 1).



Рис. 1. Вид страницы раздела «Тематические выставки»

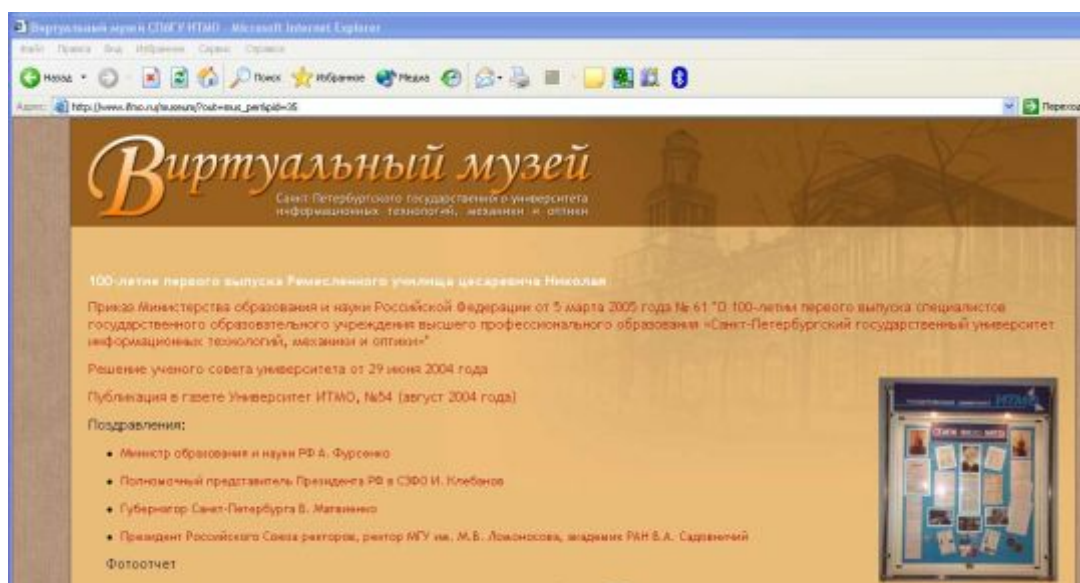


Рис. 2. Вид страницы, посвященной 100-летию первого выпуска специалистов механико-оптическим и часовым отделением Ремесленного училища цесаревича Николая

На втором этапе (2005–2007 годы) создания виртуального музея Университета запланированы:

- разработка лекций для студентов по курсам «История науки и техники», «Введение в специальность»;

- разработка специализированных разделов музея «История оптического приборостроения», «История оптического образования»;
 - создание виртуальных тематических экскурсий;
 - пополнение музея новыми материалами
- и ряд других мероприятий.

Таким образом, в результате проведенной в 2000–2004 годах работы создана первая очередь виртуального музея Университета и определены среднесрочные перспективы его развития (2005–2007 годы). Разработка виртуального музея Университета показывает, что его экспозиции позволяют вовлечь в учебный и научный процесс вуза новые материалы, которые до его создания были доступны лишь небольшому кругу сотрудников. Музей способствует более активному использованию информационной среды Университета, воспитанию студенчества, расширению научной проблематики и углублению содержания ряда учебных дисциплин. Следовательно, виртуальный музей Университета обладает уникальными возможностями для изучения истории оптического приборостроения и оптического образования.

Литература

1. Труды VI Всероссийской научной конференции «Научный сервис в сети Интернет». Новороссийск, 2004.
2. Интернет-порталы: содержание и технологии. Сб. научн. ст., вып. 2. М.: Просвещение, 2004. 499 с.
3. Васильев В.Н., Колесников Ю.Л., Чуфаров Е.В., Шеламова Т.В., Щербакова И.Ю. Виртуальный музей университета как средство изучения истории оптического приборостроения и оптического образования // Оптический журнал. 2005. Т.72. №3. С.69–73.
4. Колесников Ю.Л., Потеев М.И., Шеламова Т.В., Куркин А.В. Использование информационных технологий для построения виртуального музея университета // В книге: Труды X [Всеросс. конф.](#) «Тематика'2003», Т.1. СПб, 2003. С.212–213.
5. Колесников Ю.Л., Потеев М.И., Шеламова Т.В. Виртуальный музей истории создания и развития образовательного учреждения как составляющая его информационной среды // Научно-технический вестник СПб ГИТМО (ТУ). Выпуск 9. СПб: 2003. С.8–11.
6. Колесников Ю.Л., Шеламова Т.В., Щербакова И.Ю. Виртуальная образовательная среда и система изучения истории вуза как факторы формирования исторического сознания молодежи // В сб.: Наука и техника: Вопросы истории и теории. Выпуск XX. СПб.: СПбФ ИИЕТ РАН, 2004. С. 177–178.
7. Колесников Ю.Л., Шеламова Т.В., Щербакова И.Ю. Опыт создания виртуального музея университета и перспективы его использования в образовательной среде // В сб.: Актуальные проблемы вузовских музеев. СПб: Изд-во РГПУ им. А.И.Герцена, 2004. С.29–30.
8. Васильев В.Н., Колесников Ю.Л., Щербакова И.Ю., Куркин А.В., Шеламова Т.В. Виртуальный музей университета и его администрирование с помощью специализированного управляющего терминала // В сб.: Труды VI Всероссийской научной конференции «Научный сервис в сети Интернет». Новороссийск, 2004. С. 35–36.

К ВОПРОСУ О СОВЕРШЕНСТВОВАНИИ ПРЕПОДАВАНИЯ ДИСЦИПЛИНЫ «КОНЦЕПЦИИ СОВРЕМЕННОГО ЕСТЕСТВОЗНАНИЯ»

С.М. Вергезова

Научный руководитель – к.т.н., профессор М.И. Потеев

Согласно государственным образовательным стандартам высшего профессионального образования, дисциплина «Концепции современного естествознания» включена в обязательный минимум содержания профессиональных образовательных программ более чем по 25 направлениям и специальностям. К их числу относятся прежде всего гуманитарные и педагогические области образования. Имеется примерная программа этой дисциплины, утвержденная Минобразованием. В настоящее время по этой дисциплине издано более 70 учебников и учебных пособий, на сайтах Интернет – более 10 учебников.

Для определения путей совершенствования учебной литературы по дисциплине «Концепции современного естествознания» был проанализирован целый ряд учебных изданий и несколько электронных учебников. Их библиографические данные представлены в списке литературы. Список насчитывает более 60 наименований. На основании этого анализа намечены пути совершенствования учебной литературы.

Анализ учебных изданий сопровождался ранжированием объектов исследования по некоторым критериям, которые считались в процессе этого исследования основными:

1. полнота раскрытия вопросов программы;
2. системность изложения;
3. научность изложения;
4. доступность изложения;
5. наличие внутридисциплинарных и междисциплинарных связей;
6. наличие приложений (контрольные вопросы и задания, рекомендуемая литература, поясняющие и дополняющие тексты, словари, темы докладов и рефератов и т.п.);
7. наличие указателей (предметных, именных, адресов сайтов в Интернет и т.п.);
8. объем издания.

Эти параметры оценивались по пятибалльной шкале (от 1 до 5 баллов), а результаты суммировались в таблице.

В зависимости от полученных результатов все рассмотренные учебные издания и электронные учебники были разделены на пять групп: в первую группу вошли учебники, набравшие более 32 баллов; во вторую – набравшие от 17 до 31 баллов; в третью группу – набравшие менее 16 баллов; в четвертую группу вошли электронные учебники, размещенные на сайтах Интернет; пятую группу составили методические пособия и рекомендации.

При анализе также учитывалась, но не оценивалась специальность авторов издания (физика, математика, химия, экология, техника, философия).

Книги *первой группы*, например [8, 12, 13, 15, 17, 24, 25, 40, 42], представляют собой своего рода энциклопедии естествознания. Они содержат большое количество фактического материала, максимально полно отражающего требования «Примерной программы». Эти книги знакомят с естествознанием как неотъемлемым компонентом еди-

ной культуры, спецификой рационального научного мышления, а также закладывают основы целостного взгляда на окружающий мир как единство природы, человека и общества.

Книги первой группы отличаются высоким уровнем изложения материала. Все факты изложены хорошим литературным языком, доступно и в то же время научно, с учетом последних достижений науки, техники, культуры. Системный подход к изложению, наличие внутридисциплинарных и междисциплинарных связей способствует тому, что у студентов формируется целостное знание о мире природы, синтетические представления о законах и взаимных связях, существующих в окружающем мире.

Четкая структура этих учебников и наличие указателей делают их удобными для работы как студентов, так и преподавателей. Наличие разнообразных приложений мотивирует студентов. Включение в эти издания таких дополнений, как жизнеописания великих ученых, отрывки из их трудов, факты из истории научных открытий, стихи и цитаты делает чтение учебника увлекательным, развивает кругозор.

В изданиях *первой группы* зависимости качества учебников от специальностей авторов не наблюдалось.

Явными лидерами первой группы можно назвать [17] и [40]. Они набрали соответственно 35 и 39 баллов.

Учебник [17] – настоящая энциклопедия, сочетающая научность изложения с увлекательностью и огромным количеством материалов.

Учебник [40] прекрасно структурирован и имеет строгую логику изложения материала. В нем впервые в полной мере используется богатство мировой классической и современной поэзии для иллюстрации идей и концепций. В приложении представлен уникальный список ссылок на естественнонаучные сайты в Интернет.

Интересно и необычно учебное пособие [13]. Его отличительной особенностью является то, что в нем вся информация представлена в виде опорных схем и таблиц. В них содержится в концентрированном виде вся основная учебная информация. Схемы и таблицы делают пособие весьма доступным и наглядным для студентов, а для преподавателя – готовым набором листов опорных сигналов.

В пособии [48] рассматриваются синергетика и богатство человеческой культуры, выделяются общие принципы в разнообразном мире. Доступно, научно, экскурсы в историю, религию, культуру и искусство! Но при этом имеется недостаточно материала по физике, и совсем отсутствуют материалы по химии, биологии, астрономии, экологии. Напротив, в учебнике [25] огромное внимание уделено физике, химии, современным технологиям и влиянию их на Землю, но обойден вниманием человек как объект естественнонаучного познания – отсутствует антропология, психология, социология. Таким образом, несмотря на многие достоинства этих учебников, в них имеются некоторые противоречия программе дисциплины.

Книги из *второй группы* набрали заметно меньшее количество баллов, чем книги первой группы, прежде всего по следующим двум причинам:

- а) недостаточная комплектация приложений;
- б) уклон в одну из областей знания в ущерб другим (что явно связано со специальностями авторов).

Поэтому учебные издания *второй группы* в зависимости от специальности авторов были разделены на две подгруппы:

- написанные специалистами в области философии (например [6, 11, 16, 30, 41]);
- написанные специалистами в области физики, математики, химии, экологии, техники и т.д. ([3, 7, 19, 21, 32, 41, 53]).

Учебники, написанные специалистами в области философии, отражают главным образом историко-философское понимание естественных наук, социокультурное значение естественнонаучных открытий. Основное внимание уделяется человеку и его

месту в этом мире. Формируется целостный взгляд на мир. А сведения по физике, химии, биологии и т.д. изложены кратко, без лишних подробностей и не всегда удовлетворяют требованиям синтетического подхода к изучению естественных наук, предусмотренного курсом.

Напротив, учебники, написанные специалистами в области физики, математики и т.д., больше внимания уделяют сути различных естественнонаучных открытий и концепций. В них подробно и доступно для гуманитариев излагается большое количество фактов из физики, химии, астрономии, биологии. При этом в некоторых учебниках за фактами и формулами теряется сам человек, значение этих открытий для него.

Еще одной отличительной чертой учебников второй группы является их прикладной характер, т.е. направленность на конкретную специальность. Например, в [38], наряду с изложением естественнонаучных теорий и технических изобретений, приводятся примеры их применения на практике (в криминалистике). В [19] прекрасно описаны самоорганизация и эволюция на примере социально-экономических процессов. В [18] увлекательно изложены связи и видимые противоречия геологии с законами физики и химии.

Учебные издания из *третьей группы* (курсы и конспекты лекций), например, [4, 22, 44, 54], характеризуются следующим:

- очень малое количество фактического материала, несоответствие программе;
- материал изложен обзорно – тезисно, очень сжато;
- достаточно примитивное изложение;
- отсутствуют материалы по некоторым темам (например, география, астрономия, химия, биологическая природа человека, синергетика);
- материал не всегда удовлетворяет требованиям системного подхода;
- часто отсутствуют внутридисциплинарные и междисциплинарные связи;
- недостаточная комплектация приложений.

Для анализа и оценки электронных учебников и учебных пособий использовались те же критерии, что и для печатных изданий, и в дополнение было оценено удобство навигации по сайту.

Разработки [55, 59–61] наиболее полно удовлетворяют нашим критериям. Полнота охвата материала и системный взгляд на мир сочетаются с научностью и доступностью изложения. Удобство навигации по сайтам [55] и [61] делает их полноценными самостоятельными учебными пособиями.

Пособие [59] – не столько последовательное изложение учебного курса, сколько увлекательная и содержательная беседа по поводу наиболее интересных, значительных проблем современного естествознания, в частности, физики, биологии, антропологии, философские размышления о возникновении жизни и цивилизации, о взаимосвязи материи, пространства и времени, законов природы, принципов детерминизма, мира в его бесконечном многообразии. Пособие предназначено прежде всего студентам-философам, но, безусловно, будет полезно и для студентов всех других специальностей.

Очень интересно пособие на сайте [61]. В нем представлена современная естественнонаучная картина мира с точки зрения физики. Рассматриваются такие темы, как физика полей, физика Вселенной, физическая интерпретация биологических законов, физические принципы развития живых систем, физические аспекты биосферы и экологии и многие другие, необычные и интересные.

Информация на сайтах [56–58] представляет собой более кратко изложенный курс лекций, читаемых авторами в вузе. Данные тексты представляют собой лишь вспомогательный материал, облегчающий усвоение курса. В них чувствуется недостаток соответствия программе (не рассматриваются химия, биология, очень кратко изложена си-

нергетика), отсутствуют приложения (исключение составляет [56]). Навигация по этим сайтам не всегда удобна, а в [60] просто выложены лекции в формате Word.

Наконец, учебно-методические пособия *пятой группы*, например, [2, 9, 14, 20, 43, 47, 49, 54], представляют различные варианты построения курса, рекомендации преподавателям по изложению той или иной темы в зависимости от специальности и формы обучения.

Таким образом, книги *первой группы* можно рассматривать в качестве универсальных пособий для студентов всех специальностей. Книги *второй группы* могут использоваться в качестве пособий для студентов всех специальностей, с учетом их деления на философские и нефилософские. Книги *третьей группы* можно использовать для лекторов в качестве примерного конспекта лекций, и для дистанционного образования – в дополнение к другим учебникам.

Электронные учебники из *четвертой группы* могут разнообразить материалы курса, заинтересовать студентов и помочь при дистанционном изучении дисциплины.

Литература

1. Алексеев В.И. Концепции современного естествознания: Учебник в 2-х т. Т.1 – 153 с., Т.2 – 95 с. Владивосток, 2000.
2. Алексеев С.И. Концепции современного естествознания: руководство по изучению дисциплины. М., 2003. 25с.
3. Альтшулер Е.Ю., Маслов Р.В., Позднева С.П. Концепции современного естествознания: Учебное пособие для студентов гуманитарных специальностей. Саратов, 2002. 184 с.
4. Ахлибинский Б.В., Храленко Н.И. Основные концепции современного естествознания. М., 2000. – 226с.
5. Бабушкин А.Н. Современные концепции естествознания: Лекции по курсу. СПб, 2001. 208 с.
6. Буданов В.Г., Мелехова О.П. Концепции современного естествознания: Учебное пособие для спец. 0611. М., 1999. 116с.
7. Голиков П.А., Зайцев В.В., Орлов Ф.П. и др. Концепции современного естествознания: Учебное пособие для слушателей МВД России. Саратов, 1999. 100 с.
8. Горбачев В.В. Концепции современного естествознания: Учебник для вузов. М., 2003. 592 с.
9. Гордиенко В.А., Гордиенко В.Л. Концепции современного естествознания: Сборник учебно-методических материалов. М., 1998. 32с.
10. Горелов А.А. Концепции современного естествознания: Учебное пособие для студентов вузов. М., 2000. 512с.
11. Горохов В.Г. Концепции современного естествознания: Учебное пособие. М., 2003. 412с.
12. Грушевицкая Т.Г., Садохин А.П. Концепции современного естествознания: Учебное пособие для вузов. М., 2003. 670 с.
13. Грядовой Д.И. Концепции современного естествознания. Структурный курс основ естествознания: Учебн. пособие в схемах, определениях и таблицах М., 2001. 283 с.
14. Гутман Э.Е. Концепции современного естествознания: Методическое пособие. М., 1999. 47 с.
15. Данилова В.С., Кожевников Н.Н. Основные концепции современного естествознания: Учебное пособие для студентов вузов. М., 2000. 256 с.
16. Девятова С.В., Купцов В.И. Концепции естествознания – вопросы и ответы: Учебное пособие. М., 2002. 172 с.
17. Дубнищева Т.Я. Концепции современного естествознания: Учебник для студентов вузов. М., 2001. 827 с.
18. Дуничев В.М. Концепции современного естествознания: Учебное пособие. Южно-Сахалинск, 1999. 119 с.
19. Дыбов А.М., Иванов В.А. Концепции современного естествознания: Учебное пособие. Ижевск, 1999. 320с.
20. Еремина З.И. Концепции современного естествознания: Учебно-методическое пособие. Саранск, 2002. 32с.
21. Иванов-Шиц А.К. Концепции современного естествознания: Учебн. пособие в 3-х частях. Ч.1 – 32 с., Ч.2 – 58 с., Ч.3 – 69 с. М., 2000–2001.
22. Ивашковская Т.В., Павлов В.А. Концепции современного естествознания: Конспект лекций. СПб, 2000. 48 с.
23. Ильясова Т.В. Концепции современного естествознания: Учебное пособие для студентов педвузов. Оренбург, 2001. 420 с.

24. Карпенков С.Х. Концепции современного естествознания – практикум: Учебное пособие для вузов. М., 1998. 239 с.
25. Карпенков С.Х. Концепции современного естествознания: Учебник для высшей школы. М., 2000. 640 с.
26. Клименко И.С., Энгвер Н.Н. Концепции современного естествознания: Учебное пособие. М., 2001. 196 с.
27. Концепции современного естествознания – Хрестоматия: сост. Борщов А.С., Заров Д.И., Собинов В.М., Абросимова И.А. Саратов, 2002. 84 с.
28. Концепции современного естествознания: 100 экзаменационных ответов. /Под ред. Самыгина С.И. Ростов-на-Дону, 2002. 272с.
29. Концепции современного естествознания: Учебное пособие /Кропотин О.В., Прокудина Н.А., Суриков В.И. и др. Омск, 2002. 80 с.
30. Концепции современного естествознания: Учебное пособие для студентов гуманитарных и экономических специальностей./Под ред. Самыгина С.И. Ростов-на-Дону, 2003. 448 с.
31. Кочергин А.А., Кочергин А.Н., Егоров А.Г. Концепции естествознания: история и современность: Учебн. пособие в 3-х т. Т.1 – 447с., Т.2 – 408с., Т.3 – 576с. М., 1998–99.
32. Логвинов В.В. Концепции современного естествознания: учебное пособие. М., 2002. 192 с.
33. Миронов А.В. Концепции современного естествознания: Учебное пособие для юридических вузов. М., 2003.206 с.
34. Мотылева Л.С., Скоробогатов В.А., Судариков А.М., Концепции современного естествознания: Учебник для вузов. СПб, 2000. 320с.
35. Невважай И.Д., Пугачева Л.С., Соколенко В.М., Концепции современного естествознания: Курс лекций. Саратов, 2000. 156 с.
36. Никитенко В.А., Прунцев А.П. Концепции современного естествознания: Конспект лекций. М., 2002. 208 с.
37. Нуруллин Р.А. Концепции современного естествознания: текст лекций. Казань, 2002. 32с.
38. Основы естественнонаучных знаний для юристов./ под ред. Россинской Е.Р. М., 1999. 600с.
39. Пластинин В.В. Концепции современного естествознания: Курс лекций в 2-х ч. Ч.1 – 126 с., Ч.2 – 115с. Омск, 2002.
40. Потеев М.И. Концепции современного естествознания: Учебник для студентов вузов. СПб, 1999. 350с.
41. Рузавин Г.И. Концепции современного естествознания: Курс лекций. М., 2002. 336 с.
42. Рыбаков Г.М., Концепции современного естествознания: Учебное пособие в 2 т. М., 2000.
43. Рыбалов Л.Б., Россолимо Т.Е. Концепции современного естествознания: Учебно-методическое пособие и программа курса. М., 1997. 80 с.
44. Смык А.Ф. Концепции современного естествознания: Учебное пособие. Красноярск, 2000. 102с.
45. Соломатин В.А. История и концепции современного естествознания: Учебник для студентов вузов. М., 2002. 464 с.
46. Солопов Е.Ф. Концепции современного естествознания: Учебное пособие для студентов вузов. М., 2001. 232 с.
47. Тихонов А.И. Концепции современного естествознания: Методическое пособие. Иваново, 2002. 68 с.
48. Трухачева В.А. Концепции современного естествознания: Учебное пособие. Петрозаводск, 2000. 252 с.
49. Ушакова Е.В., Рассыпнов В.А. Концепции современного естествознания: Учебно-методический комплекс для гум. специальностей вузов. Барнаул, 1996. 16 с.
50. Хапачев Ю.П. Концепции современного естествознания: Курс лекций. Нальчик, 2000. 80 с.
51. Хорошавина С.Г. Концепции современного естествознания: Курс лекций. Учебник для студентов вузов. Ростов-на-Дону 2002.-480с.
52. Черепанов К.А., Черепанова В.К. Концепции современного естествознания: Конспект лекций. Новокузнецк, 2002. 92 с.
53. Шапиро С.В. Концепции современного естествознания: Учебное пособие. Уфа, 2002. 428 с.
54. Щенникова Л.С., Щенников А.А. Концепции современного естествознания: Учебно-практическое пособие (для дистанционного обучения). М., 2000. 84с.

Источники в Интернет

55. Дубнищева Т.Я., Мицель А.А. – www.campus.fortunecity.com/bates/313
56. Иванов-Шиц А.К. – www.limm.mgimo.ru/science/
57. www.juristy.ru/konspekt/natsciense.htm
58. Сипаров С.В. – www.so-znanie.nm.ru/oglav1.htm, www.philosophy.ru/edu/ref
59. Аруцев А.А., Ермолаев Б.В., Кутателадзе И.О., Слуцкий М.С. – www.nrc.edu.ru/est/pos/ , www.philosophy.ru/edu/ref
60. Кожевников Н.М. – www.nimiko.euro.ru
61. Горбачев В.В. – www.hi-edu.ru/x-books-free/book131

ОСНОВНЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ РЫНКА ТРУДА В РЕСПУБЛИКЕ БУРЯТИЯ

Ж.С. Базарова

Научный руководитель – д.э.н., профессор С.Б. Смирнов

В работе охарактеризованы основные тенденции развития рынка труда республики и определены наиболее эффективные направления социальной политики государства на рынке труда.

Введение

Современный этап развития экономики страны характеризуется переходом к рыночным отношениям, которые предполагают существование и функционирование множества рынков, охватывающих разнообразные области человеческой деятельности. В этой системе особое место принадлежит рынку труда, что связано с его социальной направленностью. Цель данного исследования – показать этапы становления рынка труда в Республике Бурятия, проблемы формирования и стабильного существования в современных условиях.

Важное значение, с точки зрения выработки стратегии регулирования рынка труда и занятости населения конкретного территориального образования, имеет информация о структуре, о составе безработных страны или конкретного региона; половозрастной принадлежности; имеющегося образования; возраста и т.д. Определение наиболее эффективных направлений социальной политики государства на рынке труда требует систематического проведения анализа ситуации и учета тенденций, характерных для современного рынка труда республики.

1. Этапы становления рынка труда в Республике Бурятия

Рынок труда в Бурятии, как и в России в целом, начал формироваться в 1991 году с принятием Закона «О занятости населения в РСФСР». Была образована Федеральная служба занятости России, начало действовать ее региональное подразделение в Республике Бурятия. Таким образом, государство приняло на себя предусмотренные Законом функции по регулированию рынка труда.

Вопросами занятости населения занималась государственная служба трудоустройства, которая проводила тщательный учет трудовых ресурсов, людей, вступающих в период экономической активности и выбывающих из него, просчитывала количество действующих и вновь вводимых рабочих мест, и на основе этих показателей составлялись балансы трудовых ресурсов.

Главной целью государства было достижение сбалансированности трудовых ресурсов с потребностями экономики путем поддержания баланса всех видов ресурсов, находящихся в распоряжении государства. Отличительной чертой действовавшей системы управления административно-командного типа была жесткая регламентация всех социально-трудовых отношений, которые были крайне формализованы и складывались в условиях преимущественно государственной формы собственности.

В целом затратный характер экономики, порождая дефицит в ресурсах, приводил к дефициту рабочей силы. Сбалансированности трудовых ресурсов с потребностями экономики никогда не достигалось. Более того, при постоянной нехватке рабочей силы ее цена оставалась неизменно стабильно – низкой, что поддерживало состояние отчуждения работника от результата труда.

В ходе проводимых экономических реформ 80-х годов начали формироваться условия для создания рабочей силы. Государство признало право собственности за работником на его рабочую силу. Появились кооперативы, стал привлекаться иностранный капитал, у работника возникла возможность выбора и трудовой мобильности. По

мере нарастания конкурентной борьбы на рынке товаров и услуг и вытеснения из него неконкурентоспособной продукции отечественных предприятий началось массовое высвобождение занятых на них рабочих и инженерно – технических работников. Появился избыток рабочей силы, ее предложение впервые превысило спрос.

В связи с резким ухудшением положения в экономике республики по сравнению с Иркутской областью напряженность на рынке труда возрастала несопоставимо высокими темпами. Так, если в 1994 году на 28830 ищущих работу в Иркутской области было заявлено 4437 вакансий¹, то в Республике Бурятия на 15452 человека – только 402 вакансии. В 1995–1996 годах примерно такое соотношение сохранилось. Уже в 1992 году в недавно созданные районные и городские центры занятости населения обратилось по вопросу трудоустройства 13865 человек, а потребности в рабочей силе заявили только 205 предприятий на 1444 человека.²

Всего в течение года было трудоустроено 4464 человека. В последующие годы разрыв между заявленной работодателем потребностью в рабочей силе и фактическим ее предложением увеличивался. В 1996 году обратилось 19678 человек, а рабочие места смогли получить только 6226. Резко возросло предложение рабочей силы со стороны 16–29-летних работников. Если с 1993 по 1996 годы общее предложение рабочей силы увеличилось на 34%, то со стороны молодежи – на 57%. В то же время вероятность трудоустройства молодежи сократилась с 30% от числа обратившихся в 1993 году до 24% в 1996 году. Одновременно с этим возникла проблема социальной помощи высвобождаемым работникам, ставшим безработными.

Поскольку ситуация в экономике республики продолжала ухудшаться, все больше и больше предприятий сокращало объемы производства или останавливалось совсем, увеличивалась продолжительность периодов безработицы.

Однако, вопреки общепринятым в рыночной экономике стандартам поведения, работодатели в Республике Бурятия действовали своеобразно. Следует отметить, что помимо открытой безработицы в Бурятии имеет место и так называемая скрытая (латентная) безработица. Однако вернее было бы говорить не о скрытой безработице, а о излишней численности занятых. Несмотря на падение объемов производства, в Бурятии не произошло, как ожидалось, обвального высвобождения работников. Под давлением различных причин, часто не имеющих экономического характера, на предприятиях сохранялась излишняя численность, которая зачастую основывалась на готовности подавляющего большинства получать минимальную зарплату в обмен на сохранение занятости. Тысячи людей в надежде сохранить за собой рабочее место при возможном положительном изменении ситуации в экономике добровольно ушли в неоплачиваемые отпуска или стали работать неполную рабочую неделю.

Анализ демографической ситуации в Бурятии показал, что в последние годы наблюдаются неблагоприятные изменения процессов воспроизводства трудовых ресурсов вследствие падения рождаемости, роста смертности, оттока населения из республики.

Показатель	1987 г.	1990 г.	1993 г.	1999 г.	2002 г.
Число родившихся, чел	23,3	18,2	11,4	11,0	11,3
Число умерших, чел	8,0	9,1	11,7	12,6	12,7
Естественный прирост/убыль	15,3	9,1	-0,3	-1,6	-1,4

Таблица 1. Естественное движение населения Республики Бурятия (в расчете на 1000 человек)

Важнейшим показателем, характеризующим трудовой потенциал населения республики, является численность трудовых ресурсов, которая включает трудоспособное

¹ Занятость и безработица в Иркутской области: департамент федеральной государственной службы занятости населения по Иркутской области. – Иркутск, 1998. – С. 8.

² Аналитические и отчетные материалы Министерства труда и занятости населения Республики Бурятия.

население в трудоспособном возрасте и лиц, работающих за пределами трудоспособного возраста (подростки и лица пенсионного возраста). Резкое снижение рождаемости, рост смертности, продолжающийся отток населения из республики отрицательно скажется в ближайшем будущем на формирование трудовых ресурсов.

По расчетам Госкомстата Республики Бурятия, предположительная численность населения в трудоспособном возрасте к 2008 году увеличится, а в дальнейшем резко снизится. Увеличение возникнет в результате вступления в эту группу молодежи, родившейся в 80-е годы, когда был отмечен рост числа родившихся. Одновременно из этой группы будут выходить малочисленные группы лиц, родившихся в 1941–1945 годах. В дальнейшем в трудоспособный возраст начнут входить лица, родившиеся в первой половине 1990-х годов, когда рождаемость начала резко снижаться, а выходить – лица, родившиеся в годы послевоенного подъема рождаемости. Вследствие этого численность лиц трудоспособного возраста резко упадет.

Значительную роль в формировании количественного и качественного состава трудовых ресурсов играет миграция. Миграция населения является важным источником изменения численности и структуры населения, а значит рынка труда.

В результате окончания строительства БАМа в середине 80-х годов, распада СССР, непринятия конкретных мер по закреплению прибывшего населения и кризисных процессов в экономике последнего периода наметился отток населения из республики. Так, за пределы ее в период 1986–1994 годов выехала 11,5 тыс. человек, причем более 70% из числа выбывших – население в трудоспособном возрасте. Отток населения из республики продолжается и в 1996 году составил 1,3 тыс. человек. В основном это жители города Северобайкальск, Муйского и Кяхтинского районов.

Однако следует заметить, что интенсивность миграционных процессов в республике по сравнению с другими регионами России выражено достаточно слабо, поскольку на нее воздействует целый комплекс социально-экономических, природно-климатических, этнических и других факторов, которые не располагают к привлечению дополнительной рабочей силы в регионе. Наиболее интенсивный обмен населением наблюдается между городом и селом внутри республики. Это связано со стремлением молодежи переселиться в город.

2. Современное состояние рынка труда в Республике Бурятия

Существенной задачей данного исследования является определение возможных масштабов рынка рабочей силы в республике.

Численность экономически активного населения республики составила 501 тысячи человек в 2002 году (в 1995 г. – 508 тыс. человек) (см. табл. 2). Из таблицы видно, что с каждым годом уменьшается численность экономически активного населения в общем объеме трудоспособного населения. Наибольший пик безработных граждан приходится на 1998–1999 годы, главной причиной является «августовский» кризис 1998 года. Начиная с 2000 года, ситуация на рынке труда начинает стабилизироваться. Так, в 2002 году численность незанятого населения составляет 83,8 тыс. человек (в 1999 – 119,5). Это связано с некоторым «оздоровлением» экономики республики.

Численность занятых в экономике в 1999 году снизилась по сравнению с 1995 годом, а численность незанятых лиц выросла. Начиная с 2000 года, наметилась тенденция снижения темпов роста незанятых. Общее сокращение численности занятых, прежде всего, связано со структурной диспропорцией спроса на рабочую силу и ее предложения в экономике республики. Наблюдается снижение численности работающих в отраслях материального производства на фоне увеличения занятости в большинстве отраслей непродуцированной сферы. Процесс сокращения численности затронул все отрасли народного хозяйства республики, кроме кредитования и страхования.

В настоящее время имеется достаточно большое число лиц, формально не имеющих работы и не зарегистрированных органами служб занятости. Это явление вполне можно отнести к нерегистрируемой открытой безработице. Основными причинами ее возникновения является: низкий уровень пособия по безработице и невысокие шансы получить работу через систему службы занятости. Однако из-за ограниченных возможностей свободного трудоустройства численность обратившихся граждан в службы занятости в последнее время резко увеличилось.

Показатель	1995 г.	1998 г.	1999 г.	2000 г.	2002 г.
Численность постоянного населения, тыс.чел.	1050,6	1048,0	1045,3	1041,1	1039,2
В том числе:					
Общая числ-ть трудоспособного населения, тыс.чел	598,2	602,5	604,8	609,6	613,9
В том числе:					
Экон-ки активное население, тыс.чел.	508,2	499,8	494,9	496,2	501,0
Из них:					
Занятые в экономике	420,4	386,9	375,4	397,9	417,2
Безработные, всего	87,5	112,9	119,5	98,3	83,8
Экономически неактивное население, тыс.чел	111,0	105,2	109,9	144,4	165,9
Доля экономически активного населения в общей численности трудовых ресурсов в %	84,9	82,9	81,8	81,3	81,6

Таблица 2. Экономически активное население в общем объеме трудовых ресурсов

	1996 г.	1999 г.	2000 г.	2001 г.	2002 г.
Численность граждан, ищущих работу, всего	19618	15927	17713	25553	22454
Из числа обратившихся имеют статус безработного	10252	8385	7760	10882	8312
Из числа безработных получают пособие по безработице	7968	6747	5644	8978	7761
Численность трудоустроенных	6226	7865	7894	11067	11372

Таблица 3. Численность граждан, зарегистрированных в службе занятости населения

В 2002 году в службы занятости обратилось в 1,1 раза больше незанятых граждан, чем в 1996 году. Из числа обратившихся официальный статус безработного получили в 1997 году 52,2% , а в 2002 году – 37,0%. Из них получили пособия 77,7% и 93,3% соответственно. Это свидетельствует о том, что в течение исследуемого периода значительно расширилась политика государства на рынке труда в части пассивных мер по социальной защите безработных. Численность граждан, трудоустроенных органами службы занятости, значительно колебалась. В течение 1997 года было трудоустроено 6226 человека. В последующие годы разрыв между заявленной работодателем потребностью в рабочей силе и фактическим ее предложением увеличился. В 2002 году, например, в службы занятости обратились 21707 человек, а рабочие места смогли получить только 11372.

Кроме того, следует отметить, что спрос на рабочую силу и предложение зачастую не совпадают. Работодателям часто требуется рабочая сила совсем иного профессионального и квалификационного состава, чем предлагается. Так, наиболее массовые профессии, требующиеся в республике – это штукатуры-маляры, каменщики, столяры, электрогазосварщики. На 1.10.2003 г. в банке вакансий Улан-Удэнского городского Центра занятости населения находилось 3984 вакансии, представленные 436 предприятиями на 383 профессии. Всего вакансий за январь – сентябрь текущего года насчитывалось 11100. Основную часть из них составляют мужские и рабочие профессии – 72%. Значительно меньше число вакансий, требующих наличия среднего специального и высшего образования—186 мест, что составило 23,7 % всех вакансий. В большинстве своем это рабочие места, рассчитанные в основном на мужчин. Для женщин подавляющую часть вакансий составляют медицинские сестры – 64 рабочих места.

Следует отметить, что большинство вакантных рабочих мест для женщин предлагают бюджетные учреждения, в которых низок уровень заработной платы и большие задержки (до 6 месяцев) по ее выплате. Все это, естественно, не может привлекать безработных. Текучесть кадров среди среднего и младшего медицинского персонала в г. Улан-Удэ стала уже хроническим явлением.

Вместе с тем, учитывая тот факт, что женщины составляют в настоящее время более 70% всех безработных, причем большая часть из них имеет среднее специальное и высшее образование, становится очевидным, что шансы получить работу, тем более по специальности, для большинства из них является весьма призрачными. Возможность трудоустройства различных категорий населения на рынке труда неодинакова.

Картина трудоустройства выпускников показывает ухудшение ситуации с устройством на работу как в количественном, так и в качественном выражении. Выпускники высших и средних специальных учебных заведений устраиваются, в основном, на неквалифицированные работы и не по своей специальности. Причин тому много: отсутствие вакансий по их профессиям, неконкурентоспособность на рынке труда (нет опыта работы, низкая квалификация), задержки в выплате заработной платы на предприятиях, ее мизерная величина и многое другое. Самой главной причиной является несбалансированность спроса и предложения рабочей силы на рынке труда.

По данным Городского центра занятости населения, спрос на квалифицированных специалистов не так высок, как хотелось бы. Главным препятствием для выпускников вузов при устройстве на работу является отсутствие стажа работы по специальности.

Продолжающаяся реформа перехода экономики на рыночные отношения существенно изменила условия деятельности промышленных предприятий Республики Бурятия. Впервые, начиная с 1999 года, состояние экономики Республики Бурятия характеризуется некоторым ростом промышленного производства. Так, объем промышленной продукции предприятий города за 1999 год в фактических ценах превысил соответствующий уровень прошлого года на 51%. Это происходит вследствие того, что промышленные предприятия республики, такие как ОАО «Приборостроительное производственное объединение», ОАО «Электромашина», ОАО «Улан-Удэстальмост», Улан-удэнский авиационный завод, Улан-удэнский стекольный завод, тонкосуконная мануфактура, начинают получать государственные заказы на свою продукцию. По данным Госкомстата, динамика индекса физического объема производства товаров промышленности города положительна: так, в январе–сентябре 2003 года по сравнению с соответствующим периодом прошлого года данный показатель составил 127%. В социальной сфере реальные денежные доходы увеличились на 25%. Параллельно с этим начинает развиваться малый бизнес. Правительство республики наметило принципиально новый курс на развитие частного предпринимательства. Значительно увеличилось количество занятых в торговле и общественном питании, жилищно-коммунальном хозяйстве при снижении количества занятых в образовании, культуре и искусстве.

Однако это совсем не говорит о том, что отсутствуют проблемы с трудоустройством. Основные из них практически не меняются: дефицит квалифицированных кадров, серьезные деформации профессионально-квалифицированной структуры и низкое качество рабочей силы – все это ведет к дисбалансу спроса и предложения рабочей силы. С другой стороны, работодатели не удовлетворяют ищущих работу низкой заработной платой и несвоевременностью ее выплат, необоснованными требованиями (по полу, возрасту, владение несколькими не смежными между собой специальностями и др.)

Таким образом, в работе описаны тенденции развития рынка труда в Республике Бурятия. Исходя из вышесказанного, можно сделать некоторые выводы, обобщая анализ каждого из рассмотренных вопросов.

Республиканские особенности занятости и формирования рынка труда в Бурятии проистекают из специфики ее демографических особенностей, сложившегося уровня и структуры занятости, обусловленной отраслевой специализацией и некомплексностью хозяйства, его конверсионной утяжеленностью, приоритетным развитием сырьевых и добывающих отраслей, негативной динамикой и низкой эффективностью производства, невысокой инвестиционной активностью.

Экономика Республики Бурятия в доперестроечный период характеризовалась высоким уровнем занятости при низком уровне жизни и эффективности труда, слабой территориальной отраслевой мобильностью рабочей силы, неразвитостью инфраструктуры рынка труда. Грянувший кризис 90-х годов негативно сказался на рынке труда, как впрочем и на всей экономике республики. По сути дела действия государства в начале 90-х годов представляли собой шаги по созданию инфраструктуры рынка труда, его законодательному и нормативному обеспечению, формированию государственной политики в области использования рабочей силы и занятости населения

Начиная с 2000 года, ситуация на рынке труда начинает стабилизироваться. Это связано с некоторым «оздоровлением» экономики республики. Состояние экономики Республики Бурятия характеризуется некоторым ростом промышленного производства. В сегодняшний день главной проблемой занятости остается не безработица, а неэффективное использование трудоустроенной рабочей силы, в первую очередь находящейся в вынужденном простое.

Бурятия относится к депрессивным регионам с высокой безработицей. Поэтому для эффективного управления рынком труда в регионе следует решить задачи макроэкономического порядка. Финансово-экономическая помощь со стороны Федерального правительства должна носить адресный характер, способствующей созданию различных механизмов воспроизводства рабочих мест.

Литература

1. Аналитические и отчетные материалы Министерства труда и занятости населения Республики Бурятия.
2. Батуева Р.Н. Улан-Удэ: рынок труда сегодня. // Бурятия. 22 ноября 1998. С.4.
3. Боровик В.С., Ермакова Е.Е. Занятость населения: Учебное пособие. Ростов-на-Дону: Феникс, 2001. 320 с.
4. Годовые отчеты о деятельности Центра занятости 1994–2003 гг.
5. Закон «О занятости населения в Российской Федерации».
6. Занятость и безработица в Иркутской области: департамент федеральной государственной службы занятости населения по Иркутской области. Иркутск, 1998. С.8.
7. Маслова И.С. Российский рынок труда. М.: Экономика, 1993. 125 с.
8. Никифорова А.А. Рынок труда, занятость и безработица. М.: Международные отношения, 1991. С.4.
9. Почепцов Г.Г. Теория коммуникаций. М.: «Рефл-Бук», 2001. С. 24.
10. Пыгин Н.М. Рынок труда, итоги и прогнозы. // Бурятия. 28 янв. 2000. С. 4.
11. Статистический сборник Госкомстата Республики Бурятия «Социально-демографическое развитие Республики Бурятия».
12. Труд и занятость в России. М.: Экономика, 1998. 56 с.
13. Экономическая теория. СПб.: СПб ГУЭФ, 1997. 327 с.
14. <http://gov.buraytia.ru>

ПРОБЛЕМЫ ИННОВАЦИОННО-ИНВЕСТИЦИОННОГО ПУТИ РАЗВИТИЯ РОССИЙСКОЙ ЭКОНОМИКИ И СПОСОБЫ ИХ РЕШЕНИЯ

С.В. Харжавин

Научный руководитель – д.э.н., доцент О.В. Васюхин

В национальной экономике одним из ключевых моментов является сокращение разрыва социально-экономического уровня развития страны по отношению к уровню развития ведущих стран мира.

Анализ проблем, связанных как с определением стратегических направлений перехода к рыночной экономике, так и с интегрированием России в мировое сообщество вынуждает форсировать переход к инновационно-инвестиционной модели развития нашей страны. Ведь производство конкурентной продукции на мировом рынке в современных условиях возможно лишь на инновационной основе, а материальную базу современной экономики создает инвестиционный капитал.

Для эффективного формирования инновационно-инвестиционного рынка необходима реализация широкого спектра мероприятий: обеспечение финансовыми ресурсами приоритетных направлений научных разработок, создание механизмов государственного стимулирования эффективного использования частных и иностранных инвестиций в наукоемкие области, в приоритетные научно-технические разработки, а также расширение форм инфраструктуры инновационного рынка (технопарки, технополюсы, научные центры, торговые инновационные биржи), приближение кредитов и банковского капитала к реальному производителю, государственная поддержка и стимулирование инновационно-инвестиционных проектов.

В настоящее время Правительством РФ планируется в течение ближайших десяти лет достичь двукратного роста валового внутреннего продукта страны. Важной проблемой на пути осуществления данной задачи является переориентация «сырьевой» российской экономики на экономику инновационного типа. Поэтому одним из приоритетных направлений развития страны является достижение ускоренного роста инвестиций в промышленности. Без внедрения инвестиций невозможно проведение технического обновления предприятий на основе введения инновационных технологий и повышения конкурентоспособности российских товаров, в том числе и на мировом рынке. Если данная задача не будет выполнена, то неизбежно дальнейшее технологическое отставание отечественной промышленности от зарубежных производителей.

Вопрос об увеличении валового внутреннего продукта, поставленный перед Правительством РФ, был воспринят значительной частью общества как новаторский и смелый. Однако в действительности речь идет о достижении примерно тех же основных количественных показателей, на которые СССР вышел в конце 80-х годов 20 века. В итоге экономический результат, который Россия достигнет через десять лет при успешной реализации программы по удвоению валового внутреннего продукта, окажется недостаточным для выхода на современные рубежи социально-экономического и технологического уровня ведущих стран мира.

Вот уже более десяти лет в России почти не вкладывались средства в модернизацию производства. Оборудование устарело не только морально, но и физически, что ведет к техногенным катастрофам. По данным конъюнктурных исследований, средний возраст основной части машин и оборудования в целом по промышленности в 2003 году составил 20,7 лет. При этом пятая часть оборудования приобретена до 1970 года, а удельный вес «молодого» оборудования, приобретенного после 1999 года, составляет 11 процентов. В ситуации, когда преимущественно происходит замена лишь отдельных единиц оборудования, структура производства остается утяжеленной устаревшими технологиями, не позволяющими перейти к выпуску инновационной и конкурентоспо-

собной продукции. Инженерные кадры на предприятиях представлены людьми преклонного возраста, не владеющими современными технологиями.

По статистике, сейчас в России лишь 10% предприятий можно назвать «инновационно-активными». В США таких предприятий – до 70 процентов. За последние годы в России не было открыто ни одного крупного месторождения. По данным Российской Академии наук, в переполненном финансовыми ресурсами топливно-энергетическом комплексе только чуть более 3% компаний разрабатывают инновационные проекты. И это при том, что 10% всех ученых мира живут в нашей стране. Таким образом, большинство перспективных научных разработок не реализуются на практике.

В 2003 году 56% объема производимой товарной продукции принадлежало топливно-энергетическому комплексу и металлургии, в то время как на долю продукции машиностроения в объеме промышленной продукции пришлось всего 20%.

По мнению автора, в первую очередь необходимо преодолеть усиливающуюся экспортно-сырьевую направленность экономики страны. Следует отметить, что увеличение валового внутреннего продукта страны сопряжено не только с увеличением экспорта природных ресурсов, но и с развитием производственных отраслей. В то же время доля валовых инвестиций в валовом внутреннем продукте, во многом определяющая объем инвестиций, в последние годы в России находится на уровне 21–23%. Между тем именно высокая доля валовых инвестиций (свыше 30%) позволила в свое время совершить экономический рывок Японии, а затем Южной Кореи, Сингапуру, Тайваню, Гонконгу, Малайзии. В Китае данный показатель находится на уровне 40%, что обеспечивает стране самую высокую в мире динамику экономического роста (в среднем 10% в год).

Объем инвестиций в основной капитал отечественной экономики сейчас примерно в 3 раза ниже, чем в 1990 году. Крайне медленно увеличивается инновационная активность предприятий. Если ситуация не изменится, Россия не сможет обеспечить кардинальную модернизацию технологической базы и структуры своей экономики, а значит, и те темпы и качество роста, которые необходимы для сокращения отставания от развитых стран.

Выходом из ситуации могли бы послужить не только заимствования в виде кредитов и займов, но и прямые инвестиции в капитал предприятий, что гораздо выгоднее обходится предприятию. Но, как считает автор, такой путь не для всех возможен по причинам непрозрачности хозяйственной деятельности предприятия. Инвесторы не торопятся становиться собственниками, а владельцы предприятий не слишком приветствуют новых акционеров, видя в них угрозу для своего положения в будущем.

Разумеется, в ближайшее время невозможно реорганизовать корпоративное поведение и управление на отечественных предприятиях таким образом, чтобы вывести их на уровень ведущих зарубежных конкурентов и заинтересовать инвесторов в участии, поскольку невозможно это сделать принудительно, а действующее законодательство несовершенно по формам и процедурам, сопутствующим притоку инвестиций. Поэтому необходимо совершенствовать нормативно-правовую базу инвестиционной деятельности, и Правительство РФ в данном направлении работает:

- принято постановление Правительства РФ от 31.03.98 г. № 374 «О создании условий для привлечения инвестиций в инновационную сферу»;
- принято постановление Государственной Думы Федерального Собрания РФ от 21 июня 2001 года №1664-III ГД «О Федеральном законе «Об инновационной деятельности и о государственной инновационной политике»;
- утвержден указ Президента Российской Федерации от 30 марта 2002 года «Основы политики Российской Федерации в области развития науки и технологий на период до 2010 года и дальнейшую перспективу».

Принят также комплекс нормативно-правовых документов для реализации механизма, позволяющего в дальнейшем использовать пенсионные накопительные средства населения на инвестиционные цели.

Однако данные программы Правительства на практике в полной мере не реализуются. Главной причиной этого является отсутствие реальных денежных средств для финансирования инвестиционных программ в достаточном объеме.

Развивающиеся компании, обладающие значительными оборотными средствами, достаточными для развития инвестиционных программ, в нашей стране в основном ориентированы на отрасли топливно-энергетического комплекса (ОАО «Газпром», ОАО «Лукойл», РАО «ЕЭС России» и ряд других) либо на спекулятивную деятельность – торговлю. Предприятия данных отраслей в текущих условиях рыночной экономики имеют значительную добавленную стоимость и постоянный спрос на свою продукцию и, следовательно, процветают. Руководство предприятий таких отраслей стремится инвестировать свободные денежные средства в собственный бизнес и не спешит финансировать промышленный сектор в условиях рыночной экономики переходного периода.

Данная тенденция сохраняется, поскольку производственные предприятия в условиях экономических преобразований столкнулись с целым комплексом проблем: устаревшие основные фонды; продукция, не выдерживающая конкуренции; дефицит пополнения квалифицированных кадров; зачастую значительные собственные долги. Таким образом, производственные предприятия перестали являться конкурентоспособными. Предприятия с таким потенциалом не могут рассчитывать на благоприятную инвестиционную привлекательность со стороны, собственных средств для развития не достаточно, а государственное финансирование практически прекратилось.

По мнению автора, в настоящее время российской экономике необходим приток иностранного капитала при условии долевого участия с российскими собственниками. Участие иностранных специалистов в развитии промышленного сектора экономики России способствует внедрению дополнительных инвестиций, использованию современных технологий на производстве, введению квалифицированных иностранных специалистов и современных методик управления предприятием.

С целью привлечения иностранного капитала для участия в инвестиционных проектах, необходимо проводить инвестиционные выставки федерального и регионального уровней при непосредственной поддержке государства, а также законодательно стимулировать и поощрять участие иностранных инвесторов, а также закреплять их права и обязанности на российском рынке. Важным шагом в реализации этой задачи является вступление России во Всемирную торговую организацию.

Международный и отечественный опыт показывает: экономические реформы целесообразно проводить системно с участием всех заинтересованных сторон. В итоге, чтобы бизнес смог оперативно реагировать на изменения ситуации на рынке, Правительству необходимо вовремя реализовывать механизмы государственного регулирования, целесообразно проводить и оценивать текущую конкурентоспособность как отдельного федерального субъекта страны, так и предприятий на его территории, прогнозировать основные направления развития предпринимательства в целом.

Хотя многие отечественные предприятия испытывают недостаток оборотных средств, финансовые ресурсы в России есть. В подтверждение данного положения говорит следующий факт: только за 1 полугодие 2003 года российские предприятия направили за рубеж 9,7 миллиардов долларов. Треть из них поступила в бывшую в то время оффшорную зону на Кипр, около 13% – в Виргинские острова. Основная часть инвестиций из России за рубеж направлялась в виде целевых банковских вкладов.

В 2003 году почти в полтора раза выросла сумма банковских вкладов физических лиц, превысившая 1,5 триллиона рублей. Наиболее популярные средства – банковские

вклады и наличная валюта – в России сейчас крайне убыточны. Практически отсутствует не только возможность приумножения накоплений, но и даже компенсация инфляции, под сомнение ставится вопрос сбережения собственного капитала в банковском секторе. Растущие доходы населения некуда инвестировать, поскольку в стране не отработана четкая инвестиционная политика. Население не имеет необходимой информации о возможностях инвестирования собственного капитала. В связи с экономическими преобразованиями в стране и неопределенностью в будущем население стремится вкладывать собственные средства либо в вечные ценности (недвижимость, драгоценные металлы), либо держать в сбережении. Таким образом, существенная часть капитала страны не доходит до промышленного сектора экономики.

Следует отметить, что без инновационно-инвестиционного прорыва, по мнению автора, Россия не сможет реализовать задачи, поставленные перед Правительством по удвоению валового внутреннего продукта в ближайшие десять лет. Поэтому государству необходимо формировать и стимулировать инвестиционные программы для частных инвесторов.

Разработка инвестиционной концепции развития страны обусловлена имеющимися недостатками в системе управления инвестициями: невысокая эффективность государственных инвестиций, ограниченность информационной среды инвестиционной деятельности вследствие отсутствия долгосрочной перспективы. Все перечисленные недостатки снижают эффективность инвестиционной деятельности развития страны.

Органы государственной власти, по мнению автора, должны способствовать развитию следующих отраслей:

- с высокой добавленной стоимостью на одного работающего;
- играющих роль «связующего звена»;
- имеющих потенциал роста в перспективе;
- позволяющих повысить вертикальную интеграцию предприятий отрасли.

Сдерживающими факторами в развитии инвестиционной деятельности в условиях перехода к рыночной экономики явились снижение роли государства в воспроизводственном процессе, отсутствие единой программы инвестиционного развития и методов ее реализации, структурных преобразований в экономике, вследствие чего наблюдается преимущественно сырьевая ориентация инвестиций в действующие производства, переориентация инвестиций в жилищное строительство. Сохраняется сложность и дороговизна согласования процедур инвестиционных проектов, высокая степень бюрократизации процесса согласования, недостатки координации между государственными структурами. Отсутствуют четкие механизмы и регламентированные сроки рассмотрения инновационных проектов, имеет место правовая незащищенность проектов, которые усложняют инвестиционную деятельность и приводит к притеснениям со стороны контролирующих органов, что усложняет инвесторам реализацию инновационных проектов.

Существенным тормозом инвестиционного процесса выступает отсутствие доступной базы данных по объектам промышленной недвижимости, инженерным коммуникациям и пятнам застройки, недостаточная проработанность предлагаемых инвесторам проектов, отсутствие единой концепции ведения инвестиционных проектов.

Одной из основных проблем инвестиционного пути развития страны остается недостаточность и дороговизна кредитных ресурсов. С учетом ограниченности государственных бюджетных ресурсов для инвестиций сохраняется актуальность привлечения заемных средств в кредитных организациях. Финансовые институты страны готовы работать в сфере кредитования инвестиционной деятельности предприятий, однако, в связи с повышенным риском инвестиционных проектов, запрашивают сверхвысокие процентные ставки, что невыгодно инвестору-заемщику. Помимо этого, кредитные организации в целях снижения риска заручаются залогом – обеспечением сделки (недви-

жимость, оборудование и другое). Однако государственные, муниципальные унитарные предприятия не в состоянии предложить в обеспечение сделки собственные фонды, поскольку они находятся в ведении государства. Процесс согласования кредитования государственного предприятия не отработан и занимает значительную часть времени. В итоге государственным предприятиям сложно рассчитывать на финансирование кредитными учреждениями, что тормозит их развитие.

Реализация инновационно-инвестиционной модели развития России невозможна без укрепления банковской системы. Инвестирование всегда связано с рисками, инновационная деятельность наиболее рискована. Это одна из основных причин того, что банки неохотно кредитуют инновационные проекты.

По мнению автора, в инвестировании государственных предприятий обязательно необходимо участие органов власти, выступающих гарантом и куратором инвестиционного проекта. Необходимо создавать и развивать государственные программы стимулирования участия кредитных организаций в инвестиционных проектах (например, программа льготного налогообложения).

По мнению автора, для более полной реализации инвестиционного потенциала страны представляется важным осуществление следующих мер:

- реализация программ по осуществлению единой инвестиционной политики государства;
- повышение эффективности использования инвестиционных ресурсов;
- повышение инвестиционной направленности и стимулирование финансовых институтов страны;
- привлечение и стимулирование иностранных инвесторов;
- административная поддержка в области развития объектов инвестирования (государственные гарантии под инвестиционный проект, установление режима льготного налогообложения участников инвестиционного проекта).

В целом, инвестиционная политика базируется на сценарии инновационной стратегии развития страны. Ключом к выходу российской экономики из кризиса является эффективная инновационная политика государства, направленная на развитие промышленного сектора экономики страны при непосредственной инвестиционной поддержке. Успешное решение проблемы создания новой технологической базы экономики как основного фактора ее конкурентоспособности на мировых рынках возможно лишь при пересмотре организационной структуры, системы управления с созданием инновационных технологий в стране. Инновационная политика промышленных предприятий может быть реализована путем создания технопарков, технополисов с полным научно-производственным циклом и разно профильными производствами.

С точки зрения автора, для России как государства с переходной экономикой только единая инновационно-инвестиционная стратегия развития может определить путь социально-экономических преобразований. В связи с этим среди первоочередных задач современной промышленной политики развития инновационно-инвестиционных процессов в реальном секторе экономики следует назвать: проведение единой технологической политики с целью достижения высокого качества научно-исследовательских, проектных и технологических работ; стимулирование взаимодействия отраслевой науки и производства; правовое и организационное обеспечения формирования дилерских сетей на рынках инвестиционных ресурсов и систем послепродажного обслуживания, лизинговых компаний; формирование рыночных институтов для продвижения капиталов и накоплений между отраслями и сферами отечественной экономики; стимулирование иностранных инвесторов финансировать промышленное производство в России. При этом государство должно выступать в роли общественного посредника, в задачи которого входит создание благоприятных условий для достижения компромисса между промышленным сектором экономики и финансовыми институтами страны.

По мнению автора, только опережающие темпы роста валового внутреннего продукта по сравнению с темпами роста реальных инвестиций могут характеризовать эффективность инновационно-инвестиционной деятельности и свидетельствовать о повышении отдачи вложений в реальный сектор экономики.

Литература

1. Бард В.С. Инвестиционные проблемы российской экономики. М.: Экзамен, 2000.
2. Гусаков М. Устойчивое развитие инновационной экономики регионов // Экономика. Политика. Инвестиции. 2004. №1 (11). С. 57.
3. Кузнецов С. Реализация инвестиционного потенциала Санкт-Петербурга. // Экономика. Политика. Инвестиции. 2004. №1 (11). С. 58.
4. Москвин В. Условия для реализации инвестиционных проектов в России. // Инвестиции в России. 2002. №4. С. 3–9.
5. Стратегия развития Российской Федерации до 2010 года. // Коммерсант. Проект «Центр стратегических разработок». 2000. № 51.

ПРОБЛЕМА СОЗДАНИЯ ИНТЕГРИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ

Н.С. Емельянов

Научный руководитель – д.т.н., профессор О.В. Емельянов

Во всем мире организации стремятся к созданию интегрированных систем управления для повышения эффективности и результативности работы. В статье рассматриваются проблемы, которые сопряжены с процессом выбора подхода к созданию интегрированной системы управления предприятием.

Пойти по пути поиска интегрированной системы менеджмента может лишь та организация, перед которой стоит задача повышения эффективности и результативности деятельности. В большинстве случаев руководство предприятия обращает свое внимание на тот или иной нормативный документ, созданный в различных институтах стандартизации, но, к сожалению, одним стандартом не охватить все требования, которые предъявляются к современной организации. На данный момент немногим предприятиям удастся эффективно соответствовать всем требованиям. В основном, это связано с тем, что не существует универсального руководства по созданию интегрированной системы управления, а методика сводится к дифференцированному созданию рабочих групп по каждому направлению.

Основное стратегическое преимущество подхода к внедрению требований нескольких стандартов как к единому интегрированному процессу заключается в системном решении и предупреждении возникновения проблем, связанных с различными аспектами деятельности компании. На практике этот подход должен сформировать «единую культуру» внутри организации, когда качество, экология и профессиональное здоровье и безопасность рассматриваются как равнозначные составляющие управленческой деятельности.

Существенными преимуществами интегрированного подхода также являются:

- уменьшение объема документации системы менеджмента благодаря объединению общесистемных процедур;
- упрощение внедрения системы за счет проведения совместных внутренних аудитов, совместного обучения, единого анализа со стороны руководства;
- сокращение организационных и финансовых затрат на сертификацию системы.

Разнообразие систем менеджмента и используемых при их создании стандартов быстро росло и охватило новые сферы деятельности. Основными из них являются: создание качественной и безопасной для потребителя продукции, минимизация рисков в процессе производства, социальная ответственность перед сотрудниками предприятия, соблюдение экологических норм и др. Перефразируя определение, приведенное в стандарте ИСО 9000:2000, система менеджмента может быть описана как последовательный набор взаимосвязанных и взаимодействующих мер и практических методов, направленных на установление политики и вытекающих из нее целей. Все эти действия должны обеспечивать систематическое управление и улучшение организационных и других процессов организации [1].

Система менеджмента имеет как внутренние, так и внешние параметры. Когда организация выделяет средства на достижение поставленных целей, она гарантирует сторонним организациям последовательное удовлетворение требований, в том числе законодательного характера. Внутренние и внешние параметры тесно взаимосвязаны. Как для самой организации, так и для заинтересованных в ее деятельности сторон эффективная система менеджмента должна обеспечивать идентификацию, управление и минимизацию связанных с бизнесом рисков, создавая при этом максимальные возможности для осуществления своей деятельности. Внедрение системы, в частности, должно сопровождаться:

- уменьшением рисков, не позволяющих организации достигать целей, и расширением возможностей, способствующих решению поставленных задач;
- уменьшением риска невыполнения требований, потребностей и ожиданий заинтересованных сторон, а также повышением вероятности выполнения их требований;
- расширением возможностей для улучшения деятельности организации с целью повышения удовлетворенности потребителей и уменьшения негативных воздействий на окружающую среду, что в целом улучшает положение организации на рынке.

Первой и наиболее значимой в создании интегрированной системы менеджмента является проблема, связанная с выбором основы этой системы, которая должна задать правила получения, обработки и анализа информационных потоков для принятия управленческих решений.

Наиболее эффективными системами менеджмента можно назвать такие системы, которые способствуют улучшению деятельности в соотношении с повседневной работой. Как правило, такие системы менеджмента должны базироваться на двух принципах: цикле Деминга–Шухарта и процессном подходе.

Цикл Деминга–Шухарта описывает последовательность действий, определяемых как «планируй–делай–изучай–действуй» (PDCA), которые направлены на достижение целей более экономичным и эффективным путем [2] (рис. 1). Эффективная и результативная система менеджмента включает в себя методы и практику (которые часто документируются как процедуры) для обеспечения выполнения этих действий. Систематическая идентификация и управление процессами внутри организации и в особенности взаимосвязь и взаимодействие между процессами называется «процессным подходом» (рис. 2).



Рис. 1. Цикл Деминга–Шухарта

При создании системы менеджмента следует с самого начала исходить из процессов, действующих в организации. Их рассмотрение предполагает установление требований на входе и выходе каждого процесса, оценку рисков и возможностей, определение и внедрение механизмов управления, измерения и анализа результатов деятельности, а для завершения цикла – определение возможностей улучшения и их реализация. Система, основанная на сочетании принципов PDCA и процессного подхода, должна служить основой для дальнейшего выбора стандарта на систему.

Следующей проблемой является выбор стандарта, соответствию которому обеспечило бы соответствие всем требованиям, предъявляемым организации со стороны партнеров, клиентов, поставщиков, законодательства, работников и т.д. Общего стандарта на систему менеджмента, в основе которой лежали бы указанные выше принципы, не существует. Если бы он существовал, единая модель менеджмента организации

включала бы в себя все существующие области менеджмента, что автоматически вело бы к созданию интегрированной системы. Однако ИСО и другие организации разработали отдельные стандарты и руководящие указания для каждой конкретной области менеджмента (качество, охрана окружающей среды, профессиональное здоровье и безопасность, безопасность продуктов питания и т.д.).

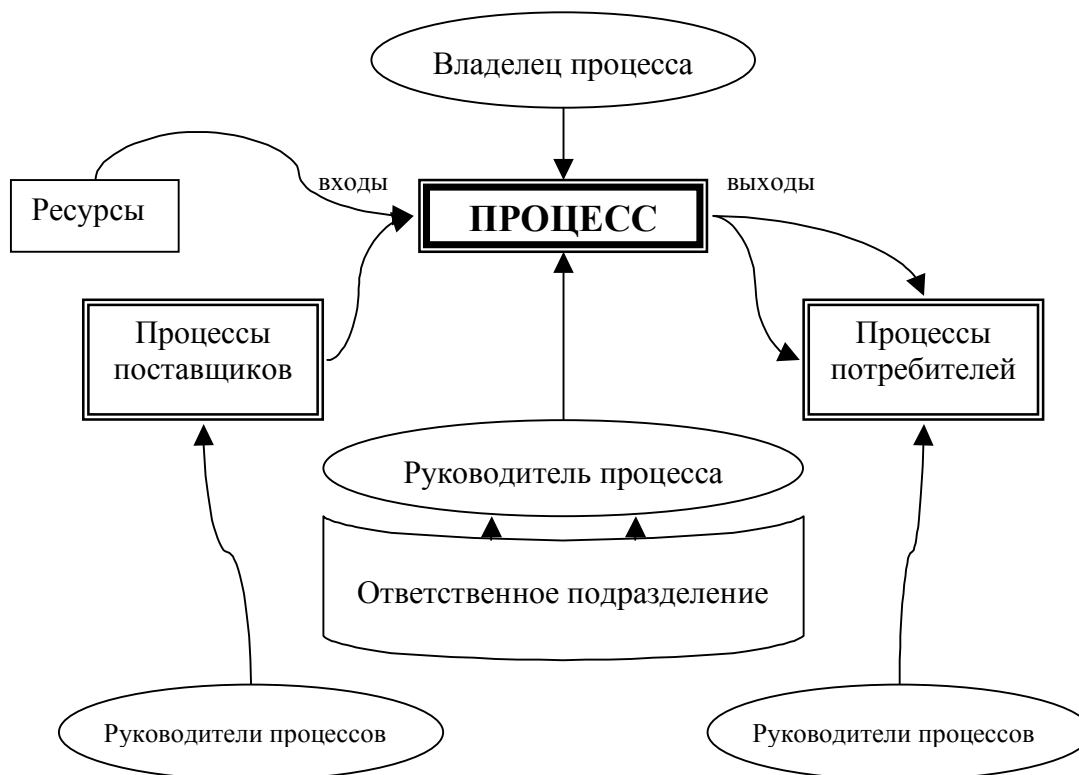


Рис. 2. Схема «процессного подхода»

Далее за выбором стандарта или перечня стандартов, который бы охватывал все запросы руководства по направлениям соответствия установленным нормам, следует выявление общей основы. Другими словами, все выбранные стандарты должны сходиться в одном – идентификации системы менеджмента, так как именно идентификация системы менеджмента позволит в дальнейшем говорить об общих инструментах следования политике и достижении целей в различных для каждого из стандартов областях. С целью реализации принятой политики организация должна осуществлять управление критическими аспектами, связанными с воздействием на качество, окружающую среду, профессиональное здоровье и т.д. И механизмы управления в интегрированной системе должны сформироваться в процессе синтеза основополагающих принципов каждого из интегрируемых стандартов.

Когда определена основа интегрированной системы менеджмента и интегрируемые стандарты, необходимо выработать методику внедрения системы. Это наиболее емкий этап процесса создания интегрированной системы менеджмента, основными задачами которого являются:

- сформулировать и обосновать методические положения создания и использования организационной и экономической модели выделения ресурсов на создание интегрированной системы управления;
- сформулировать и обосновать методические положения создания и использования организационной и экономической модели выполнения и контроля выполнения внедрения интегрированной системы управления;

- сформулировать и обосновать систему целей организации (от миссии до целей структурных подразделений), которая позволила бы проводить всесторонний анализ деятельности в обозримые сроки.

После выбора методики следует процесс практической реализации всех этапов внедрения системы менеджмента. Данный процесс требует создания эффективной матрицы ответственности и полномочий рабочей группы по созданию интегрированной системы управления и наличия полного алгоритма действия согласно методике, разработанной на предыдущей стадии создания системы.

Завершающим этапом в процессе создания интегрированной системы управления предприятием является ее сертификация. В том случае, если все предшествующие этапы были выполнены в необходимом объеме, процесс сертификации не представляет значительных трудностей, но в процессе ресертификации (например, системы менеджмента ИСО ресертифицируются один раз в четыре года), при обязательном ежегодном инспекционном контроле, может возникнуть необходимость внесения изменений в организацию работы системы менеджмента, это будет подтверждением всестороннего анализа со стороны руководства и очередным витком спирали цикла PDCA – основополагающим принципом системы управления.

В заключение необходимо отметить, что интегрированная система управления должна разрабатываться отдельно для каждой организации, так как «стандартной интегрированной модели, способной всех удовлетворить», не существует. В каждой организации интегрированная система управления должна базироваться на процессах этой организации и строиться в соответствии с проводимой ею политикой и поставленными целями. Именно эта мысль выражена в одной из колонок редактора журнала «Европейское качество. Дайджест»: «Каждая компания должна искать свой собственный путь достижения совершенства, используя весь накопленный человечеством опыт...» [3].

Описанные в данной статье основные проблемы, возникающие при создании системы, помогут при разработке системы менеджмента продуктивно совместить, с одной стороны, общие и широко используемые в практике многих организациях принципы и подходы, а с другой – условия, характерные для конкретной организации.

Литература

1. ГОСТ Р ИСО 9001-2001 (ИСО 9000:2000).
2. The New Economics for industry, government, education. W. Edwards Deming. 1996. Massachusetts Institute of Technology Center for Advanced Engineering Study Cambridge, MA 02139.
3. От качества к деловому совершенству // Европейское качество. Дайджест. 2004. №4. С.1.

ОСОБЕННОСТИ ФОРМИРОВАНИЯ РОССИЙСКОГО МЕНТАЛИТЕТА

О.И. Орлов (Санкт-Петербургский государственный университет кино и телевидения)
Научный руководитель – д.соц.н., профессор Л.Я. Орлова

Понимание особенностей формирования российского менталитета чрезвычайно важно для формирования отечественного гражданского общества и решения целого ряда важнейших проблем экономической и политической жизни в современной России. Прежде всего, очертим само смысловое поле менталитета. Как известно, сам термин имеет латинское происхождение, пришел в русский язык из французского и обозначает склад ума, мироощущение, миропонимание¹.

Представляется, что основу менталитета составляют несколько взаимосвязанных и взаимообусловленных «пластов»:

- национальная культура в широком смысле слова, которая вбирает в себя историческую память, традиции, обычаи;
- предрасположенность или непредрасположенность общества к заимствованию мировой культуры. Здесь мы выходим на проблему открытого и закрытого общества;
- мера усвоения гражданами различных видов духовной культуры, прежде всего той, которая необходима в процессе выборов – политической, правовой, социально–психологической;
- религия, ее влияние на общество;
- национальная психология;
- среда проживания (ландшафт, климат, народы, окружающие данный этнос).

Суммируя сказанное, можно отметить, что менталитет есть социо–культурно–психологическая характеристика этноса.

Понять особенности российского менталитета невозможно без обращения к его историческим корням.

В рамках одной статьи невозможно подробно проанализировать все исторические корни национального менталитета, поэтому мы остановимся на тех, которые лежат в основе проявления отечественного менталитета, влияющего на поведение электората в современных условиях.

Буферное положение России между Европой и Азией издавна обусловило пересечение двух социокультурных типов – европейского и азиатского, или лично–центристского, ставящего в центр внимания личность, ее свободу, естественные права, и социоцентристского, ориентированного на коллектив, общество, государство. При этом взаимодействие этих социокультурных типов в российском обществе весьма своеобразно: оно предполагает не только их переплетение и взаимообогащение, но и непрерывную борьбу между ними.

Фактически весь процесс формирования древнерусского государства «работал» на внутреннюю противоречивость русской души и политической культуры. Длительный период существования самоуправляющихся по сути республик на севере страны (Новгородская и Псковская вечевая демократия), свободное казачество, вольные землепашцы (до установления крепостного права), а равно и государство, формировало анархические наклонности, нигилистическое отношение к праву и государственной власти.

Следует отметить, что анархические наклонности в российском менталитете организационно оформились в начале XX века в движения и политические партии. Идеино подготовленное достаточно длительным периодом исторического развития, российское анархическое движение отразило почти полувековой опыт деятельности русских революционеров – народников. Несмотря на известные различия в политических доктринах разных течений российского анархизма, все они были единодушны в отношении к го-

сударству и его атрибутам насилия и угнетения. Государство признавалось ими источником социального неравенства и всех бедствий народа.

Государство, по мнению анархистов, создается со всеми необходимыми для его государственного управления лакеями: судьями и адвокатами, шпионами и жандармами, тюремщиками и палачами – всей этой «сворой отчаянных подлецов и развращенных до мозга костей дармоедов», которые в руках эксплуататоров являются орудием «для обирания производителей и откармливания эксплуататоров»².

По мнению одного из основных теоретиков русского и международного анархизма П.А. Кропоткина, необходимость социальной революции проистекала не только из критики капитализма и государственности, но также из внутренней потребности перемен в обществе. «Есть эпохи в жизни человечества, – писал он, – когда глубокие потрясения, громадный переворот, способный расшевелить общество до самой глубины его основ, становится жизненно необходимым»³. Автор приводит сравнение ситуации, сложившейся в Европе в 70-80-х гг. XIX века, с обстановкой накануне падения римской империи. Особенно остро он чувствовал моральную отчужденность, прекращение «нравственной связи между людьми». Нарушение такого баланса в буржуазном обществе Кропоткин объяснял эксплуатацией человека человеком. Привилегированные классы – «праздные люди», которые вместо «чувства солидарности» приобрели «преступный эгоизм». Но самое страшное в них – полное отсутствие патриотизма.⁴ Анархические партии и группы существуют и в современных условиях. И, несмотря на их малочисленность, при соответствующем развороте социально-политической ситуации они могут оказать влияние на политический процесс в России.

Будучи дочерней цивилизацией по отношению к Византии, Россия восприняла от нее не только православие, но и культуру в целом, прежде всего имперскую идею, реализация которой привела к превращению страны в многоэтническую империю. Понимание данного обстоятельства подводило к осознанию необходимости подчинения государству. При этом власть терпели, и когда она ослабевала, всякий раз предпринимались попытки ее свергнуть.

Утвердившийся авторитаризм в сочетании с многочисленными феодальными пережитками породил, в свою очередь, стихийный монархизм в индивидуальном сознании. Одновременно с этим формируется патернализм населения, ведущий в конечном итоге к массовой политической инерции, что в условиях отсутствия демократических принципов и форм связи между властью и гражданами укрепляет уже существующий правовой нигилизм.

Из Византии массовым сознанием был воспринят наднациональный характер как самой политической власти, так и государственности. Это проявилось в интернациональном подходе к формированию политической и интеллектуальной элит российского общества и, в конечном счете, обусловило и по сей день отсутствие русского государства и неразвитость политического сознания русского народа.

Прерывность исторического развития российского общества, многочисленные и разнообразные революции, крестьянские войны, восстания на протяжении длительного периода времени определили ориентацию его менталитета на отрицание предыдущих этапов, норм и ценностей в них доминирующих. И в настоящее время кардинально менялись местами прежние и новые ценности. На смену тотальному отрицанию капитализма пришло восторженное и бездумное подражание ему, заимствование даже тех ценностей, которые давно уже перестали быть таковыми в экономически развитых демократических странах. Безработица, эксплуатация, коррупция, культ денег и наживы воспринимаются определенной частью современного российского общества как нормальное явление. И напротив, гражданское равенство, социальная справедливость, патриотизм, ответственность государства перед народом, привычные ранее ценности вытесняются из общественного сознания. На этом фоне в сознании людей утверждается

скептицизм, безверие и разочарование в государственной власти и политической элите. Поскольку прежние нравственные ценности менталитета вытеснены, а новые еще не утвердились, возникает вакуум, который все равно требуется заполнить идеями, с одной стороны, базирующимися на исторической традиции, а с другой – адекватно отвечающими социально-экономическим условиям современности.

Исследуя общественное сознание, Ж.Т. Тощенко отмечает иррациональное совмещение взаимоисключающих оценок, установок и намерений в одной и той же личности. Наш дезориентированный соотечественник мечется, одновременно пытаясь бежать «от себя и общества ... в противоположных направлениях. В основе столь мучительного расслоения личности – утрата образцов, эталонов поведения».⁵ В качестве таких идей предполагаются идеи защиты государства, целостности страны, патриотизма. Являясь сами по себе, безусловно, правильными, они в то же время требуют для реализации их в жизни соответствующего отношения со стороны государства, которое пока не наблюдается.

Резкая смена политических и экономических ориентиров, дезорганизация системы ценностей порождают «ложное» сознание, отражающее объективную реальность искаженно и тенденциозно. Представленное, в основном, «в виде мифов и стереотипов, ложное сознание оперирует не столько реальной, сколько желаемой картиной мира». При этом немалые усилия, чтобы навязать населению России «удобное», виртуальное восприятие общественного процесса, приложили официальные органы. Рост масштабов ложного сознания «превращает общество в... подобие толпы, манипулирование которой становится делом техники».

Утрата общественным сознанием устойчивости явилась предпосылкой возникновения и широкого распространения «манипулируемого» сознания. Среди причин автор указывает: «плутократические тенденции в осуществлении политической власти теми, кто ... узурпировал право распоряжаться национальным достоянием», а также превращение СМИ в инструмент «препарирования информации» в интересах собственников журналов и газет.

Многообразные метаморфозы общественного сознания, являясь индикатором неблагополучия и отражением социальных болезней, в свою очередь, порождают «парадоксальность общественного сознания». Отмеченное явление обрело массовость, став характерными признаком современной России с ее неустойчивым развитием и неопределенным вектором исторического процесса.

В России сформировалось особое правовое сознание, при котором формальные права и процедурные правила играют подчиненную роль. Центральное место, скорее, занимает социальная справедливость. Справедливость, как и личные взаимоотношения, основывается на авторитете и доверии, а не на формальных критериях. В России на протяжении веков взаимоотношения – как на уровне бытовых отношений, так и на политическом уровне – регулировались не правом, а обычаями, традициями. Правовой нигилизм как политических элит, так и народа в современном российском обществе проистекает из опыта и привычек многих поколений. Право обладает ограниченной дееспособностью, оно является выражением политики и власти. Закону подчиняются тогда, когда считают это справедливым.

На фоне правового нигилизма проявляется еще одна особенность российского менталитета – вера в доброго правителя. В России всегда была велика пропасть между народом и элитами. В крепостнической России это пропасть между основной массой населения – крестьянами и помещиками, в советский период – между партноменклатурой и всеми основными слоями общества, в современный период – все увеличивающаяся пропасть между чиновниками всех уровней и гражданским обществом. Сложилась определенная психологическая формула: «Царь-то хороший, да бояре плохие». Харизматичность российского менталитета хорошо известна и детально разобрана в

научной литературе. Подчеркнем лишь особенность российской харизматичности, которая проявляется в том, что вера в доброго правителя соединяется с уверенностью, что он должен быть истинным, «настоящим». Под «настоящим правителем» понимается право наследования в случае монархического правления или легитимности в случае избрания. Но в данное понятие вписывается также выполнение той программы, с которой проявляется тот или иной харизматический лидер на российской политической сцене. Если заявленная программа не выполняется, заменяется другой или уже исчерпала себя, а лидер продолжает играть свою роль, то возникают общественные настроения, которые можно передать фразой «А царь-то – не настоящий». И в этом случае народ ментально меняет свою верноподданническую позицию на диаметрально противоположную. Достаточно вспомнить лишь некоторые эпизоды отечественной истории. Лжедмитрий I победоносно вступает в Москву, поскольку на тот период в общественном сознании укореняется мысль о его подлинности, и спустя полгода его убивает толпа, разуверившаяся в его подлинности. Примерно такое же отношение мы наблюдаем во время Пугачевского бунта. Выдавший себя за императора Петра III, Емельян Пугачев на начальной стадии крестьянского восстания пользовался широкой поддержкой крестьян, казаков и представителей нерусских народов Поволжья, примкнувших к нему. Затем по мере его военных неудач вера в подлинного царя ослабевала и к концу восстания сошла на нет.

Таким образом, мы можем констатировать, что особенности формирования российского менталитета связаны с его историческими корнями. Это, во-первых. А во вторых, на его формирование оказывают влияние те социально-экономические процессы, которые происходят в российском обществе сегодня:

- радикальные изменения основ экономической, социальной, политической и духовной жизни, массовое перемещение в Россию различных групп населения из Ближнего зарубежья и возникновение вследствие этого новых межэтнических, демографических, территориальных и иных образований;
- социальное расслоение общества с крайне незначительным средним слоем и значительным слоем граждан, оказавшихся за чертой бедности;
- изменение и усложнение социальной структуры общества, появление в ней новых социальных групп, усиление вертикальной и горизонтальной мобильности;
- расширение информации и на ее основе переоценка уроков прошлого, настоящего и перспектив будущего.

Проецируется из прошлого в настоящее и такая черта национального менталитета, как крайность в оценках и суждениях. Маятник общественного мнения отклоняется очень далеко от центра, сметая все на своем пути. Амплитуда колебаний этого маятника в нашем российском менталитете намного шире, а, следовательно, и разрушительнее, чем соответствующая амплитуда западноевропейского менталитета.

Наша практика знает много примеров, когда из отечественной культуры выплывает многое, а порой и все позитивное, наработанное историей. Затем – мучительное и, как правило, долгое осознание бесперспективности нового мифа. Мифотворчество, безусловно, черта российского менталитета. Трудно не согласиться с Л.Н. Гумилевым, «что мифы, несмотря на прозрачность своей природы, совсем не безвредны. Они норовят подменить собой эмпирическое обобщение наблюдаемых фактов, т.е. занять место науки и заменить аргументацию декларациями, подлежащими принятию без критики. Проверить данный миф невозможно. Когда миф торжествует, то наступает подлинный упадок науки, да и всей культуры»⁶.

В современных условиях мифологичность национального менталитета не только сохраняется, но приобретает новые черты.

Обращает на себя внимание стойкое совмещение в общественном сознании россиян обширных анклавов рационально-критического мышления, характерных для есте-

ственнонаучных и инженерно–технических сфер деятельности, и широкого, неиссякающего потока социальной мифологии, устойчиво воспроизводимой в общественно–политической жизни общества. Причины подобного воспроизводства в немалой степени обусловлены провалом в области массового социогуманитарного, обществоведческого образования, что превращает население огромной России в удобный объект корыстных и циничных манипуляций со стороны правящих кланов и клик.

Акцентируя внимание на сходстве мифологии и идеологии, Ж.Т. Тощенко констатирует: «Идеология выполняет определенные социальные функции, отражает интересы определенного класса, сообщества, выражающиеся в типах мышления и поведения, а также в программах социального действия». Тем самым вскрывается изощренная «механика» манипулирования, при помощи которой общественному сознанию россиян навязываются способы мышления, восприятия действительности и массового поведения, обеспечивающие реализацию интересов правящего класса, назойливо выдаваемые многими СМИ за интересы общенациональные.

Анализируя природу и современные процессы возникновения стереотипов, трудно заметить главенствующую роль средств массовой информации и официальной пропаганды в формировании искаженного восприятия реальности. Причина в том, что практический опыт конкретного человека неизбежно ограничен и «представление о том или ином процессе или явлении соотносится не с реальным объектом, а с имеющейся информацией». И потому не является преувеличением один из принципов американской журналистики: «если чего–то не было в СМИ, того, можно считать, не было и в жизни». Отсюда очевидна двуединая социальная роль средств массовой информации: объективное информирование и просвещение широких слоев населения. Однако вместо этого в российских условиях СМИ широко используются в качестве инструмента манипулирования общественным сознанием. Манипуляция общественным сознанием тормозит развитие творческих сил народа.

Любой национальный менталитет, и российский в том числе, нельзя рассматривать как однозначно заданную величину, некую константу. Поскольку один и тот же этнос в разные фазы своего этногенеза (периода развития) ведет хозяйство разными способами, то и вырабатывает разный менталитет. В этом отношении интересным является замечание Л.Н.Гумилева. «Так называемый «национальный характер» – миф, ибо для каждой новой эпохи он будет другим, даже при ненарушенности последовательности смены этногенеза»⁷.

Осознание отечественного менталитета в качестве переменной величины, которая видоизменяется и дополняется по мере исторического развития, чрезвычайно важно при проведении любых экономических и политических реформ. В современной научной литературе по проблемам национального менталитета сложились два взаимоисключающих подхода. Сторонники одного склонны видеть в современном российском менталитете прямое продолжение российского менталитета прошлого с его уникальностью и неповторимостью для других народов и культур. Другой подход предполагает, что в современных условиях, для которых характерны общие экономические и политические тенденции, высокая степень информированности, взаимозависимости стран и народов, нет необходимости в глубоком анализе исторических корней отечественного менталитета и учете его характерных черт в экономических и политических преобразованиях.

Представляется, что и тот и другой подходы носят односторонний характер. Как показывает наша современная социально–экономическая практика, влияние национального менталитета на различные процессы в обществе весьма велико. Поэтому наиболее продуктивным видится метод сравнения. Факторы, влияющие на эволюцию национального менталитета (экономические, политические, психологические и т.д.), действуют без границ, создавая контуры современной цивилизации. Например, под влияни-

ем этих факторов сложилось общее направление на развитие рыночных отношений при регулирующей роли государства. Однако, действуя в самобытных социокультурных условиях России, эти факторы формируют иные черты национального менталитета.

Значение российской социокультурной среды, пропускающей через себя факторы, влияющие на развитие национального менталитета, безусловно, велико. Поэтому сама эта среда выступает как самостоятельный фактор эволюции отечественного менталитета.

Литература

1. Современный словарь иностранных слов. М., 1993. С. 371.
2. Революция неизбежна // Хлеб и Воля. 1903. №4. Ноябрь. С.5.
3. Кропоткин П.А. Великая Французская революция. 1789–1793. М., 1979. С.5.
4. Кропоткин П.А. Речи бунтовщика. СПб, 1906. С.13.
5. Тощенко Ж.Т. Парадоксальный человек. М., 2001. С.55.
6. Гумилев Л.Н.Тысячелетие вокруг Каспия. Баку, 1991. С.296.
7. Гумилев Л.Н. Этногенез и биосфера земли. Л.: Гидрометеиздат, 1990. С.358.

МЕЖДУНАРОДНО-ПРАВОВЫЕ ПРОБЛЕМЫ ФОРМИРОВАНИЯ ОБЩЕГО ИНФОРМАЦИОННОГО ПРОСТРАНСТВА СТРАНАМИ СНГ

Г.В. Алексеев

Научный руководитель – д.соц.н., профессор Л.Я. Орлова

Международно-правовое обеспечение информационного обмена между странами СНГ является важной составляющей безопасного развития региона. Заключенные на сегодняшний день в рамках СНГ международные договоры демонстрируют разносторонний характер информационного сотрудничества в рамках СНГ. А.С. Третьяков полагает, что в рамках СНГ за 10 лет его существования уже сложилась определенная система сотрудничества, ориентированная на взаимодействие его участников в интересах обеспечения национальной и коллективной безопасности.¹

После распада СССР страны, объединившиеся в состав СНГ, принимают Соглашение о сотрудничестве в области информации (Бишкек, 1992).² Данный документ рассматривает международный информационный обмен как форму международного сотрудничества. Субъекты информационного обмена именуются в тексте соглашения «участниками информационных обменов», к их числу соглашение относит «граждан, их объединения, редакции средств массовой информации, государственные и правительственные учреждения, а также их представительства на территории государств-участников Содружества».

В дальнейшем Концепция формирования информационного пространства Содружества Независимых Государств (Москва, 18 октября 1996 г.)³ производит деление всех субъектов информационного обмена на три категории:

- собственник информационных ресурсов, информационных систем, технологий и средств их обеспечения – субъект, осуществляющий владение и пользование указанными объектами и реализующий полномочия владения, пользования, распоряжения указанными объектами;
- владелец информационных ресурсов, информационных систем, технологий и средств их обеспечения – субъект, осуществляющий владение и пользование указанными объектами и реализующий полномочия распоряжения в пределах, установленных законом;
- пользователь (потребитель) информации – субъект, обращающийся к информационной системе или посреднику за получением необходимой ему информации и пользующийся ею в соответствии с порядком, установленным собственником или владельцем информации. При этом термин «участник информационного обмена» не используется, а правовое положение субъекта обмена зависит от его роли в рамках информационного правоотношения.

Однако, вместе с тем, Соглашение о сотрудничестве в области информации закрепило ряд международно-правовых обязательств, которые приняли на себя страны участники СНГ. Исходя из анализа документа, обязанности, принятые на себя государствами СНГ, можно разделить на две группы. Во-первых, это обязанности содействовать всем формам сотрудничества между различными средствами массовой информации

¹ Третьяков А.С. Противодействие международному терроризму: военные аспекты сотрудничества государств-участников СНГ // Право и безопасность. № 4 (5). Декабрь 2002.

² Соглашение о сотрудничестве в области информации // Бюллетень международных договоров. 1993. № 10. С. 30.

³ Решение о Концепции формирования информационного пространства Содружества Независимых Государств (Москва, 18 октября 1996 г.) // Ведомственное приложение к «Российской газете» от 7 декабря 1996 г., Информационный вестник Совета глав государств и Совета глав правительств СНГ «Содружество». 1996. № 4. Ср. 87.

своих стран для обеспечения широкого и свободного распространения и обмена информацией, включая создание общих информационных структур, а также содействовать обмену специалистами в сфере информационных обменов и созданию необходимых условий для профессиональной деятельности представителей средств массовой информации. Во-вторых, это обязанности не препятствовать доступу к официальным, общественным и другим источникам информации, а также распространению приобретенной на законных основаниях информации представителям средств массовой информации каждой из Сторон, а также обеспечивать на своей территории свободное распространение официальных информационных бюллетеней дипломатическими и другими официальными представительскими органами другой страны. Другими словами, государственные органы стран СНГ приняли на себя обязательства содействовать международным информационным контактам и не препятствовать им.⁴

В соответствии со статьей 9 Соглашения о сотрудничестве в области информации обеспечение выполнения обозначенных выше обязанностей возложено на министерства иностранных дел сторон или иные ведомства, которые по поручению правительств будут проводить по мере необходимости встречи для подведения итогов сотрудничества и разрешения возможных спорных вопросов, как правило, не реже одного раза в год.

26 мая 1995 года в Минске Совет глав правительств Содружества Независимых Государств решил поручить Исполнительному секретариату Содружества Независимых Государств с участием Межгосударственного координационного совета Межгосударственной телерадиокомпании «Мир», полномочных представителей и заинтересованных ведомств государств-участников Содружества Независимых Государств организовать работу по подготовке предложений о проведении государствами-участниками Содружества согласованной политики по формированию единого информационного пространства Содружества Независимых Государств.⁵

Результатом этого решения стало принятие Решения Совета глав правительств СНГ о Концепции формирования информационного пространства СНГ.⁶ В соответствии с концепцией, формирование информационного пространства СНГ – совокупность согласованных на межгосударственном уровне мероприятий и условий по развитию на взаимовыгодной основе межгосударственных информационных обменов в интересах сотрудничества государств-участников СНГ в согласованных сферах деятельности и в соответствии с международными принципами распространения информации в целях обеспечения и удовлетворения информационных потребностей граждан, предприятий, организаций и органов управления государств Содружества.⁷

В свою очередь, международные соглашения в соответствии со статьей 2 Соглашения об информационном обеспечении выполнения многосторонних соглашений (Москва, 24 сентября 1993 г.) сами нуждаются в определенном «порядке получения и использования информации от государств-участников Содружества и использования имеющихся у них информационных структур, банков данных, каналов связи, систем защиты и стандартов обмена информацией. При этом следует обеспечить техническую,

⁴ Бюллетень международных договоров. 1993. № 10. С. 30.

⁵ Решение о проведении согласованной политики по формированию единого информационного пространства Содружества Независимых Государств (Минск, 26 мая 1995 г.) // Ведомственное приложение к «Российской газете» от 7 июля 1995 г., Информационном вестнике Совета глав государств и Совета глав правительств СНГ «Содружество». 1995. № 2. С. 139.

⁶ Решение о Концепции формирования информационного пространства Содружества Независимых Государств (Москва, 18 октября 1996 г.) // Ведомственное приложение к «Российской газете» от 7 декабря 1996 г., в специальном выпуске «Инвест-курьер» журнала «Панорама приватизации», январь 1997 г., стр. 74, Информационный вестник Совета глав государств и Совета глав правительств СНГ «Содружество». 1996. № 4. С. 87.

⁷ Концепция формирования информационного пространства Содружества Независимых Государств (Москва, 18 октября 1996 г.) // Ведомственное приложение к «Российской газете» от 7 декабря 1996 г.

программную, организационную совместимость сетей связи и информационных систем, правовой режим распространения информации».⁸

В соответствии со статьей 1 Соглашения об информационном обеспечении, «... исполнительный Секретариат СНГ несет функции накопления, систематизации и анализа информации по выполнению многосторонних соглашений, заключенных между государствами-участниками Содружества».⁹ Развитие нормативно-правовой базы взаимодействия в информационной сфере и повышение эффективности информационного взаимодействия в согласованных сферах, представляет собой реализацию принципа информационного сотрудничества в рамках СНГ.

Существенное значение здесь играет Соглашение об обмене правовой информацией (Москва, 21 октября 1994 г.), где в статье 1 отмечается, что «... с целью эффективного выполнения многосторонних и двусторонних соглашений между государствами-участниками Содружества, создания единого информационного пространства и развития сотрудничества в правовой сфере Стороны согласились осуществлять обмен правовой информацией».¹⁰

Из содержания статьи 3 Соглашения об обмене правовой информацией необходимо сделать следующие выводы:

Во-первых, «Стороны, осуществляя согласованные меры по созданию единой распределенной системы обмена правовой информацией, соответственно определяют:

- полномочный орган (министерство, комитет, ведомство), осуществляющий координацию работ по созданию национальных банков данных, используемых для межгосударственного обмена правовой информацией;
- перечень субъектов-пользователей информацией».

Во-вторых, предметом соглашения является правовой режим информации – нормативно установленные правила, определяющие степень открытости, порядок документирования, доступа, хранения, распространения и защиты информации.¹¹

На государственные органы статьей 4 Соглашения об обмене правовой информацией возлагается обязанность поддерживать национальные ресурсы правовой информации, определенные для совместного использования (банки данных) в актуальном (контрольном) состоянии и «нести ответственность за полноту, точность, достоверность и своевременность предоставления информационных данных и услуг».¹²

В свое время Президентом РФ планировалось создание межгосударственной системы правовой информатизации в целях поддержания и развития информационно-правового сотрудничества с заинтересованными государствами-членами Содружества Независимых Государств и формирования единого информационно-правового пространства. Функции генерального подрядчика по проведению работ в области создания указанной межгосударственной системы предлагалось возложить на Федеральное агентство правительственной связи и информации при Президенте Российской Федерации.¹³

Другим проявлением реализации принципа информационного сотрудничества в рамках СНГ является Соглашение об обмене информацией в области внешнеэкономич-

⁸ Соглашение об информационном обеспечении выполнения многосторонних соглашений (Москва, 24 сентября 1993 г.) // Бюллетень международных договоров. 1994. № 1. С. 14.

⁹ Там же.

¹⁰ Соглашение об обмене правовой информацией (Москва, 21 октября 1994 г.) // Бюллетень международных договоров. 1995. № 2. С. 34.

¹¹ Там же.

¹² Там же.

¹³ Указ Президента РФ от 19 октября 1993 г. № 1665 «Об информационно-правовом сотрудничестве Российской Федерации с государствами-членами Содружества Независимых Государств» // Собрание актов Президента и Правительства Российской Федерации от 25 октября 1993 г., № 43, ст. 4108.

ческой деятельности (Москва, 24 сентября 1993 г.).¹⁴ В соответствии со статьей 1 данного документа договаривающиеся стороны будут обмениваться нормативными актами, информацией и текстами документов по следующим направлениям: «о внешнеэкономических отношениях, включая валютное регулирование; о таможенных пошлинах и тарифах; о квотировании и лицензировании товаров; о деятельности на рынках третьих стран», и т.д. по тексту статьи.

В соответствии со статьей 6 Соглашения об обмене информацией в области внешнеэкономической деятельности, «Договаривающиеся Стороны обязуются не осуществлять продажу или передачу экономической информации, являющейся предметом настоящего Соглашения, третьим странам без согласия Договаривающейся Стороны, представившей такую информацию. При необходимости Договаривающиеся Стороны согласуют порядок передачи информации третьим странам». Для демонстрации роли государственных органов в информационном сотрудничестве также важно, что в соответствии со статьей 7 того же документа для выполнения Соглашения издаются соответствующие внутренние акты с поручениями министерствам и ведомствам, т.е. его выполнение также возлагается на государственные органы сторон.

Соглашение о принципах и формах взаимодействия государств-участников СНГ в области использования архивной информации (Минск, 4 июня 1999 г.) также посвящено информационному сотрудничеству, так как в соответствии со статьей 1 данного документа «Стороны рассматривают обмен архивной информацией как важнейшую форму сотрудничества. Стороны всемерно содействуют обеспечению широкого обмена архивной информацией в соответствии с их национальным законодательством».

Статья 3 Соглашения в области использования архивной информации устанавливает национальный режим для стран СНГ в архивном деле. В соответствии с ним «Стороны обеспечивают юридическим и физическим лицам государств-участников настоящего Соглашения доступ к архивной информации, хранящейся в их государственных архивах, в соответствии с порядком, установленным их национальным законодательством».

Более детально сотрудничество в области обмена архивной информацией рассматривается в ведомственных соглашениях между архивными службами государств СНГ. Например, Соглашение о сотрудничестве между Государственной архивной службой Российской Федерации и Государственной архивной службой Республики Молдова (Киев, 3 ноября 1994 г.) фактически установило общий для Молдавии и России режим хранения архивной информации. Данный режим сформирован такими обязательствами сторон, как:

- обмен опытом работы между архивными учреждениями по интересующим их направлениям деятельности государственных архивов;
- обмен информацией по возникающим у них проблемам архивного дела и обмен специалистами-архивистами на взаимной основе;
- обмен законодательными и регламентирующими документами по организации и порядку работы архивных учреждений, а также архивоведческой и научно-методической литературой по архивному делу;
- обмен списками справочных материалов по архивным документам, опубликованным в течение календарного года;
- осуществление совместных публикаций архивных документов и организовывать документальные выставки по согласованной тематике.¹⁵

¹⁴ Бюллетень международных договоров 1994 г., N 1, стр. 17.

¹⁵ Ст. 2 - 9 Соглашения о сотрудничестве между Государственной архивной службой Российской Федерации и Государственной архивной службой Республики Молдова (Киев, 3 ноября 1994 г.).

Ряд обязательств, однако, все же являются общими для всех стран СНГ. Так, в частности, статья 6 Соглашения о принципах и формах взаимодействия государств-участников СНГ в области использования архивной информации (Минск, 4 июня 1999 г.) демонстрирует, что национальный режим касается только открытой информации, так как на стороны возлагается обязанность по обеспечению взаимного обмена «открытой информацией о документах, находящихся на государственном архивном хранении».¹⁶

Другим примером отраслевого информационного сотрудничества в рамках СНГ может служить Соглашение об обмене информацией в сфере образования государств-участников Содружества Независимых Государств (Минск, 31 мая 2001 г.).¹⁷ В соответствии со статьями 1–4 данного документа Стороны обмениваются информацией о вопросах образования, подготовки и аттестации научных и научно-педагогических кадров на основе действующих национальных правил в сфере образования, международных соглашений в информационной сфере и используют для этого общие и национальные информационные ресурсы.

С другой стороны, региональное сотрудничество в рамках СНГ привело к региональному партнерству в рамках ЕврАзЭС. Решение Межгосударственного Совета Республики Беларусь, Республики Казахстан, Кыргызской Республики, Российской Федерации и Республики Таджикистан от 26 октября 1999 г. № 51 «О Соглашении о правовом обеспечении формирования Таможенного союза и Единого экономического пространства»¹⁸ занимает позицию относительно необходимости универсальных принципов информационного сотрудничества во всех отраслях. В соответствии со статьей 13 данного решения, «...в целях информационного обеспечения интеграционного взаимодействия, создания информационной основы для гармонизации национальных законодательств Стороны признают необходимость всестороннего информационного обмена». И стороны, «используя национальные информационные ресурсы, согласились создать в Интеграционном Комитете интегрированную базу законодательства, статистической и другой социально-экономической информации, включая копии эталонных банков данных национальных законодательных и иных нормативных правовых актов на электронных и бумажных носителях, и осуществлять ее регулярное обновление».¹⁹ При этом такое комплексное информационное сотрудничество осуществляется средствами органов власти в их интересе и включает в себя интегрированную базу законодательства и статистической и другой социально-экономической информации.²⁰

Таким образом, можно сделать вывод, что реализация принципа сотрудничества в информационной сфере между органами государственной власти стран СНГ привела к формированию общей информационной базы общедоступной информации, к которой граждане и организации стран СНГ могут получить доступ на принципах национального режима.

Следующим направлением деятельности является развитие социальных и культурных связей и реализация информационных прав и свобод граждан. Здесь важное место занимает целый ряд соглашений, направленных на регулирование функциониру-

¹⁶ Бюллетень международных договоров, декабрь 2000 г., № 12.

¹⁷ Соглашение об обмене информацией в сфере образования государств-участников Содружества Независимых Государств (Минск, 31 мая 2001 г.), Постановление Правительства РФ от 19 октября 2001 г. № 737. // Бюллетень международных договоров, март 2002 г., № 3.

¹⁸ СЗ РФ от 20 октября 2003 г. № 42. Ст. 3989.

¹⁹ Решение Межгосударственного Совета Республики Беларусь, Республики Казахстан, Кыргызской Республики, Российской Федерации и Республики Таджикистан от 26 октября 1999 г. № 51 «О Соглашении о правовом обеспечении формирования Таможенного союза и Единого экономического пространства» // СЗ РФ от 20 октября 2003 г. № 42. Ст. 3989.

²⁰ Статья 14. Там же.

ния средств массовой информации. Данная функция изначально, по мнению стран участниц СНГ, должна была возлагаться на независимые медиакорпорации.

Данный подход соответствует европейским стандартам в области информационных прав и свобод человека, отраженным, например, в Резолюции №2 4-ой Европейской Конференции Министров по политике в области средств массовой коммуникации №Журналистские свободы и права человека» (Прага, 7–8 декабря 1994 г.). Там, в частности, отмечается, что «поддержание и развитие подлинной демократии требуют наличия и укрепления свободной, независимой, плюралистической и ответственной журналистики. Это требование воплощается для журналистики в необходимости:

- информировать индивидуумов о деятельности органов государственной власти и частного сектора, предоставляя им таким образом возможность сформировать собственное мнение;
- позволять индивидуумам или группам высказывать их мнения, содействуя таким образом информированию органов государственной власти и частного сектора, а также всего общества, об этих мнениях;
- подвергать постоянному критическому рассмотрению деятельность различных органов власти».²¹

Реализация данного принципа связана с целым рядом соглашений в рамках СНГ, совершенных в отношении Межгосударственной телерадиокомпании. Соглашение о создании Межгосударственной телерадиокомпании (Бишкек, 9 октября 1992 г.) по своему содержанию представляет реализацию обозначенной выше статьи 2 Соглашения о сотрудничестве в области информации (Бишкек, 1992).²²

Устав МТРК провозгласил в качестве основной цели деятельности Компании «более полное и качественное удовлетворение потребностей граждан государств-учредителей во всесторонней объективной информации о событиях и процессах, происходящих в государствах-учредителях и на мировой арене, а также обеспечение средствами телевидения и радиовещания их культурных запросов».²³

Реализация целей и задач МТРК, исходя из видов деятельности телерадиокомпании, осуществляется посредством общих для всех СМИ приемов, а именно «производство и распространение телевизионных и радиопрограмм на территориях государств-учредителей» и сопряженные с этим виды деятельности. Спецификой деятельности МТРК является, во-первых, то, что МТРК создает продукцию, общую для всех стран СНГ, а во-вторых, МТРК обязано выделять эфирное время для официальной информации, предоставляемой уполномоченными на то представителями государств-учредителей.²⁴ Государственные органы, соответственно, обязаны задействовать МТРК в национальных интересах своей страны.

Анализ положений устава относительно юридического статуса МТРК позволяет сделать вывод, что компания является юридическим лицом и регистрируется в соответствии с законодательством Российской Федерации, на территориях государств-учредителей МТРК действует исходя из статуса юридического лица по российскому праву в соответствии с действующим законодательством этих государств.

Компания может открывать на территориях государств-учредителей и за пределами Содружества филиалы, корреспондентские пункты, дочерние предприятия и представительства, которые будут вести свою деятельность в соответствии с законодательством данного государства-учредителя. В случаях, когда филиалы и дочерние предпри-

²¹ Резолюции № 2 4-ой Европейской Конференции Министров по политике в области средств массовой коммуникации «Журналистские свободы и права человека» (Прага, 7–8 декабря 1994 г.).

²² Соглашение о сотрудничестве в области информации // Бюллетень международных договоров. 1993. № 10. С. 30.

²³ Решение о вопросах Межгосударственной телерадиокомпании от 22 января 1993 года.

²⁴ Там же.

ятия Компании учреждаются как юридические лица, они не отвечают по долгам и обязательствам Компании. Компания отвечает по долгам и обязательствам таких филиалов и дочерних предприятий в пределах своего долевого взноса в их уставный капитал.

Учредителями Компании на паритетных и договорных началах являются государства, подписавшие Соглашение, одобrivшие и признающие положения Устава Компании. Государства-учредители являются акционерами Компании через своих представителей, т.е. через государственные органы и предприятия. Акционеры Компании имеют право участвовать в управлении делами Компании (они также имеют другие административные полномочия) и несут обязанность вносить вклад (оплачивать акции) в случае увеличения уставного капитала Компании пропорционально своим долям в акционерном капитале, а также финансировать текущую оперативно-производственную деятельность Компании пропорционально своим долям в акционерном капитале. Другими словами, государства приняли на себя обязательства по финансированию МТРК. И вместе с тем учредители-акционеры обязаны в соответствии с уставом обеспечивать соблюдение международных прав журналистов. Таким образом, правовая конструкция управления МТРК не должна оказывать влияние на редакционную политику компании.

Впервые в международных соглашениях МТРК приобретает название «Мир» в Соглашении о международно-правовых гарантиях беспрепятственного и независимого осуществления деятельности Межгосударственной телерадиокомпании «Мир» (Ашхабад, 24 декабря 1993 г.).²⁵ Данное соглашение направлено на придание МТРК «Мир» политической и финансовой самостоятельности, что в целом не соответствует статусу юридического лица, а, напротив, свойственно международным организациям.

Эта юридическая особенность привела к некоторым проблемам с функционированием МТРК, отраженным, в частности, в Решении Экономического Суда СНГ от 27 июня 2000 г. № 01-1/1-2000 «Об ограничении привилегий и иммунитетов, предоставленных МТРК «Мир» при осуществлении коммерческой деятельности, не связанной с профессиональными функциями». Там сделан вывод, что, «несмотря на известную сложность разграничения гуманитарных (публично-правовых) и частно-правовых (производственных и коммерческих) целей профессиональной деятельности МТРК «Мир» и создание ее в такой организационно-правовой форме, как закрытое акционерное общество, не означает, что МТРК «Мир» как международное юридическое лицо теряет возможность обладать привилегиями и иммунитетами, предоставляемыми ей в договорном порядке». Хотя, вместе с тем, отмечается, что «статус МТРК «Мир» как юридического лица не носит побочного, вспомогательного характера, как это имеет место в международных организациях гуманитарного типа, а служит средством для реализации профессиональной международной деятельности, имеющей гуманитарную направленность».

Сущность гарантий самостоятельности телекомпании заключается в следующих двух аспектах:

- государства-участники на своих территориях предоставляют Компании и ее должностным лицам, а также представителям Компании такие привилегии и иммунитеты, в частности: архивы и документы Компании, а также ее помещения неприкосновенны; имущество Компании не подлежит обыску, реквизиции, конфискации и изъятию в административном порядке; представители Компании пользуются при выполнении своих официальных функций иммунитетом от личного ареста или задержания, а также от административной и судебной юрисдикции.

²⁵ Соглашении о международно-правовых гарантиях беспрепятственного и независимого осуществления деятельности Межгосударственной телерадиокомпании «Мир» (Ашхабад, 24 декабря 1993 г.) Соглашение ратифицировано Российской Федерацией Федеральным законом от 3 августа 1995 г. N 124-ФЗ // Информационный вестник Совета глав государств и Совета глав правительств СНГ «Содружество». 1993. № 5. С. 81. Бюллетень международных договоров. 1995. № 3. С. 4.

- компания, ее имущество освобождаются от прямых налогов и сборов любого характера, кроме платежей за коммунальные и другие подобные услуги, а также Компания освобождается от таможенных сборов и ограничений при ввозе и вывозе предметов, предназначенных для использования в целях, предусмотренных Уставом МТРК «Мир».²⁶

Затем были и другие документы относительно функционирования МТРК «Мир», в частности, решение о Проекте многофункциональной телерадиовещательной спутниковой системы «Мир» для государств-участников Содружества Независимых Государств (Москва, 9 сентября 1994 г.),²⁷ решение о проекте Соглашения о дальнейшей деятельности Межгосударственной телерадиокомпании «Мир» Содружества Независимых Государств (Минск, 26 мая 1995 г.), решение о неотложных мерах по развитию Межгосударственной телерадиокомпании «Мир» (Москва, 17 января 1997 г.).²⁸

Несмотря на то, что соглашение о неотложных мерах было достигнуто, сами неотложные меры, однако, не приняты до сих пор. Например, Межгосударственный Координационный Совет МТРК «Мир» должен был разработать и в установленном порядке внести на очередное заседание Совета глав правительств Содружества проект Положения о статусе радио-, телевизионного и телекоммуникационного каналов межгосударственного общения Содружества Независимых Государств. Данного положения на сегодняшний день нет.

Признание сторонами того факта, что выполнение органами исполнительной власти и управления Соглашения о международно-правовых гарантиях беспрепятственного и независимого осуществления деятельности Межгосударственной телерадиокомпании «Мир» от 24 декабря 1993 года целесообразно осуществлять в соответствии с Решением Экономического Суда Содружества Независимых Государств от 21 декабря 1995 года, по сути дела, означает, что государственные органы стран СНГ регулярно нарушали данные международные соглашения.

Несмотря на сложности в реализации проекта МТРК «Мир», по пути создания международных телерадиокомпаний пошел и Союз Белоруссии и России в рамках Договора между Российской Федерацией и Республикой Беларусь о совместной телерадиовещательной организации Союза Беларуси и России (Москва, 22 января 1998 г.).²⁹ В качестве цели создания телерадиокомпании провозглашается создание общего информационного пространства и ей предоставляется время на каналах государственного телевидения.

Другая юридическая конструкция призвана способствовать формированию единого информационного пространства СНГ посредством распространения печатных СМИ. В этой области заключено Соглашение о сотрудничестве государств-участников Содружества Независимых Государств в области периодической печати (Минск, 4 июня 1999 г.).³⁰ Из содержания данного соглашения можно сделать вывод, что государства, входящие в состав СНГ, воздействуют на печатные СМИ с целью формирования в не-

²⁶ См. подробнее Решение Экономического Суда СНГ от 15 сентября 1998 г. № 01-1/3-98 «О толковании пункта 1 статьи 6 Соглашения о международно-правовых гарантиях беспрепятственного и независимого осуществления деятельности МТРК «Мир» от 24 декабря 1993 г.

²⁷ Решение о Проекте многофункциональной телерадиовещательной спутниковой системы «Мир» для государств - участников Содружества Независимых Государств (Москва, 9 сентября 1994 г.) // Информационный вестник Совета глав государств и Совета глав правительств СНГ «Содружество». 1994. № 2. С. 113.

²⁸ Решения Совета глав правительств Содружества Независимых Государств.

²⁹ Договор между Российской Федерацией и Республикой Беларусь о совместной телерадиовещательной организации Союза Беларуси и России (Москва, 22 января 1998 г.) // СЗ РФ от 22 ноября 1999 г., № 47, ст. 5626, Бюллетень международных договоров, февраль 2000 г., № 2.

³⁰ Соглашение о сотрудничестве государств-участников Содружества Независимых Государств в области периодической печати (Минск, 4 июня 1999 г.) // Бюллетень международных договоров, декабрь 2000 г., № 12.

которой степени общей редакционной политики с учетом национальной специфики государств. Вместе с тем следует отметить, что в сфере печатных СМИ не отвергается механизм создания транснациональных юридических лиц. В соответствии со статьей 7 Соглашения о сотрудничестве государств-участников СНГ в области периодической печати государства «содействуют созданию и деятельности совместных предприятий, в том числе с государственным участием, в области периодической печати, а также иным формам сотрудничества».³¹

Контроль за выполнением соглашения возлагается на «компетентные органы Сторон», которые в соответствии со статьей 8 Соглашения «регулярно обмениваются сведениями и информацией о ходе выполнения настоящего Соглашения». Под компетентными органами следует понимать, в силу содержания Соглашения о сотрудничестве в области информации (Бишкек, 1992)³² и Концепции формирования информационного пространства Содружества Независимых Государств (Москва, 18 октября 1996 г.),³³ органы исполнительной власти в целом и министерства иностранных дел Сторон, в частности.

Нельзя не обратить внимания на информационное сотрудничество, направленное на реализацию информационных прав и свобод граждан в Рамках Евразийского Экономического сообщества, которое активно развивается в настоящее время. Ключевую роль здесь играет Соглашение о сотрудничестве в распространении массовой информации (Астана, 20 сентября 2002 г.)³⁴ утвержденное Решением Межгосударственного Совета Евразийского экономического сообщества от 20 сентября 2002 г. № 79 «О Соглашении о сотрудничестве в распространении массовой информации».³⁵

Соглашение о сотрудничестве в распространении массовой информации основано на принятых ранее в рамках СНГ правовых документах о сотрудничестве в сфере информации и направлено на расширение сотрудничества.³⁶ Рабочие группы из представителей соответствующих государственных органов совместно с представителями предприятий и организаций, действующих в области информации, координируют взаимодействие по выполнению странами ЕврАзЭС принятых на себя взаимных обязательств в информационной сфере.

В соответствии со Статьей 12 Соглашения о сотрудничестве в распространении массовой информации, государственные органы законодательной власти сторон приняли на себя обязанность по сближению законодательства своих государств в сферах, связанных с предметом Соглашения, в том числе в налоговой, таможенной, валютной, в части финансового регулирования деятельности редакций средств массовой информации, а также организаций, занимающихся распространением продукции средств массовой информации.

³¹ Бюллетень международных договоров, декабрь 2000 г., № 12.

³² Соглашение о сотрудничестве в области информации // Бюллетень международных договоров. 1993. № 10. С. 30.

³³ Решение о Концепции формирования информационного пространства Содружества Независимых Государств (Москва, 18 октября 1996 г.) // Ведомственное приложение к «Российской газете» от 7 декабря 1996 г., Информационный вестник Совета глав государств и Совета глав правительств СНГ «Содружество». 1996. № 4. С. 87.

³⁴ Соглашение о сотрудничестве в распространении массовой информации (Астана, 20 сентября 2002 г.) // Бюллетень международных договоров, ноябрь 2003 г., № 11.

³⁵ Соглашение утверждено постановлением Правительства РФ от 31 марта 2003 г. №174, вступило в силу для РФ 21 апреля 2003 г.

³⁶ Соглашения в рамках СНГ рассмотрены выше: Соглашение о сотрудничестве в области информации государств – участников Содружества Независимых Государств от 9 октября 1992 года, Соглашение о сотрудничестве в области периодической печати государств-участников Содружества Независимых Государств от 4 июня 1999 года и положения Концепции формирования информационного пространства Содружества Независимых Государств от 18 октября 1996 года.

Таким образом, государства СНГ и ЕврАзЭС гарантируют своим гражданам свободный доступ к информационным ресурсам друг друга в соответствии с международно-правовыми гарантиями, однако существующие соглашения полностью обходят такой аспект, как защита граждан от опасной и вредоносной информации, в то время как единое информационное пространство предусматривает общие и общие требования к легальности информации. Сближение законодательства государств ЕврАзЭС в информационной сфере необходимо снабдить не только финансовыми гарантиями, но и едиными стандартами в сфере диффамации, пропагандистских действий, иных потенциальных информационных правонарушений.

Важная сфера информационного сотрудничества стран СНГ связана с необходимостью информационного сотрудничества государственных органов между собой с целью повышения эффективности управления и решения совместных задач. Среди таких задач важное место занимает охрана внешних границ государств СНГ, поскольку от ее решения напрямую зависит строительство общего рынка.

В соответствии с Соглашением о порядке финансирования работ по созданию межгосударственной подсистемы информационного обмена между пограничными войсками государств-участников Содружества Независимых Государств (АСИО-ПВ СНГ) (Бишкек, 9 октября 1997 г.), создание данной системы было возложено на Федеральное агентство правительственной связи и информации при Президенте Российской Федерации как на генерального исполнителя. ФАПСИ, соответственно, применило при создании системы собственные стандарты в сфере защиты информации, которые приняты в РФ. В свою очередь, в рамках ЕврАзЭС общие принципы организации информационного взаимодействия в сфере охраны государственных границ стали предметом Соглашения об информационном взаимодействии государств-членов Евразийского экономического сообщества по пограничным вопросам (Алматы, 14 сентября 2001 г.)³⁷

Сопоставляя содержание Статьи 5 Соглашения об информационном взаимодействии государств-членов Евразийского экономического сообщества по пограничным вопросам и Концепции межгосударственной подсистемы информационного обмена между пограничными войсками государств-участников СНГ, нельзя не отметить, что содержание принципа открытости информационной системы более полно раскрыто именно в документах ЕврАзЭС. В частности установлено, что информационные ресурсы по пограничным вопросам, предоставляемые Сторонами для межгосударственного обмена, являются открытыми и общедоступными для пользователей информационных ресурсов Сторон.

Информационные ресурсы ЕврАзЭС по пограничным вопросам, в отличие от информационных ресурсов СНГ, в соответствии с достигнутыми договоренностями включают в себя не только информацию с грифом «секретно», но также совершенно секретные сведения и сведения особой важности. Соответственно, такие информационные ресурсы подлежат обязательному учету и защите.³⁸

Целями защиты информационных ресурсов являются предотвращение угрозы безопасности Сторон при любых формах несанкционированного доступа и незаконного вмешательства в информационные ресурсы и системы, сохранение межгосударственных и государственных секретов каждой из Сторон, конфиденциальности документированной информации и имеющихся в информационных системах персональных данных, предотвращение утечки, хищения, утраты, искажения и подделки информации. Данные функции возлагаются на специальные службы сторон, в компетенцию которых входит защита информации.

³⁷ Решение Межгосударственного Совета Евразийского экономического сообщества от 20 сентября 2002 г. № 79 «О Соглашении о сотрудничестве в распространении массовой информации» // Информационный бюллетень ЕврАзЭС. 2002. № 4.

³⁸ Информационный бюллетень ЕврАзЭС. 2002. № 4.

Таким образом, автор полагает, что в географических границах СНГ при продвижении сотрудничества государств в рамках ЕврАзЭС создана правовая основа единого информационного пространства, которое образовано информационными ресурсами различных категорий доступа. Значение положений международного права относительно обязательств по взаимному предоставлению закрытой информации и ее защите можно уяснить на основе анализа системы защиты закрытой информации в РФ.

Литература

1. Соглашение о сотрудничестве в области информации // Бюллетень международных договоров. 1993. № 10. С. 30.
2. Решение о Концепции формирования информационного пространства Содружества Независимых Государств (Москва, 18 октября 1996 г.) // Ведомственное приложение к «Российской газете» от 7 декабря 1996 г., Информационный вестник Совета глав государств и Совета глав правительств СНГ «Содружество». 1996. № 4. С. 87.
3. Решение о проведении согласованной политики по формированию единого информационного пространства Содружества Независимых Государств (Минск, 26 мая 1995 г.) // Ведомственное приложение к «Российской газете» от 7 июля 1995 г., Информационном вестнике Совета глав государств и Совета глав правительств СНГ «Содружество». 1995. № 2. С. 139.
4. Соглашение об информационном обеспечении выполнения многосторонних соглашений (Москва, 24 сентября 1993 г.) // Бюллетень международных договоров. 1994. № 1. С. 14.
5. Соглашение об обмене правовой информацией (Москва, 21 октября 1994 г.) // Бюллетень международных договоров. 1995. № 2. С. 34.
6. Указ Президента РФ от 19 октября 1993 г. № 1665 «Об информационно-правовом сотрудничестве Российской Федерации с государствами - членами Содружества Независимых Государств» // Собрание актов Президента и Правительства Российской Федерации от 25 октября 1993 г., № 43, ст. 4108.
7. Соглашение об обмене информацией в сфере образования государств – участников Содружества Независимых Государств (Минск, 31 мая 2001 г.), Постановление Правительства РФ от 19 октября 2001 г. № 737. // Бюллетень международных договоров, март 2002 г., № 3.
8. Соглашение о сотрудничестве в области информации // Бюллетень международных договоров. 1993. № 10. С. 30.
9. Соглашении о международно-правовых гарантиях беспрепятственного и независимого осуществления деятельности Межгосударственной телерадиокомпании «Мир» (Ашхабад, 24 декабря 1993 г.) Соглашение ратифицировано Российской Федерацией Федеральным законом от 3 августа 1995 г. N 124-ФЗ // Информационный вестник Совета глав государств и Совета глав правительств СНГ «Содружество». 1993. № 5. С. 81. Бюллетень международных договоров. 1995. № 3. С. 4.
10. Соглашение о сотрудничестве государств-участников Содружества Независимых Государств в области периодической печати (Минск, 4 июня 1999 г.) // Бюллетень международных договоров, декабрь 2000 г., № 12.
11. Соглашение о сотрудничестве в распространении массовой информации (Астана, 20 сентября 2002 г.) // Бюллетень международных договоров, ноябрь 2003 г., № 11.
12. Третьяков А.С. Противодействие международному терроризму: военные аспекты сотрудничества государств-участников СНГ // Право и безопасность. № 4 (5). Декабрь, 2002.

СОЦИАЛЬНЫЙ ФРАКТАЛ КАК ФАКТОР МИНИМИЗАЦИИ УРОВНЯ НЕОПРЕДЕЛЕННОСТИ В СОЦИУМЕ

В.В. Скопцов

Научный руководитель – д.соц.н., профессор Л.Я. Орлова

Современные гуманитарные науки испытывают острую потребность в достоверной оценке качества жизни человека. Наиболее общим состоянием, с которым подавляющее число людей отождествляют свою удовлетворенность жизнью, является состояние счастья. Судя по всему, интервал «счастье – несчастье» включает в себя подавляющее число психофизиологичес

ких состояний человека. Интуитивно ясно, и собственный опыт это подтверждает, что в значительной степени счастье человека носит случайный характер. На бытовом уровне случайность указанного состояния обозначается словом «удача». Как известно, человеческое существование складывается из последовательностей «замысел – результат». Несовпадение замысла с результатом ассоциируется в общем случае с неудачей – неудачным исходом очередного статистического испытания. Неудачник в подавляющем числе человеческих языков есть синоним понятия несчастливый человек.

Очевидно, что развитая социальная инфраструктура предоставляет человеку возможность реализовать свои замыслы в условиях меньшей неопределенности, чем в условиях, например, необитаемого острова, где позитивное решение вопроса еды, ночлега и безопасности может стать серьезной и даже неразрешимой проблемой. Следовательно, по мере развития цивилизации уровень неопределенности в жизни человека в долгосрочной перспективе объективно снижался.

Так или иначе, становится ясно, почему не только в аспекте существования человека как вида, но и на бытовом уровне проблема уровня неопределенности в реализации и замысла цивилизации и замыслов членов социума играет чрезвычайно важную роль.

Целесообразность так называемого фрактального подхода основана на том, что фракталы позволяют компактно «сжимать» информацию, строить простые модели, что, в свою очередь, позволяет более эффективно строить прогнозы, описывать самоорганизующиеся процессы. В круговороте циклов развития наций у Дж. Вико, а также в концепциях развития локальных цивилизаций у Н. Данилевского, О. Шпенглера, П. Сорокина, А. Тойнби и др. уже усматривается наличие фрактальных закономерностей. Существует уверенность, что было бы целесообразно проанализировать наследие этих мыслителей через призму социальной фрактальности, ибо обобщение результатов такого анализа могло бы послужить основой разработки фрактальной концепции общества и его развития в целом и, в частности, современной цивилизации¹.

Эволюцию органического мира нередко определяют как последовательное освобождение живых организмов из-под власти среды. Указанная эволюция сопровождается усложнением организмов. Аналогичная тенденция имеет место и в человеческом социуме, где последовательное усложнение организации социума позволяет социуму в целом, а также его членам все более успешно противостоять натиску неопределенности.

Несмотря на то, что человек является вершиной эволюции, приходится признать, что человечество в настоящий момент не в достаточной мере понимает законы эволюции мироздания в целом. Точно также мы не понимаем и нашего собственного назначения в природе. Человек обладает еще недостаточно развитым мозгом, или, по крайней мере, не имеет достаточно знаний, чтобы понять принципы самоорганизации социума. Непонимание этих принципов понуждает его воспринимать процессы в социуме как хаос, а хаос, в свою очередь воспринимается как стресс, вызывающий массовое неадекватное поведение членов социума.

Тот факт, что геном человека не содержит ни одного гена, который бы не встречался у других живых организмов, населяющих планету Земля, неопровержимо подтверждает, что наличие у нас так называемого разума не обусловлено биологически.

Обретение человеком того, что мы называем разумом, очевидным образом должно считаться результатом культурного процесса, таким же его результатом, как речь, прямохождение и многое другое.

В настоящем исследовании предлагается рассмотреть ситуацию, при которой человеческий вид по мере своего развития все в большей и большей степени насыщал свое социальное поведение отношениями обмена, причем не только в эмоциональной и физиологической формах, как это принято у других животных, но и отношениями имущественного обмена. Очевидно, что при этом естественным образом могла развиться специализация или, как ее чаще называют, разделение труда.

В дальнейшем именно на основе социального обмена и разделения труда и возникла человеческая цивилизация, сформировавшая ныне существующий вид животных, который принято называть *homo sapiens*.

Учитывая вышеизложенное, можно утверждать, что с не меньшим основанием наш вид мог бы именоваться *homo commutates* (человек обменивающийся), что прямо бы указывало на его генезис и на экономический источник его существования.

Действительно, обмены даже среди приматов редки, существуя в зачаточной форме только у шимпанзе и, возможно, у некоторых других обезьян. Однако у человека это одна из самых сильных социальных черт, достигающая уровня, сопоставимого с обменом пищей у термитов и муравьев. В результате только человек создал экономику, и в основе ее по-прежнему лежат отношения обмена и специализация. Возникает только один вопрос: почему память об этом у современного человека оказалась утрачена до такой степени, что принимаемые им экономические решения не способствуют ни специализации, ни повышению объема и количества отношений обмена.

Для удобства рассуждений прибегнем к форме, наиболее привычной нашему виду. Будем говорить о «замысле» цивилизации, как если бы некто на заре ее зарождения указал на зарождающиеся социальные обмены и специализацию как на краеугольные камни инструмента сознательного созидания будущего, именуемого нами теперь как Человеческая Цивилизация.

Говоря о замысле цивилизации, следует признать, что указанный объект применительно к данной работе – не более чем модель, удобство которой обусловлено возможностью ретроспективно оценить структуру системы принципов и условий, которая позволила человеческой популяции создать человеческую цивилизацию. И, что самое важное, такого рода подход в перспективе позволил бы сконструировать такую систему принципов и условий, которая позволила бы человечеству с большей уверенностью и, возможно, с большим пониманием осуществлять свою бесконечную битву с неопределенностью.

Один из авторов теории социального обмена, американский социолог Джордж Каспар Хоманс (род. в 1910 г.), полагал, что само существование людей есть не что иное, как постоянный обмен имущественными, интеллектуальными, информационными, эмоциональными ценностями. В сочетании с феноменом специализации (разделения труда) социальный обмен образует уникальный механизм, производящий по выражению А.Смита так называемое «богатство народов».

Очевидно, что в процессе эволюции Вселенной и человеческой цивилизации в частности имело и имеет место некоторое равновесие между хаосом и порядком. Преобладание хаоса уничтожало социум, а зарегулированность в социуме вызывала снижение внутренней энергии и постепенное его угасание.

Эта симметрия предполагает, что все процессы, ведущие к росту хаоса, уравновешены процессами, приводящими к росту порядка (конкурирующие процессы) при статистическом равновесии системы в целом.

В настоящее время к процессам, приводящим к росту упорядоченности, относят не только процессы, управляемые живой природой и человеком в том числе, но и так называемые фрактальные процессы², когда упорядоченность в неживой природе возникает из хаоса без каких-либо видимых внутренних побуждений. Особенно интересен тот факт, что фрактальные процессы довольно просто моделируются на компьютере, а их результаты в виде причудливых многоцветных, а подчас и объемных изображений даже выставляются в художественных галереях. Наиболее зримым и наглядным примером фрактального процесса является пламя, где частички, обладающие способностью к сгоранию, образуют цепочки последовательных вспышек, в сумме своей формирующих причудливо изменяющиеся картины огня.

Современная литература не дает нам достаточно функционального для наших целей определения фрактала. Многие авторы трактуют его чересчур широко, причем чаще всего имеют место попытки охарактеризовать результаты «работы» фрактального процесса – «мертвое изображение» – а не сам фрактальный процесс. Нас же интересует сам фрактальный процесс. Применительно к социуму, состоящему из человеческих особей, имеющих некий социальный и юридический статус, будем понимать под фракталом цепочки, возникающие между членами социума в процессе человеческих отношений (имущественных, информационных, интеллектуальных, эмоциональных физиологических и иных). Таковые отношения во времени и пространстве образуются между двумя и более членами социума в процессе так называемой человеческой деятельности. В зависимости от сложности и мощности этой деятельности размеры соответствующих «цепочек» или, в более общем случае, «сетей», могут достигать десятков, тысяч и более элементов. Очевидно, что во времени такие структуры могут быть относительно изменчивы как по структуре, так и по количеству элементов в гиперпространстве вида деятельности.

Описанные выше сети отчасти напоминают химическую реакцию на уровне молекул и атомов. Другая аналогия приводит нас к понятию «цепная реакция» и т.д.

В процессе выполнения конкретного вида человеческой деятельности, скажем в процессе производства хлеба, имеют место как относительно устойчивые отношения, так и отношения краткосрочные или даже случайные. Это, в принципе характерно для подавляющего большинства видов человеческой деятельности.

Таким образом, за модель зон так называемой автономной (от государства) активности мы принимаем *социальный фрактал* – социальный процесс, осуществляемый подмножеством членов социума, объединенных (осознанно или неосознанно) социальными отношениями и выполняющих (осознанно или неосознанно) вид или подмножество видов человеческой деятельности.

Возвращаясь к проблеме снижения уровня неопределенности в социуме, будем считать, что такие детерминаторы, как организации, начиная от созданных государством и кончая основной ячейкой общества – семьей, не вполне устраняют неприемлемые уровни неопределенности в социуме, поэтому социум спонтанно порождает социальные фракталы, позволяющие выжить социуму в целом.

Известно, что для обеспечения возможности построения фрактального образа его элементы должны удовлетворять определенным, в том числе вероятностным свойствам. Анализируя статику и динамику человеческого социума, представляется возможным, что целый ряд объектов, вмонтированных в социум, и государство возникли как самоорганизовавшиеся структуры по типу фрактала. В то же время, вследствие вполне детерминированных процессов, возникали структуры, созданные по типу организмов – организации. Налицо два независимых процесса, направленных на установление по-

рядка в социуме в противовес процессам дезинтеграции: один – управляемый сознанием людей (организации), другой – происходящий помимо воли членов социума, но при их участии (социальные фракталы).

В отношении создания организаций у человечества имеется достаточно большой опыт, однако полностью исключить процесс фрактальной самоорганизации из ткани социума представляется неразумным. Можно предположить, что самоорганизация в социуме происходит вследствие невозможности компенсации влияния элементов хаоса на ткань социума только за счет функций организаций – сознательных действий руководящих социумом властных структур.

Итак, приняв за основу необходимость присутствия в социуме фрактальных процессов самоорганизации, следует выяснить, какими свойствами должны обладать субъекты, из которых могут самоорганизовываться фрактальные структуры социума.

Говоря о социуме, о его способности к самоорганизации и к саморазвитию, мы неизбежно приходим к необходимости выявления внутренних побуждений указанных процессов. Очевидно, что для всех членов социума чрезвычайно актуальна проблема поддержания собственной жизнедеятельности. Времена охоты и собирательства для нашего социума остались далеко позади. Физическим источником «богатства народов» стало разделение труда. Моментом возникновения «богатства народов» является момент реализации товаров и услуг, производимых членами социума. Однако к осознанию того факта, что истинным внутренним побуждением к самоорганизации и саморазвитию является необходимость реализации товаров и услуг, которая, собственно, и создает «богатство народов», по определению А. Смита, человечество пришло только в конце прошлого века. Заметим, что даже после этого открытия правительства планеты продолжают руководствоваться в первую очередь фактами наличия или отсутствия полезных ископаемых на соответствующей территории, а не степенью развития условий для максимизации отношений обмена между членами социума. Другими словами, даже после устранения неопределенности в источнике побудительной мотивации в действиях членов социума продолжает существовать хаотичность в принятии решений государственной властью. Принимаются нормативные акты, иные решения, объективно снижающие деловую активность в социуме в угоду личным амбициям власть предержащих и просто влиятельных лиц. Установленный факт, имеющий юридическое значение, не нашел своего нормативного закрепления в системе права и системе законодательства. Его формулирование в виде принципа «свобода предпринимательства» не является достаточно конструктивным применительно к праву и экономике.

Усложняя свою организацию, социум заимствовал и заимствует способы противостояния неопределенности у живой природы. Сюда относится, например, способ накопления ресурсов, начиная от накопления запасов пищи и кончая применением коллективных действий. Однако в процессе эволюции, человек, как бы подтверждая свою исключительную способность к самоорганизации, использует и такой способ борьбы с неопределенностью, как страхование рисков. Институт страхования возник первоначально как игра завсегдатаев портовых кабаков, где делались ставки по поводу успешного возвращения отплывающих судов. По мере возрастания ставок выяснилось, что самим участникам путешествия есть смысл тоже поучаствовать в такой игре, правда, делая ставку на гибель собственного судна. Действительно, в случае гибели судна родственники погибшего могли получить причитающийся ему выигрыш. Такого рода защиты от неопределенности в живой природе не существует.

Учитывая то, что страхование рисков является относительно новым способом борьбы социума с неопределенностью, исследуем его и попробуем получить нетривиальные выводы.

Очевидно, что в условиях, когда страховые риски велики, значительная доля совокупного общественного продукта в конечном счете идет на соответствующее возме-

шение ущерба, неизбежно снижая эффективность общественного производства. Дело снижения уровня этих рисков отнюдь не является «делом рук самих утопающих». И в то же время никакие страховые компании, которые объективно заинтересованы в сохранении достаточно значимых (реальных) уровней страхуемых рисков, не станут способствовать их снижению, но важнее всего то, что у них для этого нет абсолютно никаких инструментов.

Учитывая тот факт, что именно государству наносит ущерб снижение совокупного общественного продукта и только государство имеет возможность, используя организационно-технические мероприятия, значимо влиять на уровни страхуемых рисков, вся выручка от страховых взносов, за вычетом ее части, предназначенной для возмещения вреда от наступивших страховых случаев и отчислений в фонды перестрахования, должна направляться на организационно-технические мероприятия с целью уменьшения уровней страхуемых рисков.

Логично предположить, что такой объект социума, как гражданское общество также является социальным фракталом и в принципе не может быть создан на основе алгоритмов построения организаций. Наиболее естественным объектом для создания структур, функционирующих в зонах автономной активности социума, является человек. Однако современный юридический статус человека в российском государстве настолько усечен, что не позволяет ему эффективно (с наименьшими затратами) вступать в отношения обмена (имущественного, информационного, интеллектуального, эмоционального, физиологического и т.п.) с другими членами социума. Подтверждением указанной неэффективности служит демографическая ситуация, существующая в России с начала 90-х годов прошлого века. Гражданин РФ, например, лишен права без специальной государственной регистрации заниматься предпринимательством, лишен права без государственной регистрации находиться в местах, отличных от его «места жительства», и т.д. Но самое главное, не решен вопрос о его имущественной безопасности в процессе ведения им предпринимательской деятельности, поскольку риск утраты им имущества в процессе предпринимательства не страхуется. Кредитные и страховые истории членов социума не ведутся, и, следовательно, оценить риск совершения сделки не представляется возможным.

Только создав человека обменивающегося, обладающего параметрами, характеризующими его страховую и кредитную историю, собственность, состояние здоровья и т.п., можно рассчитывать на то, что данный элемент вкупе с другими подобными элементами начнет создавать самоорганизующуюся динамическую социальную фрактальную структуру, где и будет существовать зона автономной (от государства) активности. И именно эти «человеки обменивающиеся», в значительной степени не зависящие от влияния властных структур, но целиком и полностью находящиеся под их защитой и покровительством, смогут образовать еще как минимум одну зону автономной активности – социальный фрактал под названием Гражданское Общество.

¹ Князева Е.Н. Курдюмов С.П. Законы эволюции и самоорганизации сложных систем. М., 1994. С. 45.

² Фракталы в физике: VI Международный симпозиум. М.: Мир, 1998. С. 28, 9, 10.

ПОЛИТИКА И ИНТЕРНЕТ: ПОЛИТИЧЕСКАЯ КОММУНИКАЦИЯ В УСЛОВИЯХ РАЗВИТИЯ СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Е.Ю. Косенко, Е.А. Харламова

Научный руководитель – д.полит.н., профессор И.Б. Орлов

Существенной особенностью современных процессов глобализации является экспоненциальный рост масштабов применения новых информационных технологий в жизни общества. Особая роль в этом процессе принадлежит глобальной сети Интернет, которая, являясь специфическим интерактивным средством массовой коммуникации, уже активно используется в политической жизни общества.

Достижения в области новых технологий, на которых основывается формирование информационного общества, во многом изменили характер политической коммуникации. Важными факторами, влияющими на политический процесс, является постепенная демассификация масс-медиа и снижение издержек информационного обмена, что позволяет гражданам и их объединениям более активно участвовать в подготовке и принятии политических решений. Таким образом, обращение к теме «Политика и Интернет» объясняется настоятельной необходимостью начала проведения междисциплинарных исследований особенностей политической коммуникации в условиях развития технологий информационного общества в России.

Целью исследования является выявление особенностей функционирования политической коммуникации в условиях формирующегося информационного общества.

Современные СМИ приобретают новые специфические функции, связанные с появлением новых электронных и сетевых технологий, в том числе Интернета. Для подробного анализа этого процесса необходимо изучить тенденции развития информационного общества и новую роль средств массовой информации в условиях развития современных информационных технологий, особенно Интернета, а также проанализировать основные направления воздействия Интернета на политическую жизнь общества.

При этом необходимо заметить, что важным фактором, определившим тенденции развития информационного общества на ближайшие десятилетия, было создание в 1990/91-х гг. технологии World Wide Web (WWW или «Всемирная паутина»). Катализатором развития WWW и всего Интернета явилось изобретение графического интерфейса для просмотра HTML-документов – браузера «Mosaic» в 1993 г. С этого времени начался экспоненциальный рост Интернета, который постепенно стал ассоциироваться со «Всемирной паутиной» как совокупностью Web-серверов.

В этой связи необходимо подчеркнуть, что в период 1991/95 гг. произошли также существенные структурные изменения в американской экономике:

- расходы на информационные технологии превысили затраты на приобретение производственных технологий и основных фондов;
- большая часть добавленной стоимости всего промышленного производства было создано при помощи информационных технологий;
- обеспечен резкий рост продаж персональных компьютеров.

Важным аспектом развития информационного общества является решение проблемы раскрытия информации о деятельности властных структур.

С начала 1990-х гг. в системе предоставления информации об основных направлениях деятельности структур власти и управления все более заметное место занимает Интернет.

Говоря о воздействии Интернета на политические процессы в современном обществе, необходимо учитывать, что раскрытие информации через Интернет, помимо позитивных общественных тенденций, имеет и «побочные» негативные социальные последствия.

В США, в отличие от Европы, приняты кодексы поведения в сети добровольного характера, однако расширение использования информационных технологий Интернета в повседневной жизни населения заставляет правительственные круги по-новому взглянуть на этот вопрос.

В России проблемы раскрытия информации и защиты персональной информации в Интернете начинают активно обсуждаться на специализированных конференциях и семинарах с 1998 г.

Параллельно с внедрением Интернета в жизнь современного общества появляются новые политические технологии. Многие исследователи относят Интернет к новым электронным средствам массовой информации (СМИ), однако интернет-технологии существенно отличаются от традиционных и электронных СМИ, прежде всего своей интерактивностью и избирательностью.

В условиях развития информационного общества происходит постепенное снижение издержек на получение и передачу информации, что создает предпосылки для развития двух тенденций. Во-первых, постепенно увеличивается группа людей, имеющих возможность принимать участие в выработке и принятии политически значимых решений; во-вторых, создаются технологические предпосылки для постепенного уменьшения неравенства политических возможностей граждан.

Одним из существенных институтов политической системы являются выборы, и вполне естественно, что по мере внедрения сетевых информационных технологий в повседневную жизнь общества начали проводиться эксперименты по переносу участков для голосования в Интернет. Интернет с конца 1990-х гг. начинает активно использоваться в американских выборах в качестве нового интерактивного масс-медиа.

Большинство Web-сайтов кандидатов имело вид электронных публикаций, содержащих биографическую информацию и позиции кандидатов. На некоторых из них публиковались копии предвыборных выступлений и оперативная информация о ходе предвыборной кампании. Характерно, что тексты своих выступлений перед избирателями публиковали только 20% кандидатов, а регулярно обновляемые новости предвыборной кампании можно было прочитать на 49% сайтов. Однако следует отметить, что по оценкам обозревателей выборы 1998 года вошли в историю США как первые выборы, в которых Интернет – новое средство массовой информации – сыграло одну из главных ролей.

Впервые об Интернете как реальном инструменте политической борьбы в России стали говорить в декабре 1998 г. после публикации в сети так называемого «Когтя» – коллекции распечаток телефонных разговоров и пейджерных сообщений известных политиков (Т. Дьяченко, Б. Березовского и др.). Таким образом, в России начала реализовываться схема распространения компрометирующих материалов: «ИНТЕРНЕТ – СМИ – ОБЩЕСТВО». С начала 1999 года и по сей день отечественные газеты пестрят заголовками типа «Русскую сеть пучит от компромата»^[1], «Компромат в Интернете»^[2], «Грязь XXI века»^[3], «Сетевой компромат. Виртуальные провокаторы упражняются в мастерстве»^[4], «Из сети продолжают расти «Когти»»^[5], ««Коготь» дотянулся до кепки Лужкова»^[6] и т.п. Относительно немногочисленные случаи публикации в Интернете действительно компрометирующей информации раздувались прессой до происшествий национального масштаба. Журналисты внимательно следили за очередным «когтем» и делились с читателями предложениями, кому бы это могло быть выгодно, вплоть до таких, к примеру, странных предположений: «За сайтом «Коготь-2» могут стоять «Правое дело», «Сибирский алюминий» и Фонд эффективной политики»^[7]. Пресса своим гипертрофированным вниманием к теме «компромат в Интернете» сформировала в общественном сознании миф об Интернете как «виртуальном сливном бачке»^[8], особенно распространенный в 1999 году, когда новые «когти» на волне медийного интереса к теме появлялись чуть ли не еженедельно. Необходимо заметить, что абсолютное

большинство сайтов, раскручиваемых как «компроматные», не содержали никакой иной информации, кроме тематических подборок публикаций СМИ с пристрастными комментариями создателей ресурса. Ровно по этой схеме в сентябре 1999 года был раздут скандал с сайтом www.lujkov.ru – двойником официального сайта Лужкова. Тысячи пользователей побывали на этом ресурсе в поисках компромата на московского мэра. Вместе с тем псевдосайт Лужкова, раскручиваемый создателями как содержащий запрещенную информацию, содержал лишь дайджесты прессы, расшифровки передач Сергея Доренко и относительно безобидные карикатуры на мэра Москвы. Имеются и позитивные примеры – первой корректной комплексной политической кампанией в российском Интернете можно считать проект С. Кириенко «Московская альтернатива» (1999 г.).

Следует обратить внимание на тот факт, что на сегодняшний день Интернет, формально не являясь СМИ, начинает использоваться в политических технологиях не только как средство быстрой связи между регионами для мониторинга ситуации, но и в качестве средства борьбы с конкурентами через «слив информации», которая не может быть опубликована в печатных и традиционных электронных СМИ как средство агитации среди молодежи. Это обстоятельство особенно настораживает в связи с тем, что в 1998/99-х гг. обозначился процесс взаимопроникновения масс-медиа (традиционные СМИ и интернет-медиа), что говорит о постепенном включении Интернета в политическую жизнь российского общества.

Таким образом, внедрение новых технологий информационного общества начинает существенным образом воздействовать на социально-экономические и политические сферы общества в двух направлениях:

- поддержка равновесия в общественных структурах, и как следствие – тенденции политической стабилизации;
- разрушение общественных структур, и как следствие – возможная дестабилизация общества.

Еще одной составляющей процесса взаимодействия политики и Интернет является «аудитория» (пользователи Интернета). Естественно, что политическая активность в российском Интернете направлена на особую аудиторию. Вопросы статистики пользователей Интернета, распределение аудитории по городам и регионам России, анализ ее социально-демографического состава (образование, пол, возраст и т.п.) начали активно обсуждаться на семинарах, конференциях и в периодической печати в 1998/99-х гг. Постепенно сформировались несколько исследовательских групп, реализующих свои исследовательские программы. Первой компанией, представившей данные о численности российской аудитории Интернета, основанные на результатах репрезентативных социологических исследований, была КОМКОН.

Для более точной оценки аудитории Интернета в Москве и Санкт-Петербурге были привлечены результаты опросов взрослого населения российских регионов, которые с 1999 года проводятся Агентством региональных политических исследований (АРПИ) и Национальным институтом социально-психологических исследований (НИСПИ).

Интернет начинает всерьез рассматриваться как средство продвижения товаров и услуг, а интернет-аудитория становится весьма привлекательной с точки зрения маркетинга целевой группой. Дискуссия о методах измерения пользователей Интернета и основных понятиях и показателях позволила откорректировать исследовательские программы мониторинга аудитории российского Интернета.

Особую ценность для изучения Интернета как среды политической коммуникации представляет исследование «Интернет в России / Россия в Интернете», проведенное в конце 2000 года Фондом «Общественное мнение»^[9]. По социально-демографическим характеристикам результаты ФОМ не сильно отличаются от данных

других социологических компаний: аудитория российского Интернета – это специфическая группа населения, которая характеризуется следующими признаками: проживание в Москве, Санкт-Петербурге и крупных городах; наличие высшего (неоконченного высшего) образования; средний возраст около 35 лет; доходы средние или выше средних; оптимизм или адаптированный пессимизм (по отношению к современным трансформационным процессам). Данные ФОМ подтверждают гипотезу о том, что Интернет в первую очередь осваивается «лидерами мнений» – людьми, которые занимают активную жизненную позицию, регулярно получают информацию из СМИ и транслируют ее в процессе общения тем социальным группам, в которых имеют влияние.

По своим основным параметрам аудитория российского Интернета приблизительно соответствует социальным характеристикам так называемого «среднего класса», что позволяет сформулировать осторожный вывод о возрастании роли современных сетевых информационных технологий в политическом процессе.

Важной особенностью развития российского сегмента всемирной сети стало то, что, в отличие от Европы и Америки, в России интернет-технологии в значительно большей степени были востребованы в политической и медиа-среде, нежели в сфере бизнеса. Определяющим фактором развития российского Интернета в конце 1990-х–2001 гг. стала активность крупных медиа- и политических структур, а не коммерческих организаций. В эти годы в России происходил процесс освоения различными политическими субъектами (политическими партиями и движениями, институтами государственной власти, медиа-структурами и др.) Интернета в качестве средства политической коммуникации. Будучи порождением американской военно-инженерной мысли, развивающийся во всем мире как коммерческая информационная среда, Интернет в России был удачно включен в сложившуюся к середине 1990-х гг. медиа-политическую систему. Специфические возможности Интернета были использованы различными субъектами российской политической жизни в информационных войнах, во время парламентских и президентских выборов, в ходе осуществления международной информационной политики.

Масс-медиа сыграли особую роль в российской политике конца XX века. В эпоху перестройки СМИ одновременно были и инструментом преобразований, и зеркальным отражением состояния общества. В 1990-е годы российские СМИ фактически стали «четвертой властью» – институтом, претендующим на монопольное обладание функцией политической коммуникации в стране.

Появление Интернета в России совпало по времени с ускорением процессов медиатизации российской политики. Во второй половине 1990-х гг. развитие сети в большой степени зависело от того, как строились отношения масс-медиа и субъектов российского политического процесса. Медиатизация политики предопределила востребованность Интернета в качестве инструмента политической коммуникации.

В этих условиях Интернет оказался крайне востребованным – и теми субъектами политического процесса, которые не имели больших возможностей в традиционных СМИ, и крупными политическими игроками, использовавшими особенности интернет-среды в тех случаях, когда Интернет оказывался наиболее адекватным коммуникативной задаче средством. Перечислим основные применявшиеся свойства глобальной сети: легкость, быстрота и дешевизна публикации; возможность доступа к опубликованной информации из любой точки мира; интерактивность; неопределенный статус сетевой публикации (возможность публиковать сведения, не подходящие по формату или содержанию для традиционных СМИ). В ходе информационных операций в Интернете модель двухступенчатой коммуникации, разработанная американским исследователем П.Лазарсфельдом^[10] в конце 1940-х годов, была адаптирована к российским сетевым технологиям: Интернет-проект (Интернет-событие) в качестве импульса; отклик (шум) в традиционных СМИ; привлечение внимания целевых аудиторий к информации.

В рамках поисковой системы «Яндекс» с мая 1999 года существует так называемый «Политический НИНИ-индекс»^[11] (www.yandex.ru/nini_polit.html), который позволяет посчитать количество поисковых запросов по фамилиям политических деятелей, а также вычлениить слова, которые пользователи при поиске набирают рядом с фамилией политика. Этот интересный проект позволяет проследить не только динамику пользовательского интереса к политическим событиям за последние три года, но и проанализировать его специфику. По набору ключевых слов и построению поисковых запросов видно, что информацию по общественно-политическим событиям ищут в сети профессионалы: журналисты, редакторы, сотрудники информационных структур.

Очевидна малочисленность пользователей сети относительно населения России в целом. Вне зависимости от того, каково сегодня точное число пользователей Интернета, оно по-прежнему несопоставимо с количеством жителей огромной страны (согласно данным Госкомстата, численность постоянного населения России по состоянию на 1 августа 2001 года составляла 144 миллиона 300 тысяч человек)^[12]. По оценкам экспертов, счет аудитории российского Интернета на десятки миллионов будет возможен лишь в 2010-х годах^[13].

Развитие российского Интернета в качестве средства политической коммуникации во многом было задано специфическими чертами сетевой аудитории. Помимо социально-демографических особенностей (среди которых стоит также упомянуть большое количество пользователей Интернета в столицах и крупных городах при их практически полном отсутствии в сельской местности), такой вывод позволяет сделать активное освоение сети «лидерами мнений» – влиятельными коммуникаторами и трансляторами информации, в том числе журналистами.

Компьютерная сеть раскрепощает человека, делает его свободным, на деле создает глобальное информационное пространство^[14]. При этом целевой аудиторией обращения было, по словам Чубайса, «новое поколение, говорящее на языке компьютеров», которое «выходит на авансцену политической и экономической жизни страны» и является «надеждой и будущим России»^[14].

Таким образом, в течение 1990-х гг. Интернет в России оказался включенным в качестве еще одного коммуникативного инструмента в существующую систему обмена смыслами между субъектами политического процесса. Основными факторами развития российской сети в качестве средства политической коммуникации стали неразвитость инфраструктуры для формирования коммерческого сектора Интернета, медиатизация российской политики, приведшая к возникновению феномена включенных в медиаполитическую систему сетевых СМИ и активному использованию интернет-технологий в ходе информационно-пропагандистских кампаний, а также специфика российской сетевой аудитории и мифологизация интернет-пространства в общественном сознании.

Литература

[1] Русскую сеть пучит от компромата // Новые известия. 1999. № 30. 20 февр.

[2] Скворцов С. Компромат в Интернете // Московская правда. 1999. № 37. 25 февр.

[3] Баюн Е. Грязь XXI века // Московский комсомолец. 2000. № 227. 9 окт.

[4] Ланцман М. Сетевой компромат // Время. 1999. № 32. 24 февр.

[5] Корецкий А. Из сети продолжают расти «Когти» // Сегодня. 1999. № 88. 23 апр.

[6] Гамов А. «Коготь» дотянулся до кепки Лужкова // Комсомольская правда. 1999. № 32. 23 февр.

[7] Скворцов С. За сайтом «Коготь-2» могут стоять «Правое дело», «Сибирский алюминий», и Фонд эффективной политики // Московская правда. 1999. № 37. 25 февр.

[8] Шаров А. Виртуальный сливной бачок // Парламентская газета. 1999. № 196.

[9] «Интернет в России / Россия в Интернете», исследование доступно на сайте ФОМ (www.fom.ru), а также на специальном сайте проекта (internet.strana.ru).

^[10] Лазарсфельд П., Мертон Р. Массовая коммуникация, массовые вкусы и организованное социальное действие // Назаров М.М. Массовая коммуникация в современном мире: методология анализа и практика исследований. М., 1999. С.138-149.

^[11] НИИ – «непостоянство интересов населения Интернета».

^[12] Численность населения России продолжает сокращаться // Сообщение ТАСС. 2001. 24 сент.

^[13] В Минсвязи РФ прогнозируют, что в России к 2010 г будет более 26 млн. пользователей Internet // Сообщение Интерфакс. 2001. 6 сент.

^[14] Нилов М., Черненко Л. Как сделать прозрачным «кремлевское окно» // Российские вести. 1997. 14 окт.

К ПРОБЛЕМЕ ВЗАИМНОГО ПРОНИКНОВЕНИЯ ИНФОРМАЦИОННЫХ И ПОЛИТИЧЕСКИХ ТЕХНОЛОГИЙ

М.В. Бородин, В.А. Ермаков

Научный руководитель – д.полит.н., профессор И.Б. Орлов

Целью данной работы является рассмотрение взаимного влияния политических и информационных технологий. При подготовке сообщения использованы материалы распространенных Интернет-изданий, обновленных не ранее 1 января 2005 г.

Введение

Технология – это комплекс научных и инженерных знаний, реализованных в приемах труда, наборах материальных, технических, энергетических, трудовых факторов производства, способах их соединения для создания продукта или услуги, отвечающих определенным требованиям. Поэтому технология неразрывно связана с механизацией производственного или непроизводственного, прежде всего управленческого процесса. Управленческие технологии основываются на применении компьютеров и телекоммуникационной техники.

Согласно определению, принятому ЮНЕСКО, информационная технология – это комплекс взаимосвязанных научных, технологических, инженерных дисциплин, изучающих методы эффективной организации труда людей, занятых обработкой и хранением информации, вычислительную технику и методы организации и взаимодействия с людьми и производственным оборудованием, их практические приложения, а также связанные со всем этим социальные, экономические и культурные проблемы. Сами информационные технологии требуют сложной подготовки, больших первоначальных затрат и наукоемкой техники. Их введение должно начинаться с создания математического обеспечения, формирования информационных потоков в системах подготовки специалистов.

ИТ, влияющие на политику

Политики применяют в своей повседневной деятельности информационные технологии, такие как СМИ и Интернет. К СМИ относятся периодические и непериодические печатные издания, телевидение, радио, слухи. С развитием СМИ появились такие профессии, как специалист по связям с общественностью, имиджмейкер. PR не мог возникнуть и существовать при отсутствии рыночной экономики. Первым зародившимся PR направлением является спонсоринг – набор технологий, с помощью которых представители политики пытаются убедить общество в своей полезности. С возникновением PR технологий выделились негативные PR технологии, которые действуют через СМИ. У таких технологий всегда есть заказчик, причастность которого обычно недоказуема, и исполнитель (журналист), имя которого изменено или отсутствует. В России применяются больше печатная продукция, чем телевидение. В США, напротив, в рамках одного из социологических опросов, где выяснялось, откуда избиратели черпают информацию о политиках, были получены следующие ответы:

- из теледебатов (60%);
- из ток-шоу (45%);
- по результатам партийных съездов (25%).

Если в 1988 году теледебаты считали важными 48% американцев, то в 1992 году их число выросло до 70%. Все это говорит о глобальной роли, которую стало играть телевидение в ходе политических кампаний.

Всё большую роль в настоящее время играет Интернет. Это обусловлено тем, что все больше и больше людей начинают использовать всемирную паутину. Например, 8%

жителей России являются активными пользователями Интернет. Но в избирательных кампаниях Интернет-технологии пока бесперспективны, несмотря на самую совершенную разработку, поскольку самая голосующая часть населения – пенсионеры, наименее обеспеченная категория – чаще всего не только не имеют доступа к сети, но и с трудом представляют, что такое Интернет.

Общенациональные исследования, осуществленные в 2000 и 2001 годах Калифорнийским университетом в Лос-Анджелесе, показывают, что, хотя около половины американцев, пользующихся Интернетом, считают что Сеть «помогает им лучше понимать политику», только 23% в 2001 г. (против 29% в 2000 г.) полагало, что благодаря Интернету они «могут обрести большую политическую власть»; только 21% (против 24% в 2000 г.) полагал, что Интернет «усиливает их голос в отношении действий правительства»; только 24% (против 28% в 2000 г.) считал, что Интернет «может заставить политиков внимательнее прислушаться к мнениям граждан».

Тем не менее, интерес представляют следующие категории пользователей сети: относительно обеспеченные обыватели, имеющие собственный ПК и доступ в Интернет: предприниматели, управленцы, чиновники, специалисты самого разного профиля; в целом – люди, несколько более информированные и «продвинутые», чем самые «низшие слои». Также существуют более крупные категории, использующие «халявный» Интернет на рабочем месте: государственные и муниципальные служащие, работники многих бюджетных организаций, специалисты коммерческих структур; студенты, не имеющие собственного доступа в Интернет, но получающие его в вузах либо посещающие всевозможные Интернет-кафе, Интернет-центры.

Самая распространенная интернет-технология – спам – массовая электронная рассылка чего-то, о чем адресаты совершенно не просят, проводимая независимо от их согласия. Существует мнение, что такие Интернет-рассылки являются одним из эффективнейших средств рекламы, но реальность опровергает это. Напротив, зачастую спам выполняет антирекламную функцию, стимулируя неприятие и раздражение в адрес политика. Содержание информации в спаме может быть практически любым, так как главную роль здесь играют не содержательное наполнение, а форма воздействия. Зачастую адреса e-mail берутся из какой-нибудь базы данных. Другое направление – использование так называемого персонифицированного спама, адресованного конкретному объекту. Задача – провоцирование раздражение и создание «информационного» шума, а также эксплуатация времени объекта. Наконец, существует повсеместная рассылка различного рода «компромагов». Многие считают, что сегодня компромат на мало-мальски значимую фигуру можно найти даже с помощью наиболее популярных поисковых систем. И в этом есть доля правды. Отсутствие ответственности (анонимность) – одна из слишком очевидных особенностей Интернета. Еще одно из направлений политических Интернет-технологий – использование онлайн-СМИ в качестве источников для распространения Интернет-слухов. Конечно, эффективность Интернет-слухов значительно ниже по сравнению с привычным неформальным общением, однако и такой возможностью иногда приходится довольствоваться.

В настоящее время каждый политик имеет свой сайт. По мнению специалистов в PR-технологиях, само наличие сайта – основная проблема, так как при отсутствии сайта «достать» политика было бы труднее, но его существование сразу предоставляет широкое поле для деятельности. Все начнется с отзывов в гостевой книге, затем обсуждение на форуме. Форум станет одним из самых посещаемых разделов. Далее на доске объявлений возникают предложения о знакомстве от лиц нетрадиционной сексуальной ориентации и даже объявления от конкурентов. Затем – «невинные» разговоры в чате с использованием блатного жаргона, а также очень «яркие» откровения низших членов штабов на своих руководителей, личные отношения между ними и отдельных сторонах деятельности. Для увеличения рейтинга политиков в отдельных Интернет-изданиях

можно разместить несколько статей за их подписью по каким-нибудь особо проблемным или узкоспециальным вопросам. Также полезным может стать наличие его высказываний, преимущественно в жестких и неоднозначных формах, на самых посещаемых сайтах. Также не помешает поставить свое имя и под самыми разными открытыми соглашениями в «поддержку инициатив» и «против произвола». Затем, для усиления личностных характеристик, относящихся к наличию хобби, следует зарегистрировать политика и на сайтах нескольких специфических видов музыки, виртуальных клубов виртуальных любителей спорта, азартных игр и еще чего-нибудь.

Жертвой таких технологий является простой обыватель, который поглощает в себя, как губка, все, что слышит, видит и узнает.

Это влияние не односторонне. По мнению доктора исторических наук Григория Вайнштейна, «... с развитием Сети расширяется «область общественного» и, в частности, сфера доступной вниманию общественности политики и политической информации. При этом Интернет превращает человека из пассивного объекта информирования не только в активного, самостоятельного субъекта отбора плюрализованной массы информации, но и в полноправного участника самого процесса распространения информации, с трудом поддающегося внешнему контролю и регулированию». С доступностью Интернета каждый человек теоретически может повлиять на ход политических событий посредством привлечения внимания общественности к данной проблеме. Но «Интернет, как свидетельствует реальность, не выводит массовое политическое участие на новый, более высокий уровень. Он лишь дает некие новые средства проявления политической активности тем, кто и без информационных технологий принадлежит к числу политически активных».

Нельзя не отметить и тот факт, что информационные технологии начинают все шире использоваться узкими группами политических радикалов в террористических целях. Всемирная Сеть открывает возможности канализировать политическую активность гражданского общества в конструктивное русло, ориентировать ее на укрепление и развитие демократии. Но она же может быть поставлена на службу совершенно иным, экстремистским целям, и использована для возбуждения национальной, религиозной, социальной розни. Экстремисты разного толка (неонацисты, скинхеды, расисты, религиозные фанатики) увидели в становлении транснациональной сети Интернета чрезвычайно эффективное средство активизации своей деятельности, распространения своих взглядов и вовлечения в орбиту своего влияния новых сторонников. С момента появления в Интернете первого экстремистского сайта под названием «Штормовой фронт» (Stormfront), созданного в 1995 году одним из бывших лидеров Ку-Клукс-Клана Д. Блэком, Всемирная Паутина постоянно пополняется все новыми и новыми сайтами, пропагандирующими идеи насилия, расовой и религиозной нетерпимости.

По данным Центра Симона Визенталя, на октябрь 2001 года в Сети существовало более 1400 так называемых проблемных веб-сайтов. Судя по другим оценкам, их количество к концу 2001 года превысило 2000. По сути дела, киберпространство стало средой обитания некоего «электронного сообщества» экстремистов. Интернет оказался для членов экстремистских и террористических организаций весьма удобным средством преодоления географической разобщенности и поиска единомышленников.

Политический эффект IT

Западные политики давно уже включили в свой арсенал политических средств проекты в области IT и вообще покровительство высокотехнологичному и научному сектору экономики. Информатизация госструктур, приносящая зримый эффект, служит весомым фактором методической поддержки той или иной политической команды. Лидеры развитых стран на своих саммитах регулярно обсуждают вопросы использования информационных технологий и разрабатывают планы преодоления так называемого

«цифрового разрыва» между богатым Севером и бедным Югом. Особенно удачно образ высокотехнологичного лидера удавалось эксплуатировать Биллу Клинтону. В бытность его президентом США журналисты как-то подсчитали, что за одну неделю он двенадцать раз публично высказался о перспективах Интернета. А накануне последних президентских выборов в США Клинтон ставил в заслугу демократической партии достижения в области интернетизации американских школ.

В последнее время слова о том, что перспективы страны связаны с развитием науки и информационных технологий, вошли и в лексикон российских политиков. Причем тон задает сам президент – на ставших уже регулярными встречах с учеными и IT-сообществом. Если так и дальше пойдет, то в скором времени хорошим тоном будет считаться наличие у каждого заметного политика какой-нибудь информационной системы, созданной под его непосредственным руководством. Главное – добиться эффекта, проявив здравый хозяйственный подход, и тогда успешный IT-проект станет сильным козырем в политической карьере.

Примеры у всех перед глазами. Взять, скажем, нынешнего министра по налогам и сборам Геннадия Букаева. Его большая политическая карьера началась в 1995 году, когда он возглавлял государственную налоговую инспекцию по Башкирии. Первым в России он реализовал проект по созданию единой республиканской автоматизированной системы учета налогоплательщиков. Проект, безусловно, был удачным. Потребовавший на стадии реализации около 10 млн. рублей, он уже в первый год эксплуатации принес в несколько раз больше: в 1997 году по итогам предыдущего года башкирская налоговая инспекция доначислила около 47 млн. рублей подоходного налога, в следующем году – около 38 млн. рублей. И сегодня в Башкирии самый высокий уровень поступления налоговых деклараций: добровольно их подают более 90% налогоплательщиков, обязанных по закону это делать. Успех Геннадия Букаева не остался незамеченным: его перевели руководить московским управлением МНС, а через год, в мае 2000 года, он был назначен федеральным министром по налогам и сборам. Господин Букаев продолжает заниматься автоматизацией учета налогоплательщиков, но уже в масштабах всей страны, и возглавляемое им министерство числится в лидерах по уровню информатизации среди российских ведомств.

Регионы ставят на IT

Еще один вариант «наращивания политического веса» за счет IT опробовала администрация Ярославской области, где в прошлом году было принято решение об отмене для софтверных компаний региональной составляющей налога на прибыль. По словам заместителя губернатора Ярославской области Артура Сазонова, это был чисто политический ход: «Мы хотели показать, что наша администрация хочет развивать эту отрасль, возлагает на нее определенные надежды». Нельзя сказать, что крупные российские IT-фирмы воспользовались предложением и кинулись регистрироваться в Ярославле. «К притоку капитала, появлению новых компаний это, понятно, не привело. Я отдаю себе отчет в том, что налоги – не главная проблема софтверных компаний, хотя местные компании этой льготой пользуются», – говорит Артур Сазонов. Но администрация не сдаётся: ярославский губернатор подписал постановление о создании в области системы электронных закупок для государственных нужд. Сама ярославская администрация располагает всего двумя компьютерами, но руководство области считает, что этот шаг по достоинству оценят потенциальные инвесторы и федеральные ведомства.

На это же рассчитывает и администрация Новосибирской области. Но у новосибирского губернатора Виктора Толоконского преимущество: в его регионе система электронных торгов уже создана – первая в России. «Мы начали проводить электронные торги в начале марта, – сообщил господин Толоконский. – За месяц провели 10 торгов на сумму 350 млн. рублей и уже получаем цены в среднем на 12–15% ниже

среднерыночных. В абсолютном выражении это очень приличные деньги: в среднем мы планируем закупать товаров на 2–2,5 млрд. рублей в год, значит, сэкономим 300–350 млн. рублей. Программный продукт для создания системы мне стоит 1,5 млн. рублей, а содержание дополнительного аналитического центра при администрации – это семь человек со средней зарплатой 3 тыс. рублей».

Виктор Голоконский надеется, что громкий IT-проект повысит привлекательность региона в глазах инвесторов, равно как и имидж самого губернатора: «Кроме прямой экономии, электронные торги улучшают нравственную атмосферу вокруг госзаказа. Одна система проводит торги, никак не контактируя с поставщиками. Другая заключает контракты. Процесс разорван, и возможностей для злоупотреблений нет. Это другой имидж администрации, имидж региона. Если регион пошел на это, не боится прозрачности, значит с ним можно работать и по инвестициям». Стратегическую ставку новосибирский губернатор делает также на развитие научного потенциала региона и «сопутствующую» поддержку местного IT-бизнеса.

Кадры для госслужбы

Вообще, системы электронных закупок и электронного документооборота для госструктур становятся популярными IT-проектами (в связи с началом реализации программы «Электронная Россия»), наряду с учетными финансовыми системами (в связи с переходом на казначейское исполнение бюджетов всех уровней). И чиновникам придется решать серьезную проблему, связанную с острым дефицитом квалифицированных кадров, особенно в регионах. Впрочем, как считает первый заместитель федерального министра связи и информатизации Андрей Коротков, отношение к госслужбе начинает меняться: «Когда я еще работал в аппарате правительства, ко мне приходили молодые люди с просьбой взять их на работу. Я спрашиваю: «Вы знаете, какая у нас зарплата?» Они: «Знаем, но работа интересная». Я понимаю, что никто из них не собирается всю жизнь посвятить госслужбе, но они начали понимать, что эта работа дает определенный базис, помогает наладить горизонтальные связи». Кстати, карьера самого господина Короткова имеет вполне «техническую» основу. «Меня позвали в аппарат правительства – рассказал он, – чтобы наладить внутреннюю телеслужбу. С задачей я справился за полгода, но вот уже шесть лет состою на госслужбе». Андрей Коротков быстро дорос до руководителя департамента правительственной информации, приложил руку к переводу аппарата правительства на электронный документооборот, а в конце февраля этого года стал вторым человеком в Минсвязи РФ. Теперь он курирует от этого ведомства реализацию программы «Электронная Россия».

Заключение

Отечественная политическая наука должна стать технологичной, при этом не забывая о своем главном назначении – попечении о благе народа, обеспечении благосостояния для всех. На западе политология давно признается точной наукой, следовательно, она должна позволять упреждать, прогнозировать, моделировать социально-политическое развитие нашей страны.

Литература

1. Ольшевский А.С., Ольшевская А.С. Негативные PR-технологии. М.: ИНФРА-М, 2004.
2. Вайнштейн Г.И. Гражданское общество в информационную эпоху: Новые горизонты и новые проблемы. – <http://www.adenauer.ru>
3. Правдина М., Пичугин И. Политический козырь информационных технологий iOne.ru – <http://www.ione.ru>
4. UCLA Internet Report 2001. Surveying the Digital Future. Year Two. 2001, Los Angeles, p.82.

РАЗВИТИЕ ЛОГИЧЕСКИХ И ГНОСЕОЛОГИЧЕСКИХ ИДЕЙ И. КАНТА В РАБОТАХ РУССКИХ ЛОГИКОВ КОНЦА XIX – НАЧАЛА XX ВВ.

И.О. Ломова

Научный руководитель – к.ф.н., доцент Т.А. Новолодская

В работе проводится анализ логико-гносеологической концепции русских философов и логиков А.И. Введенского, И.И. Лапшина и Г.И. Челпанова, сформировавшихся под влиянием работ И. Канта.

Большинство работ логического и гносеологического содержания, вышедших в России в конце XIX – начале XX вв. были написаны в русле рационалистической философии и были связаны с анализом идей Канта.

Активное пробуждение интереса к Канту в России было, несомненно, связано с ростом уровня философско-теоретической рефлексии, потребности в более основательной разработке и осмыслении фундаментальных философских и мировоззренческих проблем.

Нельзя не признать, что прозвучавший на Западе призыв «назад к Канту» был услышан в России, и с этой точки зрения можно говорить об определенном параллелизме, синхронности философских процессов, происходивших в этих различных регионах.

Формально интерес к кантианству и неокантианству проявился в активизации переводов сочинений самого Канта, а также сочинений мыслителей различных школ западноевропейского неокантианства. Содержательной стороной этого процесса было то, что почти все направления русской мысли выразили свое положительное или отрицательное отношение к кантианству.

Философия И. Канта стала в России конца XIX – начала XX веков самой необходимой и в силу этого самой популярной философией. Мощное движение к метафизике, характерное для этого времени, подняло значение Канта на невероятную высоту. Какими бы смелыми ни были метафизические конструкции, какой бы очевидностью мистического опыта они не отличались, но их авторы всегда ощущали необходимость объяснить условия возможностей метафизического познания [2]. Академическое кантианство объединяет философов – последователей и продолжателей учения Канта – А.И. Введенского, И.И. Лапшина, Г.И. Челпанова. В своих работах они, дополняя, уточняя либо предлагая более простое выведение достоверного, научного знания, строили свою логико-гносеологическую систему.

С Александра Ивановича Введенского (1856–1925 гг.) началось становление кантианства в России. Интерес к философии заставил Введенского пойти на историко-филологический факультет Петербургского университета, где кафедра логики и философии была в то время представлена профессором М.И. Владиславлевым. После окончания университета Введенский готовился к профессуре, обучался в Лейпциге, Берлине и Гейдельберге. Впоследствии был избран приват-доцентом историко-филологического факультета Петербургского университета, где читал курс по логике и истории новой философии. После защиты диссертации «Опыт построения теории материи на принципах критической философии», получив степень магистра философии, Александр Иванович был приглашен на кафедру философии петербургского университета, кроме того, преподавал в Военно-юридической академии, Женском педагогическом институте, Высших женских курсах, долгое время преподавал историю философии на Бестужевских высших женских курсах. Его инициативой было создано первое в России философское общество при Петербургском университете, председателем которого он сам и являлся. Введенский предложил новую конструкцию философского критицизма, где переработал основные положения кантовской теории выведения научного знания. Собственное доказательство критицизма он называет логицизмом или «русским вариантом

критицизма». Суть последнего излагает так: «вследствие неизвестности, подчинено ли истинное бытие закону противоречия, метафизическое знание неосуществимо до такой степени, что нельзя даже узнать, трансцендентно ли истинное бытие или же оно имманентно. Одинаково допустимо и то и другое, если математика и естествознание считаются знанием» [1; 315].

«Русское доказательство критицизма» у Введенского выглядит следующим образом. Вся душевная жизнь складывается, с одной стороны, из ощущений и восприятий, а с другой – из представлений и суждений. Ощущения подвергаются логической обработке при помощи априорных идей – понятий (причина, действие, субстанция) и представлений (пространство и время). Априорные идеи являются формами деятельности сознания и логически от него неотделимы. Введенский трактует априорность в смысле необходимости исходных, недоказуемых в системе знания, но заведомо годных для знания суждений. Априорные идеи находятся в самом сознании, составляют неизбежную форму его деятельности, логически неотделимы от нее. Реальность априорных идей вытекает уже из их понятия как форм сознания. Все это сознается нами не иначе как в тех формах, которые высказаны в априорных идеях – причинность, пространство, время и т.д. Наши ощущения производятся в такой связи в тех формах, в каких мы только и можем их сознать. Знание возможно там, где можно пользоваться умозаключениями, а они, в свою очередь, обусловлены действиями логических законов. Где невозможно применение логических законов, там невозможно знание. Такова область непредставимого, или вещей в себе, и потому наши умозаключения о таких предметах не имеют никакого обязательного значения. Понятие вещи в себе важно для науки в качестве предела или границы логического знания.

Несомненное знание возможно только в том случае, если к числу субъективных, из нас самих возникающих элементов мира явлений принадлежат не только чувственно воспринимаемые его качества, но и сама их связь между собой. Если пространство, время, субстанция, причина и т.д. – формы нашего познания, то тогда всякое представление, понятие и т.д. будут подчинены этим формам. Это значит, что между знанием о предмете и способом существования предмета будет соответствие.

Научное построение критицизма А.И. Введенского в конечном итоге можно привести к следующему вопросу: если допустить как факт, что математика и естествознание образуют достоверное знание, то каковы условия, которые делают логически позволительным такое допущение, сообщая логическую правомерность нашей уверенности в достоверности математики и естествознания?

Кантовское решение этой проблемы сводится к следующим выводам:

1. Предметы математики и естествознания – время, пространство и все данные внутреннего и внешнего опыта – раскрывают перед нами не истинную сущность вещей, каковы они на самом деле, а всего лишь наши представления о вещах, т.е. внешнюю оболочку, изнанку вещей, как она отражается в нашем сознании, т.е. явление.

2. Никакая метафизика в смысле знания о «вещах в себе» не вправе притязать на авторитет науки.

3. Признав знанием математику, естествознание, мы логически неизбежно вынуждены признать полную несостоятельность всякой вообще метафизики как знания о вещах в себе.

Русский логик предлагает «новое и легкое доказательство философского критицизма», по его словам, критицизм очень легко доказать посредством рассмотрения способа действия логических законов мышления.

Аргументация в этом доказательстве такая: если допустить, что математика, естествознание на самом деле образуют достоверное знание, то такое допущение логически позволительно лишь с тем неотъемлемым условием, «чтобы мы имели неоспоримое право пользоваться в них умозаключениями» [1; 356]. В противном случае мы бы не

могли считать знанием какую-либо теорему или закон природы, так как и математические положения, и законы естествознания устанавливаются нами не иначе, как через посредство умозаключения; а так как умозаключения обусловлены логическими законами мышления, то никакие умозаключения недопустимы там, где нельзя опереться на логические законы мышления. Отсюда видно, что для решения вопроса о возможности метафизического знания необходимо исследовать логическую природу законов мышления, т.е. выяснить в первую очередь следующий вопрос: каковы они по своему действию – естественные или нормативные? «Естественными» законами автор называет те законы, которые действуют сами собой, независимо от нашего желания, например, закон тяжести, а под нормативными законами – такие законы, исполнение или неисполнение которых зависит от нас самих, от нашего умысла, например трудовой закон.

Четыре основные логические закона мышления Введенский разделяет на естественные и нормативные. Закон исключенного третьего – естественный, так как утверждая или отрицая данное сказуемое относительно его подлежащего, закон этот исключает возможность такой третьей связи между ними, которая могла бы заменить собою утверждение или отрицание, и тем самым свидетельствует о чисто естественной невозможности допустить такое третье отношение между подлежащим и сказуемым. «Мы, – пишет автор, – действительно не в состоянии мыслить такое отношение, которое не оказалось бы ни утвердительным, ни отрицательным, и как бы мы ни старались, мы прямо-таки не в силах придумать его». Естественный характер имеет и закона тождества: «мы прямо-таки не в силах при всем нашем желании поставить себя в такие условия, чтобы продолжать мыслить данную мысль и не отождествлять ее с ней самой».

Закон достаточного основания – чисто нормативный закон. Он предписывает нам не соглашаться с мыслью до тех пор, пока нам не приведут основания, достаточные для того, чтобы принудить нас согласиться с данной мыслью. Этот закон осуществляется не сам собою, а принудительно.

Закон противоречия более сложен, его анализу Введенский уделяет особое внимание. Закон носит двойственный характер и считается одновременно естественным и нормативным. Эту двойственность можно объяснить следующим образом: как показывает опыт, «нам приходится избегать противоречия в своих мыслях, а оно еще не устраняется и не исключается из них само собою, как это делается с третьим отношением сказуемого к подлежащему, которое запрещено законом исключенного третьего». Инстинктивно нам приходится делать над собой усилие, чтобы избежать противоречия, что свидетельствует о том, что закон этот нормативный. Ключ к объяснению этого кроется в смешении понятий «представление» и «мышление». По словам Введенского, надо отличать представление и мышление; эта разница с полной ясностью обнаруживается в том, что мы можем даже ясно мыслить и то, чего не в силах представить. Мы можем мыслить, и притом с величайшей ясностью, о чем вовсе не имеем никакого представления». Стоит нам разграничить эти понятия – и закон противоречия утрачивает свой мнимый двойственный характер. «Если мышлению доступно то, что решительно недоступно представлению, то естественный закон представлений еще не должен быть естественным законом мышления. Закон противоречия составляет естественный закон представлений, но как закон мышления он – нормативный. Мы в состоянии мыслить круглый квадрат, миллиард звезд, но не в состоянии представить. Тенденция подчинять мышление чуждому ему закону представлений объясняется тем, что мы обычно проверяем результаты мышления представлениями, обыкновенно считаем несущественным то, что непредставимо. Что касается умозаключений, то и они точно так же черпают санкцию своей достоверности в доверии к закону противоречия: ведь вывод любого правильного умозаключение становится для нас обязательным вследствие нашего согласия с посылками лишь постольку, поскольку мы противоречие считали неосуществимым». Неосуществимо противоречие в представлениях, поскольку они подчинены

естественному закону противоречия. Стремясь согласовать мышление с представлением, мы преднамеренно подчиняем его совершенно чуждому ему закону противоречия. Умозаключения логически позволительны лишь о представлениях и непозволительны о вещи в себе, поскольку последние не представимы. Так как знание о вещах в себе или факте их существования невозможно без умозаключения, а умозаключение может быть только о представлениях, то метафизики в смысле знания о вещах в себе не может быть. Из непознаваемости вещи в себе вытекает полная невозможность ни опровергнуть их существование, ни доказать его. Логически позволительно считать метафизику верой, верой критической, т.е. свободной от всяких опасений впасть в противоречие с наукой. Метафизика позволительна как морально обоснованная вера. И Кант тоже считал метафизику нравственной философией или морально обоснованной верой.

Другим талантливым представителем русского неокантианства является И.И. Лапшин. Главные положения своей гносеологии он изложил в труде «Законы мышления и формы познания». Лапшин, будучи учеником Введенского, не разделял метафизических построений своего учителя. Общую схему познания он полностью перенял у Канта, но задачу гносеологии ставил несколько иначе – как выяснение вопроса об отношении законов мысли к формам познания.

Лапшин задался целью несколько по-иному обосновать критицизм. Он пытается разрешить следующую проблему – возможно ли пользоваться законами мышления помимо всяких данных теории познания (имеется в виду кантовская теория познания) и распространять их на вещь в себе. Для решения этой проблемы Лапшин обращается к теории познания Канта, прослеживая конкретные формы связи между законами мышления и формами познания. Кант, замечает Лапшин, расположил законы человеческого познания как бы в три этажа: верхний этаж занимают законы формальной логики, средний этаж – законы трансцендентальной логики (категории), нижний этаж – законы трансцендентальной эстетики (формы интуиции, пространство и время). Фундамент здания – ощущения, содержание опыта.

Если обозначить законы мышления через S, категории через K, формы интуиции через F и чувственное содержание опыта через E, E1, E2 (так как содержание опыта представляет собой переменную величину), то задача исследования сводится к тому, чтобы выяснить, представляет ли всякое состояние сознания соединение всех этих элементов или нет. В последнем случае можно предположить: 1) что состояние сознания могут быть чистыми ощущениями E, E1, E2, не подчиненными ни законам мышления, ни категориям, ни формам интуиции, 2) что формы интуиции реально отделимы от ощущений, 3) что *actus purus* мысли реально отделимы от форм интуиции и ощущений, но подчинены категориям и законам мысли, 4) что *actus purissimus* мышления представляют процессы мышления, подчиненные его законам, но абсолютно не зависящие ни от ощущений, ни от форм интуиции, ни от категорий. Детальное рассмотрение этих четырех предположений, собственно, и составляет содержание довольно объемистого труда Лапшина «Законы мышления и формы познания».

Следуя за Кантом, Лапшин утверждает, что ощущения неотделимы от форм созерцания (интуиции), так как наше самонаблюдение не дает возможности установить наличие непротяженных и вневременных ощущений. Также неотделимы ощущения от категорий качества, количества и от так называемых категорий отношения – причинности, субстанциональности и взаимодействия. Неотделимы они и от единства самосознания, существование чистых ощущений вне форм созерцания, вне категорий и единства самосознания невозможно [4].

Рассматривая второе предположение, Лапшин приходит к выводу о том, что пространство и время неотделимы от чувственного содержания опыта. Лапшин дополняет, что пространство и время – не формы чистого созерцания, а только понятия, категории мысли, которые никогда не даны в созерцании.

Соглашаясь с Кантом, он придает категориям субъективное значение основных условий самого существования мира-представления. Область применения – опыт, но и самый опыт мыслим лишь при наличии категорий. Только в этом случае категории приобретают аподиктически достоверное значение объектных законов мира-представления.

Переходя к доказательству своего основного теоретического положения – законы мысли неотделимы от К, F, E, Лапшин подчеркивает, что «Я» есть простое логическое единство сознания, простое указание на то, что мир для меня – мое представление.

Единство самосознания заключается в способности отождествлять разнородный состав суждения в одном понятии и являться основой для закона тождества. Невозможность приписывать и не приписывать себе в одно и то же время известное состояние сознания есть условие для допущения закона противоречия. Законы мысли зависят и от категорий. Так, например, законы мысли опираются на синтез отождествления и различения и, следовательно, зависят от категории качества. Аналитические суждения и законы мысли, на которые они опираются, неотделимы от категорий количества, от пространственной интуиции, от аксиом времени. Законы мысли необходимо связаны с категориями причинности, субстанциональности и т. д.

Эти рассуждения, по мнению Лапшина, доказывают, что формы созерцания, категории, единство самосознания необходимо обуславливают самое существование законов мысли. Отсюда следует два важных вывода: во-первых, тот, кто отвергает значение критической теории, тем самым лишает себя права ссылаться на законы формальной логики; во-вторых, неизвестно, приложимы ли законы мысли, опирающиеся на категории, формы созерцания и ощущения, к вещам в себе.

Г.И. Челпанов определил свою позицию как «идеал-реализм». В целом он выступает с позиций Канта, но стремится очистить учение немецкого философа от психологических и физиологических интерпретаций. Противник материалистического воззрения, он стремился опровергнуть любые попытки реалистически толковать познавательные способности человека [5].

Философия, утверждали русские неокантианцы – прежде всего мировоззрения или, говоря иначе, совокупность ответов, возбуждаемых вопросом о смысле существования человека. Идеальная цель философии, замечает Г.И. Челпанов, – разрешение конечных проблем бытия. Но философия есть система научно разработанного мировоззрения, в этом ее отличие от религиозного или художественного мировосприятия. Системой научного мировоззрения философия становится тогда, когда она опирается на особую науку – теорию познания. Сам Кант отрицал мировоззренческое значение существовавшей в его время философии, полагая, что философия должна выполнять методологическую функцию. Ее назначение – изучать разум, который, в том числе, и производит идеи для рассудочного знания.

Таким образом, философия И. Канта, а точнее – его система научного знания, таящая в себе многочисленные варианты трактовки, оказала значительное влияние на формирование русской логико-философской традиции. Именно интерес к трансцендентальной логике позволял русским философам сориентироваться на проблеме выведения достоверного знания, что имело актуальное значение для науки.

Литература

1. Введенский А. И. Логика как часть теории познания. 3-е изд. СПб, 1922.
2. Введенский А. И. Иммануил Кант. Литограф. курс лекций, прочитанный в 1907 году студентам СПб. университета.
3. Водзинский Е. И. Русское неокантианство конца XIX – начала XX вв. Из-во Ленинградского ун-та, 1966.
4. Лапшин И. И. Логика мышления и формы познания. СПб., 1906.
5. Челпанов Г. И. Проблема восприятия пространства в связи с учением об априорности и врожденности. Киев, 1896. Ч.2.

ДЕТСТВО КАК ЭТНОКУЛЬТУРНЫЙ ФЕНОМЕН В ХУДОЖЕСТВЕННОЙ ИНТЕРПРЕТАЦИИ Л.Н. ТОЛСТОГО И Н.С. ЛЕСКОВА

Л.Н. Вшивцева (Ставропольский государственный университет)
Научный руководитель – д.филол.н., профессор В.М. Головкин (Ставропольский
государственный университет)

Проблематика работы относится к числу актуальных направлений фундаментального литературоведения, предусматривающих анализ духовных ценностей с точки зрения современного знания.

В XX веке большинство ученых подчеркивали, что фундаментальная роль в воспроизводстве культур отводится процессам, происходящим в детстве. Так, А.А. Белик заметил, что, «преобразуя окружающую действительность, ребенок тем самым формирует и воспроизводит важнейшее культурное качество человека в создании идеальных сущностей» [1, с. 108]. Л. Демоз утверждал, что исторические изменения происходят не только под влиянием техники и экономики, но, главным образом, под влиянием сменяющихся друг друга поколений родителей и детей. Более того, он отметил, что главным условием воспроизводства и развития элементов культуры является тип воспитания, распространенный в обществе. Мысль о том, что общество не может познать себя, не поняв закономерностей детства, обнаруживалась в произведениях европейских просветителей и романтиков XVIII века. Во второй половине XIX века эта мысль перекочевала в Россию. В произведениях писателей-классиков, таких как Л.Н. Толстой, Ф.М. Достоевский, И.А. Гончаров, Н.С. Лесков и др., детально анализируется семья ребенка, тип воспитания в этой семье, его первое знакомство с «большим» миром. Вследствие этих и других процессов складывается самосознание ребенка, формирующее в нем какое-то доминантное качество характера и постепенно приводящее его к той или иной жизненной позиции. А уже из этих составляющих его личности складывается общая культура народа. Так выглядит алгоритм развития ребенка почти во всех произведениях русской литературы о детстве второй половины XIX века – отсюда их сходство.

Одним из главных качеств русской национальной культуры XIX века многие мыслители (Достоевский, Л. Толстой, Бердяев, Франк, Лосский, Карсавин, Ильин и др.) считали высокую духовность, под которой понимали свойство души, состоящее в преобладании духовных, нравственных и интеллектуальных интересов над материальными. Проследим, как это качество зарождалось в человеке, для этого обратимся к произведениям двух великих современников – Л.Толстого и Лескова. В трилогии «Детство. Отрочество. Юность» (1852–1857) и повести «Детские годы» (1874) показано внутреннее движение героя, начиная с раннего детства и заканчивая поздней юностью, характеризующейся зрелостью в умственном и нравственном отношении. Несмотря на почти двадцатилетнюю разницу в написании, произведения имеют много совпадений в показе писателями нравственного становления героев. Эти совпадения объясняются, во-первых, господством во второй половине XIX века идеи европейских просветителей, о которой говорилось в начале статьи, во-вторых, заданным этой идеей алгоритме показа нравственного развития ребенка.

Детство главных героев повестей Толстого и Лескова можно разделить на жизнь в родительском доме (узкий, ограниченный мир) и жизнь вне дома: для Николеньки Иртеньева – это Москва, дом его бабушки; для Меркула Праотцева – это жизнь в кадетском корпусе. Самооценка главного героя толстовской трилогии осуществлялась не без влияния его близкого окружения, но важным остается тот факт, что наибольшее воздействие на мальчика оказывали не родители, а люди из его близкого окружения, в числе которых оказались гувернант Карл Иванович, экономка Наталья Савишна, юродивый Гриша. Своей любовью и самопожертвованием они возбудили в душе мальчика добрые чувства, кото-

рые никогда не умрут в его памяти. Детские раскаяния героя, вызванные осознанием того, что он плохо поступил с любящими его людьми: «Какой он добрый и как нас любит, а я мог так дурно о нем думать! Мне было досадно и на самого себя и на Карла Иваныча, хотелось смеяться и хотелось плакать: нервы были расстроены»; «У меня недоставало сил взглянуть в лицо доброй старушке; я, отвернувшись, принял подарок, и слезы потекли еще обильнее, но уже не от злости, а от любви и стыда», (о Наталье Савишне) [4, с. 52, 89] – обозначили начальный этап его самосознания. В повести Лескова таковым этапом стала ранняя самостоятельная жизнь героя в кадетском корпусе, где он получает первые уроки самопожертвования (срыв экзекуции кадетами).

Второй этап самосознания мальчиков – это отъезд из родного Петровского в Москву (для Николеньки) и отъезд домой, в Киев (для Меркула). На данном этапе происходит столкновение героев со средой. Иртеньев в доме бабушки познакомится с высшим обществом, оказавшим на него негативное влияние. Так, он с насмешкой отзовется о Карле Иваныче лишь для того, чтобы произвести впечатление на дочку светской дамы Сонечку, или обидит беззащитного мальчика Иленьку лишь потому, что это делали другие ребята. Позже он раскается в том, что совершил: «Как я не подошел к нему, не защитил и не утешил его? (Иленьку) Куда девалось чувство сострадания... Неужели это прекрасное чувство было заглушено во мне любовью к Сереже и желанием казаться перед ним таким же молодцом, как и он сам? Незавидные же были эти любовь и желание казаться молодцом! Они произвели единственные темные пятна на страницах моих детских воспоминаний» [4, с.114]. Почти то же произошло и с героем повести «Детские годы». Бал у Волосатиных, события на ярмарке в Королевце и, наконец, нелепая смерть Кнышенко привели Меркула к внутреннему кризису: «Я оплакивал свою погибшую жизнь, свое глубокое нравственное падение, страшно расстроившее мое воображение и нервы и доведшее меня до отчаяния, что я, сопричастясь бездне грязных пороков, уже не достоин и не могу взглянуть в светлые глаза моей матери, – что я лишил себя права обнять ее и принять ее поцелуй на мое скверное лицо, которое действительно осунулось и жестоко изменилось. Это произошло от большого нравственного страдания и мук, которые я испытывал, казня себя за всю развращенность, столь быстро усвоенною мною с тех пор, как я очутился на воле» [3, с. 323]. Спасательным кругом для героев повестей Толстого и Лескова окажутся их близкие: мама Николеньки и Наталья Савишна («Детство. Отрочество. Юность») и мама Меркула («Детские годы»). На маленького Иртеньева большее влияние оказала Наталья Савишна, повернувшая развитие героя в положительную сторону, сам Коля позже скажет о доброй старушке, что она «имела такое сильное и благое влияние на мое направление и развитие чувствительности...» [4, с. 145]. Для Меркула Праотцева встреча с матерью стала причиной его выхода из кризиса. Катерина Васильевна огородила сына от негативной среды, создав ему высокодуховное общество в своем лице, а также в лице семьи Альтанских, позже Филиппа Кольберга, приобщивших его к мировой культуре. Это новый этап в формировании самосознания героя, на котором он оказался между двумя точками зрения: идеей самоотвержения и необходимостью все «вещи ... принимать обыденнее». Носителями первой точки зрения явились мать Меркула, отказавшаяся от любви, принеся себя в жертву сыну, и Христя Альтанская, отвергшая любовь Сержа ради его самого же, однако, позже отказавшаяся от этой идеи во имя естественной жизни, но так и не сумевшая принимать вещи обыденнее. Носителями второй явились люди искусства, Лаптев и Кольберг, после смерти матери на эту точку зрения стал Меркул Праотцев: «Я не могу быть ученым... в душе моей горит огонь: огонь жизни; я хочу служить искусству» [3, с. 450]. Как известно, искусство связано с чувственной, эмоциональной стороной жизни и, следовательно, оно способствует нравственному, духовному и интеллектуальному развитию человека, и отрешает его от мира материальных ценностей.

Новый этап самосознания героя Толстого начнется в повести «Отрочество». Здесь он впервые столкнется с иной, отличной от его среды, действительностью – крепостничеством. Это столкновение заставит его по-иному взглянуть на окружающих его людей и мир вообще. Сам Николенька об этом периоде скажет: «Мне в первый раз пришла в голову ясная мысль о том, что не мы одни, то есть наше семейство, живем на свете, что не все интересы вертятся около нас, а что существует другая жизнь людей, ничего не имеющих общего с нами, не заботящихся о нас и даже не имеющих понятия о нашем существовании. Без сомнения, я прежде знал все это; но знал не так, как я это узнал теперь, не сознавал, не чувствовал» [4, с.163]. Вследствие нового анализа действительности у Иртеньева появится еще одна склонность – умствование, которая в сочетании с острой чувствительностью, привитой ему в детстве, приведет его к сложным нравственно-этическим и философским исканиям, отраженным писателем в «Юности». Это будет качественно новый, наивысший уровень его самосознания, характеризующийся высокой духовностью.

Таким образом, самосознание Николеньки Иртеньева и Меркула Праотцева способствовало развитию в них духовности, присущей русской культуре XIX века в целом. Нравственное становление героев осуществлялось под воздействием воспитания в родительском доме, за счет столкновения с реальной действительностью, а также путем общения к народной культуре. Следует отметить, что «ростки» духовности формировались, главным образом, в детском возрасте. Своими произведениями Л. Толстой и Лесков доказали, что детство есть процесс духовного воспроизводства важнейших культурных качеств человека.

Литература

1. Белик А.А. Культурология. Антропологические теории культур. М., 2000.
2. Головкин В.М. Русская реалистическая повесть: герменевтика и типология жанра. М.: Ставрополь, 1995.
3. Лесков Н.С. Собрание сочинений в 11 т. Т.5. М., 1957.
4. Толстой Л.Н. Собрание сочинений в 12 т. Т.1. М., 1975.

«МЫ» Е. ЗАМЯТИНА: Д-503 КАК РЕФЛЕКСИРУЮЩИЙ ГЕРОЙ

В.М. Сахарова (Ставропольский государственный университет)

Научный руководитель – д.филол.н., профессор Л.П. Егорова (Ставропольский государственный университет)

В статье анализируется одна из актуальных проблем творческого наследия Е. Замятина. Понятие рефлексии является ведущим в концепции психологизма писателя. Оно рассматривается на стыке двух наук – литературоведения и психологии. Дается его теоретическое обоснование, а также рассматривается художественное воплощение на примере конкретного текста романа Замятина «Мы».

Понятие «рефлексия» пришло в литературоведение из психологии. Оно занимает ведущее место в исследованиях проблемы «Я» («идея Я», «образ Я», «понятие Я», «Я-концепция»). Согласно теории С. Дюваля и Р. Виклунда, рефлексия начинается с сосредоточения внимания на себе и самооценки. И. С. Кон («Открытие Я», 1978) подчеркивает, что объектом рефлексии является сам субъект. Суть самосознания, по его мнению, заключена в умении человека «отделить себя как деятеля от процесса и результатов своей деятельности»¹. В литературоведении «рефлексия» – термин феноменологии, обозначающий осознание собственных мыслей и переживаний. «В бодрствующей жизни, – пишет Э. Гуссерль, – мы всегда заняты чем-то – то этим, то тем, причем на низшей ступени – непсихическим... Только рефлексия, поворот взгляда от непосредственно тематического вводит как тему в поле зрения психическую жизнь... В рефлексивном восприятии и вообще опытным постижении она охватывается и сама становится темой разнообразных занятий»².

Диалектика «Я» всегда была в центре внимания художников. Однако для одних писателей личность – это, прежде всего, активно действующее начало. Герой проявляется исключительно и всецело через свои поступки, что придает его образу внутреннее единство и автономную целостность (характерно для литературы соцреализма). Для других писателей объектом художественного исследования становится уже не деяние, а деятель, психологическая индивидуальность которого важнее той ситуации, в которой она проявляется. На передний имидж выдвигается субъективное начало, сам субъект предстает уже как «сложный, многогранный и многомерный»³. Внешняя точка зрения сменяется внутренней. Писатель пытается понять героя изнутри, изображает его как множественное и динамичное «Я». Здесь плодотворен путь развития традиций русской классики: И.С. Тургенева, Л.Н. Толстого, Ф.М. Достоевского. Предметом художественного исследования становится процесс самосознания.

С этой точки зрения особенно интересна многоступенчатая рефлексия замятинского Д-503. Свое выражение она нашла, прежде всего, в жанровом аспекте. Записи Д-503 – это дневник, исповедь, история психической болезни, солилоквиум – «беседа с самим собой», «диалогический подход к самому себе»⁴. В основе этих жанров лежит открытие «внутреннего человека». Такое жанровое своеобразие позволяет автору постепенно раскрыть героя от общего к частному, от общественного к индивидуальному.

Итак, Д-503. Изначально перед нами всего лишь номер, один из математиков Единого Государства, строитель Интеграла, социально-биологическое существо. Главный герой в своих первых записях лишь фиксирует и описывает происходящее вокруг, информирует читателя. Здесь имеет место обычное наблюдение, не предполагающее каких – либо духовных противоречий.

Во 2-й записи возникает диалог (Д-503 – О-90, Д-503 – I-330). Появляется описание чувств, эмоций. Д-503 смущается, смеется, теряется, а, значит, реагирует. Самоощущение героя переходит в более сложный процесс – самосознания. Здесь же находим элементы самооценки – неприязнь к своим рукам.

В 7-й, 8-й записях обращают внимание на себя уже внутренние процессы. Предметом изображения становится динамика мыслей, чувств, эмоций. В системе знаний Д-

503 возникает противоречие («иррациональный корень»), порождающее неприятные переживания. Три импульса выводят героя из психологической спячки: врожденная эмоциональность, посещение Древнего Дома и ошеломляющее потрясение от близости с I-330. Начинается период открытий – Д-503 открывает природу, любовь, культуру, историю, новый «отдельный мир», наконец, незнакомого «внутреннего человека»: «Но чувствую: живу отдельно от всех, один, огороженный мягкой, заглушающей звуки стеной, и за этой стеной – мой мир...»⁵. Герой находит свою душу. В его сознании формируется важная категория, определяющая личность, внутренняя доминанта, ценностное ядро – жажда свободы. Происходит перераспределение координат. Все меняется, начинает концентрироваться вокруг Д-503. Самосознание трансформируется в более сложный процесс – самоосознание.

Тем не менее, несмотря на положительную сущность внутренних изменений, поток новых впечатлений оказывается для героя губительным. Как результат – личностное расстройство – раздвоение личности, разделение самости: «Я стал стеклянным. Я увидел – в себе, внутри. Было два меня. Один я – прежний, Д-503, номер Д-503, а другой... Раньше он чуть высовывал свои лохматые лапы из скорлупы, а теперь вылезал весь, скорлупа трещала, вот сейчас разлетится в куски и... и что тогда?»⁶.

Высшая саморегуляция – ЭГО, самоосознание, трансформируется в *самопознание*. Социальное в Д-503 становится маленьким, незаметным, уходит на задний план. В центре – индивидуальное начало: биологическое (потребность быть с I –330) и психологическое (познание себя). Единое «Я» распадается на противоречия. Возникает «Я» как субъект мышления, рефлексивное. Это «Я» – активное, действующее, субъектное. Другое «Я» – как объект восприятия и внутреннего чувства, рефлексивное, категориальное. Сущность рефлексивного Д-503 складывается из двух составляющих – социальной детерминированности (социально-групповая принадлежность) и «самоатрибуции» (заклчения о себе, своих внутренних состояниях, наблюдение и оценка своего поведения в различных ситуациях). Герой пытается познать себя – это уже самоанализ (Эго-анализ). Он ищет ответы на вопросы: «что хочу?», «что надо от жизни?», «что надо от самого себя?» Тем не менее, противоречия остаются неразрешимыми, тайна «себя самого» – неразгаданной. Самореализация невозможна в полной мере из-за неспособности к самозащите, из-за неготовности к борьбе. Единое Государство убивает все внутреннее, живое. Убивает личность. Д-503 умирает как писатель, как историк, как психолог. Вновь остается биологическое существо и единица «мы». Герой испытывает катарсис – очищение, душевное облегчение после сильных эмоциональных переживаний. Однако это приводит к другому психическому заболеванию – деперсонализации, обезличиванию. Вместе с поражением человека оказывается ненужной, бесцельной его история, а будущее как временная категория вообще перестает существовать. Так история болезни заканчивается положительно: «Потому что я здоров, я совершенно, абсолютно здоров»⁷. Исповедь остается неслышанной. Дневник оказывается ненужным и нелепым, а самоанализ – незавершенным и бессмысленным.

Литература

1. Кон И.С. Открытие «Я». М., 1978. С.84.
2. Цит. по Цургановой Е. А. Рефлексия / Энциклопедия зарубежного литературоведения XX века. М., 2004. С.367.
3. См. сноску 1. С.241.
4. Цит. по Скалон Н. Р. О «ценностном центре» в дистопии / Русский роман XX века. Духовный мир и поэтика жанра. Саратов, 2001. С. 152.
5. Замятин Е. Мы. Ставрополь, 1990. С. 79.
6. См. сноску 5. С.50.
7. См. сноску 5. С.164.

АКТУАЛИЗАЦИЯ НАСЛЕДИЯ АЛЕКСАНДРА ЯШИНА В СОВРЕМЕННОЙ СОЦИОКУЛЬТУРНОЙ СИТУАЦИИ

Ю.Р. Юхневич (Ставропольский государственный университет)

Научный руководитель – д.филол.н., профессор Л.П. Егорова (Ставропольский государственный университет)

Статья раскрывает актуальность проблемы изучения наследия А. Яшина с связи признанием его творчества современной критикой и интересом современного литературоведения к проблеме неосоциологической интерпретации рассказа А. Яшина «Вологодская свадьба».

Опубликованный сорок с лишним лет назад рассказ А. Яшина «Вологодская свадьба» после долгого забвения воспринимается как остро современное произведение. Это стимулируется интересом современного литературоведения к неосоциологической интерпретации художественного текста¹ и признанием вспомнившей, наконец, о А. Яшине текущей критики: «В Яшине всегда сначала говорила совесть и только потом – художник. Теперь искусство празднует свободу, но без нравственного корня, без высокого смысла и самая причудливая художественность набивает оскомину»².

В свое время в адрес «Вологодской свадьбы» было сказано много несправедливого. А. Яшина упрекали в «очернительстве», в искажении жизни. Но писатель изображал действительность не такой, какая она должна быть, а такой, какой была она на самом деле. Любовь к своему народу и совесть художника не позволили ему создать идиллическую картину «Вологодской свадьбы» и, шире, жизни русского северного села середины XX века. Время показало, что правы были немногочисленные защитники произведения (А. Макаров, А. Романов, К. Симонов, А. Михайлов и др.)

Интерпретация рассказа Яшина «Вологодская свадьба» через социально-психологическую призму нисколько не умаляет, а, напротив, высвечивает своеобразие художника, позволяет по достоинству оценить мастерство композиции, важность ведущих мотивов, непосредственное развитие традиций русской классики. Здесь мы можем поспорить с И.В. Ничипоровым³, который, рассматривая традиции И.А. Бунина в «деревенской прозе», во-первых, не учитывает произведения Яшина, во-вторых, бунинская традиция рассматривается им односторонне, вне национальной самокритики Бунина, которую можно увидеть и у Александра Яшина, и у Федора Абрамова. Можно проследить определенную преемственность во взглядах Бунина и Яшина на саму деревенскую свадьбу: тревога и боль за человеческие судьбы сближает позиции писателей. Очевидная документальность рассказа «Вологодская свадьба» не снижает его художественной ценности, а позволяет автору выступить в двойной роли: он и объективный повествователь, прекрасно знающий людей, быт, характеры, обычаи, он и действующее лицо, принимающее активное участие в событии. (Необходимо также сказать, что сам А.Яшин при правке «Вологодской свадьбы» заменил первоначальное жанровое определение «очерк» на более, по его мнению, точное – «рассказ–очерк»).

Свое повествование А.Яшин ведет от первого лица как свидетель и очевидец, репортажно – по порядку событий. Лирическое начало проявляется достаточно зримо. Образ рассказчика вполне завершен и самостоятелен, более того, он формирует, в конечном счете, звучание произведения. Надо отметить, что присутствие автора в произведении, его действия в нем специфичны: если в начале и в конце (в своеобразном прологе и эпилоге) он – действующее лицо, то в картинах свадьбы он сам не действует, а лишь повествует о действиях других, хотя и здесь ощутимы его взгляд, его отношение, его интонация. Автор выступает как некий «летописец», задача которого – точнее воспроизвести главное, характерное в событии. Он выхватывает из происходящего «действа» те моменты, те ситуации, тех героев, которые кажутся ему наиболее важными.

Главная боль автора «Вологодской свадьбы» – не в том, что уходят из жизни пряслицы и веретена, а в том, что не пришла еще большая подлинная культура сельскохозяйственного производства. Яшин пишет, что «раньше, на конных подводах, можно было рассчитывать время довольно уверенно, теперь же дороги разбиты настолько, что в весеннее-осенней распутице, а зимою в метели и снегопады движение по тракту надолго прекращается вовсе». Старый уклад жизни родной писателю «сельщины» был далеко не прекрасным, останавливать движение жизни невозможно, да и не нужно; беды северной деревни, описанные в «Вологодской свадьбе» и других произведениях, происходят от того, что она отошла от прежней жизни, но медленнее, чем хотелось бы, приближается к новой. Не вследствие, а от недостатка развития новых социальных начал страдало яшинское Блудново, страдала вся северная деревня.

Именно поэтому с самого начала в композиции рассказа наблюдаются жизненные контрасты, причудливое соединение старого и нового. На контрастах, в основном, и построена композиция «Вологодской свадьбы»: с одной стороны, изображена милая сердцу старина (именно поэтому Яшин так подробно, со всеми деталями описывает свадебный обряд); с другой – он обращается к этой старине, к этим обрядам поневоле, поскольку новое еще не дошло до далекой вологодской деревни, нет ни кино, ни клуба, а «молодости праздники необходимы!» (Вот поэтому для молодежи «свадьба была чем–то вроде самостоятельного спектакля».)

Соединение нового и старого обнаруживается и в поведении, и во взаимоотношениях гостей, в характере героев. Прежде всего это касается героини произведения, Гали. Правда, следует сразу же отметить, что те черты «старого», которые автор отмечает в образе невесты, – это скорее извечные черты русской женщины, и они вовсе не являются отрицательными, чертами косности и старой деревенской забитости. Отсутствие же этих последних качеств и есть то новое, и едва ли не самое важное, что обрела женщина в наше время. «Среди мужчин на пиру очень скоро, – замечает автор, – объявляются типично русские правдоискатели, ратующие за справедливость, за счастье для всех... Таким людям не до веселья... Они обличают, разоблачают, требуют возмездия, протестуют и все время спрашивают: что делать, как быть, кто виноват?» – повествует писатель, готовя читателя к встрече с Василием Прокопьевичем, с характером, близким Яшину по духу, по натуре и по жизненным позициям.

Он – «бунтарь по натуре, – представляет его автор. – Он забывает про еду и пиво, как только начинает рассказывать о непорядках в лесу, при этом лицо его бледнеет, глаза блестят и требуют ответа на все его вопросы, какие ставит перед ним жизнь. А ездит он широко и знает много». И до всего ему есть дело. Притом человек этот из тех, кто к себе еще требовательней, чем к другим, и живет по совести. Когда в поле зрения автора попадает Василий Прокопьевич, то как бы уходит на задний план свадьба.

В «Вологодской свадьбе», помимо многоголосья народных характеров, поставлены многие проблемы и затронуты важные, смущающие благополучное спокойствие вопросы. Современный читатель по достоинству может оценить эту повесть, убедиться в зоркости и точности яшинского взгляда на жизнь, на ее болевые точки.

«Вологодская свадьба», а также многие другие произведения А.Яшина, впитавшие в себя основные мотивы и искания литературы 1950–1960-х гг. XX века, легли в основание «деревенской прозы» или, как ее называют еще, «почвеннической литературы», сохранившей свое значение и для сегодняшнего поколения читателей.

¹ Егорова Л.П. Задачи неосоциологической интерпретации современной русской прозы // Русская литература XX – XXI веков: проблемы теории и методологии изучения: Материалы Международной научной конференции: 10–11 ноября 2004 г. М., 2004. С. 29–33.

² Шеваров Д. Тропинка к Яшину // Деловой вестник. 2004. № 20 от 29 июня. С. 4.

³ Ничипоров И.Б. На путях постижения русской души: И.А. Бунин и «деревенская проза. // Филологические науки. – 2002. – № 1. – С.30 – 37.

Предисловие	3
Секция 1. Технологии обработки информации и программирования	5
Коротков М.А. (научный руководитель – к.ф.-м.н., доцент Ф.А. Новиков). Алгоритм укладки диаграмм состояний.....	5
Кучер А.В. (научный руководитель – д.т.н., профессор А.В. Дёмин). Вероятностная модель системы поиска и наведения	12
Корнеев Г.А. (научный руководитель – д.т.н., профессор А.А. Шалыто). Технология разработки визуализаторов алгоритмов	18
Шалаев М.П. (научный руководитель – д.т.н., профессор И.П. Гуров). Математическое моделирование аномалий в компьютерных сетях	24
Зарафьянц А.А. (научный руководитель – д.т.н., профессор Ю.А. Гатчин). Технический анализ метрик процесса разработки программных САПР	33
Корнеев Г.А., Станкевич А.С. (научный руководитель – д.т.н., профессор В.Г. Парфёнов). Методы тестирования решений задач на соревнованиях по программированию.....	36
Петров К.В. (научный руководитель – д.т.н., профессор О.Ф. Немолочнов). Метод верификации подпрограмм, написанных на языке высокого уровня.....	41
Шопырин Д.Г. (научный руководитель – д.т.н., профессор А.А. Шалыто). Реализация мультиметодов на языке программирования С++.....	46
Каунов И.В. (научный руководитель – д.э.н., профессор О.В. Васюхин). Обоснование выбора языка моделирования для имитации поведения опытного производства	49
Кучер А.В. (научный руководитель – д.т.н., профессор А.В. Дёмин). Имитационное моделирование оптического поиска	52
Яковлев Ю.А. (научный руководитель – к.т.н, доцент Т.А. Павловская). Среда исследования методов генерации входных данных для структурного тестирования программного обеспечения.....	60
Корнеев Г.А. (научный руководитель – д.т.н., профессор А.А. Шалыто). Метод преобразования программ в систему взаимодействующих автоматов.....	65
Гуров В.С., Мазин М.А. (научный руководитель – д.т.н., профессор А.А. Шалыто). Создание системы автоматического завершения ввода с использованием пакета UniMod	73
Озёрский А.М. (научный руководитель – д.т.н., профессор И.П. Гуров). Проектирование подсистемы фильтрации данных для универсального контроллера управления авторулевым устройством	88
Будько М.Ю. (научный руководитель – д.т.н., профессор Т.И. Алиев). Оценка качества передачи голоса в сетях IP-телефонии	92
Шалаева М.Б. (научный руководитель – д.т.н., профессор А.Ю. Тропченко). Повышение эффективности алгоритмов кодирования речи.....	98
Сидоров А.В. (научный руководитель – д.т.н., профессор О.Ф. Немолочнов). Верификация вычислительного процесса программы с использованием комплексных кубических покрытий.....	104

Секция 2. Информационно-управляющие системы	110
Климанов В.А. (научный руководитель – д.т.н., профессор А.Г. Коробейников). Оценка функционирования Real-Time OLAP систем	110
Дорожкин А.К. (научный руководитель – к.т.н., профессор В.В. Кириллов). Модель функционирования HOLAP систем	115
Шошмина И.В. (научный руководитель – д.ф.-м.н. профессор А.В. Богданов). Разработка российского сегмента Grid для высокопроизводительных приложений	121
Костин М.В. (научный руководитель – д.т.н., профессор Т.И. Алиев). Концептуальная модель системы условного доступа.....	129
Чэнь Жуймин (научный руководитель – к.т.н., доцент А.О. Ключев). Анализ процессов в операционной системе реального времени.....	138
Лашкевич А.Е. (научный руководитель – к.т.н., доцент Т.А. Павловская). Обзор отечественных систем моделирования управления предприятиями.....	142
Егоров С.В. (научный руководитель – д.т.н., профессор Ю.А. Гатчин). Каналы несанкционированного доступа к информации и средства их защиты в комплексах управления подвижными объектами	148
Каменева С.Ю., Михайличенко О.В. (научный руководитель – д.т.н., профессор А.Г. Коробейников). Электронная подпись.....	155
Петров В.А., Повышев В.В., Шилов И.В. (научный руководитель – к.т.н., доцент Н.Ф. Гусарова). Организация персонального документооборота с поддержкой ассоциативных связей пользователя	159
Малинин А.А. (научный руководитель – д.т.н., профессор Ю.А. Гатчин). Обеспечение целостности данных системы и предотвращение утечки информации	162
Костина А.В. (научный руководитель – д.т.н., профессор Т.И. Алиев). Концептуальная модель системы передачи и приёма сервисной информации.....	166
Секция 3. Информационные технологии в образовании и защита информации	173
Шлюжайте Ю.В. (научный руководитель – д.ф.-м.н., профессор Ю.Л. Колесников). Использование компьютерного тестирования для осуществления итогового контроля знаний студентов в области физики	173
Казаков М.А. (научный руководитель – д.т.н., профессор В.Г. Парфёнов). Реализация концепции многоуровневой системы дистанционного обучения на базе Интернет-школы программирования	176
Койнов Р.В. (научный руководитель – к.т.н., доцент Л.С. Лисицына). Принципы переноса образовательных Internet-ресурсов на карманные компьютеры.....	184
Проценко Е.А. (научный руководитель – д.т.н., профессор Л.Г. Осовецкий). Структура федерального законодательства РФ в области защиты информации	190
Коптяев К.А. (научный руководитель – д.т.н., профессор Л.Г. Осовецкий). Информационная безопасность интеллектуальных сетей	194
Шеламова Т.В. (научный руководитель – д.ф.-м.н., профессор Ю.Л. Колесников). Использование информационных ресурсов для создания виртуальных экспозиций, посвящённых знаменательным датам университета.....	197
Секция 4. Гуманитарные науки.....	200
Вергезова С.М. (научный руководитель – к.т.н., профессор М.И. Потеев). К вопросу о совершенствовании преподавания дисциплины «Концепции современного естествознания».....	200
Базарова Ж.С. (научный руководитель – д.э.н., профессор С.Б. Смирнов). Основные тенденции развития рынка труда в Республике Бурятия.....	205

Харжавин С.В. (научный руководитель – д.э.н., профессор О.В. Васюхин). Проблемы инновационно-инвестиционного пути развития российской экономики и способы их решения.....	214
Емельянов Н.С. (научный руководитель – д.э.н., профессор О.В. Емельянов). Проблема создания интегрированной системы управления предприятием	217
Орлов О.И. (СПбГУКиТ) (научный руководитель – д.соц.н., профессор Л.Я. Орлова). Особенности формирования российского менталитета.....	221
Алексеев Г.В. (научный руководитель – д.соц.н., профессор Л.Я. Орлова). Международно-правовые проблемы формирования общего информационного пространства странами СНГ	227
Скопцов В.В. (научный руководитель – д.соц.н., профессор Л.Я. Орлова). Социальный фрактал как фактор минимизации уровня неопределённости в социуме	238
Косенко Е.Ю., Харламова Е.А. (научный руководитель – д.полит.н., профессор И.Б. Орлов). Политика и Интернет: политическая коммуникация в условиях развития современных информационных технологий	243
Бородин М.В., Ермаков В.А. (научный руководитель – д.соц.н., профессор Л.Я. Орлова). К проблеме взаимного проникновения информационных и политических технологий	249
Ломова И.О. (научный руководитель – к.ф.н., доцент Т.А. Новолодская). Развитие логических и гносеологических идей И. Канта в работах русских логиков конца XIX – начала XX вв.	254
Вшивцева Л.Н. (Ставропольский государственный университет) (научный руководитель – д.филол.н., профессор В.М. Головки (Ставропольский государственный университет)). Детство как этнокультурный феномен в художественной интерпретации Л.Н. Толстого и Н.С. Лескова	259
Сахарова В.М. (Ставропольский государственный университет) (научный руководитель – д.филол.н., профессор Л.П. Егорова (Ставропольский государственный университет)). «Мы» Е. Замятина: Д-503 как рефлексирующий герой	262
Юхневич Ю.Р. (Ставропольский государственный университет) (научный руководитель – д.филол.н., профессор Л.П. Егорова (Ставропольский государственный университет)). Актуализация наследия Александра Яшина в современной социокультурной ситуации.	264

Вестник II межвузовской конференции молодых ученых. Сборник научных трудов / Под ред. В.Л. Ткалич. Том 1. СПб: СПбГУ ИТМО, 2005. 268 с.

**ВЕСТНИК II МЕЖВУЗОВСКОЙ КОНФЕРЕНЦИИ
МОЛОДЫХ УЧЕНЫХ
Сборник научных трудов
Том 1**

Научный редактор – д.т.н., профессор В.Л. Ткалич

Дизайн обложки Т.В. Точилина

Редакционно-издательский отдел СПбГУ ИТМО

Зав. РИО Н.Ф. Гусарова

Лицензия ИД № 00408 от 05.11.99.

Подписано в печать 12.06.05.

Заказ 857. Тираж 100 экз.