

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

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



ПОБЕДИТЕЛЬ КОНКУРСА ИННОВАЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ ВУЗОВ

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК

**САНКТ-ПЕТЕРБУРГСКОГО ГОСУДАРСТВЕННОГО
УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ**

июль-август 2008

№ 56

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
И ТЕЛЕКОММУНИКАЦИОННЫЕ
СИСТЕМЫ**



ГЛАВНЫЙ РЕДАКТОР

д.т.н., профессор В.О. Никифоров

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

д.т.н., доцент А.А. Бобцов, д.т.н. А.В. Бухановский,
д.т.н., профессор В.А. Валетов, д.ф.-м.н., ст.н.с. Т.А. Вартамян,
д.т.н. М.А. Ган, д.т.н., профессор Ю.А. Гатчин, д.т.н., профессор А.В. Демин,
к.т.н., доцент Н.С. Кармановский (заместитель главного редактора),
д.ф.-м.н., профессор Ю.Л. Колесников, д.ф.-м.н., профессор С.А. Козлов,
д.т.н., профессор А.Г. Коробейников, д.т.н., профессор В.В. Курейчик,
д.т.н., доцент Л.С. Лисицына, к.т.н., доцент В.Г. Мельников,
д.т.н., профессор Ю.И. Нечаев, д.т.н., профессор Н.В. Никоноров,
д.т.н., профессор А.А. Ожиганов, д.ф.-м.н., ст.н.с. Е.Ю. Перлин,
д.т.н., профессор И.Г. Сидоркина, д.т.н. О.А. Степанов,
д.т.н., профессор В.Л. Ткалич, д.т.н., профессор А.А. Шалыто

Секретарь – Г.О. Котелкова

Редактор – к.т.н., ст.н.с. Н.Ф. Гусарова

Адрес: 197101, Санкт-Петербург, Кронверкский пр., 49, СПбГУ ИТМО

Телефон: (812) 233 12 70

Факс: (812) 233 12 70 (с пометкой: для редакции
Научно-технического вестника)

[http: //books.ifmo.ru/ntv](http://books.ifmo.ru/ntv)

E-mail:karmanov@mail.ifmo.ru

Подписано к печати 30.08.2008 Тираж 350 экз. Заказ № 4(56)

Отпечатано в учреждении «Университетские телекоммуникации»
Адрес: 197101, Санкт-Петербург, Кронверкский пр., 49.

УДК 004.056.5

**МЕТОДИКА ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ
ПРОЦЕССА ОЦЕНКИ ЗАЩИЩЕННОСТИ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ ПО «ОБЩИМ КРИТЕРИЯМ»****О.Е. Зайцев, А.В. Любимов**

При построении функциональной модели «Общих критериев» (ОК) используется методика SADT и ее нотация DFD. Методика SADT изначально не может адекватно отобразить модель ОК в связи с определенной спецификой объекта моделирования, поэтому для получения методики функционального моделирования ОК в методику SADT необходимо внести некоторые коррективы.

Ключевые слова: модель, методика, моделирование, функциональный.

Введение

Национальный стандарт безопасности ГОСТ Р ИСО/МЭК 15408-2002 «Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий» («Общие критерии», ОК) действует в России с 1 января 2004 года. Опыт использования ОК говорит о том, что применение методологии ОК способствует существенному повышению качества оценки и разработки продуктов и информационных технологий (ИТ). Таким образом, оценка защищенности ИТ по ОК является перспективным направлением, а проблемы, связанные с использованием ОК, очень актуальны.

Решению многих задач способствует представление ОК в виде формальных моделей. Как правило, функциональная модель является основной и наиболее применяемой на практике. Функциональная модель ОК может быть построена с помощью методики структурного анализа и проектирования систем SADT. Данная методика изначально не может должным образом отобразить модель ОК, что связано с определенной спецификой самого объекта моделирования. Поэтому для получения методики функционального моделирования ОК в методику SADT необходимо внести некоторые коррективы. В настоящей работе представлены результаты разработки методики построения функциональной модели деятельности оценщика и разработчика по оценке защищенности ИТ на основе методики SADT, скорректированной с учетом специфики ОК как объекта моделирования.

Постановка задачи

Автоматизация многих действий по оценке и сертификации продуктов и систем ИТ возможна благодаря строгой регламентации деятельности оценщика и разработчика, описанной в «Общей методологии оценки» (ОМО) и в ОК. Для проектирования, разработки и сопровождения соответствующего программного обеспечения необходимо иметь функциональные спецификации в стандартизованной электронной форме. Представление деятельности по оцениванию защищенности ИТ, регламентируемой ОК, в виде стандартизованной функциональной модели предоставляет удобные средства контроля версий стандарта и обеспечивает возможность прослеживания последствий принимаемых в новых версиях изменений вплоть до уровня конкретных операций, что существенно облегчает как работу сотрудников испытательных лабораторий при проведении оценки, так и работу разработчиков при подготовке к ней. Наименее подготовленной к предстоящему внедрению ОК группой пользователей являются заказчики ИС и покупатели готовых продуктов. Для них формализованное графическое представление ОК является кратким справочником.

Перечисленные факторы обуславливают актуальность функционального моделирования методологии ОК и определяют постановку задачи моделирования.

Исходя из сравнительного анализа методик моделирования, для построения функциональной модели ОК была выбрана методика SADT, поскольку в данной области моделирования она практически не имеет альтернативы, методики ARIS и CALS являются избыточными в данном случае. SADT-модель дает полное, точное и адекватное описание системы, имеющее конкретное назначение. Обзор зарубежных источников выявил лишь одну попытку функционального моделирования фрагментов ОК [1]. Однако при построении диаграмм реально не использовалась какая-либо определенная методика или метод, фактически они представляют собой иллюстрации, а не формализованную модель. Модель построена не по стандарту, что не позволяет говорить об ее дальнейшей применимости в области оценки и сертификации ИТ вследствие неполноты и субъективности.

В отечественной литературе идея использования методики SADT в совокупности с методом DFD для построения формальной модели процессов оценки безопасности ИТ по стандарту ОК была предложена в [2] и получила дальнейшее развитие в [3–5]. Методика построения функциональной модели деятельности по оценке защищенности ИТ, о которой говорится в настоящей работе, в целом соответствует методологии SADT, т.е. использует принцип пошаговой детализации процессов и ресурсов. Однако специфика объекта моделирования приводит к необходимости внесения в методологию SADT весьма существенных корректив, а в некоторых случаях вынуждает использовать приемы, идущие вразрез со сложившейся практикой функционального анализа. В данной работе затрагиваются такие сложности декомпозиции и приемы их преодоления.

Методика построения функциональной модели

Детализация процессов

Детализация процессов функциональной модели проводится по принципу «сверху вниз», т.е. класс доверия детализируется на семейства, семейства – на компоненты, компоненты – на элементы.

Имена процессов не совпадают с именами соответствующих классов, а, как правило, являются именем класса с добавленным отглагольным существительным. Для дополнительного отслеживания соответствия модели тексту стандарта к имени процесса в скобках добавляется идентификатор класса. Аналогичное правило именования применяется и к семействам доверия, например, «Оценка анализа уязвимостей (AVA_VLA)» и т. д.

Определение для процессов уровня классов извлекается из третьей части ОК и ОМО, причем в качестве источника определения выбирается раздел с наиболее полной информацией. При наличии расхождений предпочтение отдается ОМО. Ссылка на источник заносится во вкладку Note свойств процесса. Аналогичная процедура применяется для процессов уровня семейств.

При моделировании действий разработчика некоторые компоненты семейств, например, содержат действия, которые лежат вне принятых границ моделирования и поэтому не должны отображаться на диаграммах модели. Однако эти действия являются существенными для процесса оценивания (как правило, вследствие того, что порождают важные для оценки ресурсы – документы) и потому должны в модели каким-то образом учитываться. Противоречие может быть разрешено двумя способами.

Во-первых, подобный компонент можно включить в модель как процесс, не имеющий входов (что противоречит синтаксису стандарта DFD). Выходы такого процесса отображаются, например, пунктиром, чтобы напомнить о том, что они являются, вообще говоря, элементами свидетельств оценки, поступающими от разработчика, а не внутренними ресурсами данной диаграммы. Подобный процесс можно дополнительно выделить с по-

мощью стандартных графических примитивов. Во-вторых, подобный компонент можно вообще не размещать на диаграмме, разместив вместо нее экземпляр внешней сущности «Разработчик» с выходящими из него необходимыми ресурсами, выделив их определенным образом, но этот вариант не допускает дальнейшей детализации процесса.

Практически все компоненты семейств доверия имеют иерархическую структуру. С ростом оценочного уровня доверия (ОУД) компонент с меньшим номером заменяется компонентом с большим номером, который обеспечивает большее доверие к безопасности (усиление компонента). Это означает, что, несмотря на то, что на диаграмме отображаются все компоненты семейства, при каждой конкретной процедуре оценивания будет выполняться только один из них. Подобную ситуацию невозможно явно отразить на DFD диаграмме, поскольку этот стандарт не содержит средств отображения логики процессов, однако ее можно описать с помощью текстового комментария.

Представление подобных иерархий методом функционального моделирования всегда вызывает большие проблемы. Причина состоит в том, что, по существу, подобные иерархии представляют собой схемы наследования, т.е. являются предметом рассмотрения в объектной области (структурное моделирование), а для представления их в рамках функциональной модели приходится вводить существенную избыточность как для процессов, так и для ресурсов. Иерархичность компонентов выражается также и в их именовании, т.е. в стандарте используются сокращенные имена, которые неточно, а иногда и превратно отражают содержание соответствующих процессов. Например, «Независимый анализ уязвимостей», помимо анализа уязвимостей оценщиком, включает в себя еще множество других действий. В итоге пришлось отказаться от полной привязки имен процессов к именам компонентов в стандарте и трактовать последние просто как названия разделов. Процессу, который соответствует компоненту, дается оригинальное имя, в наибольшей степени соответствующее его содержанию, а для привязки к стандарту используется идентификатор компонента (например, AVA_VLA), именование процессов, соответствующих элементам, производится по их содержанию, описанному в ОК, с добавлением идентификатора.

Для получения точного представления о структуре семейства можно составить таблицу, представляющую разбиение семейства на компоненты и на элементы, на которой проще отследить различия и избыточность. В этой таблице столбцами являются компоненты, а строками – номер ОУД и компонента, а также элементы действий оценщика и разработчика. В таблице схематично показана подобное разбиение, где XXX – идентификатор класса, YYY – идентификатор семейства, D – идентификатор разработчика, E – идентификатор оценщика.

ОУД	1	2	3	4
Компонент	1	2	3	4
Элементы действий разработчика	XXX_YYY.1.1D описание компонента	XXX_YYY.2.1D описание компонента	XXX_YYY.3.1D описание компонента	XXX_YYY.4.1D описание компонента
Элементы действий оценщика	XXX_YYY.1.1E описание компонента	XXX_YYY.2.1E описание компонента	XXX_YYY.3.1E описание компонента	XXX_YYY.4.1E описание компонента
Элементы действий разработчика	XXX_YYY.2.1D описание компонента	XXX_YYY.2.1D описание компонента	XXX_YYY.2.1D описание компонента	XXX_YYY.2.1D описание компонента
Элементы действий оценщика	XXX_YYY.2.1E описание компонента	XXX_YYY.2.1E описание компонента	XXX_YYY.2.1E описание компонента	XXX_YYY.2.1E описание компонента
...	...	...	...	...

Таблица. Детализация семейства

Детализация ресурсов

Под входными ресурсами в данной модели понимаются различные компоненты свидетельств оценки, под выходными – результаты выполнения действий оценщика, как правило, заключения и их обоснования. Все прочие ресурсы относятся к внутренним, если не оговорено противное.

Иерархичность компонентов доверия порождает избыточность не только при декомпозиции процессов, но и при декомпозиции ресурсов для компонентов. При этом следует учитывать, что содержимое одного и того же ресурса меняется в зависимости от того, для какого компонента этот ресурс предназначен, и это отличие должно быть отражено не только в определении ресурса, но и в его имени. Однако при детализации ресурсов, помимо эффекта иерархичности, приходится учитывать и другие, не менее значимые факторы. Прежде всего, это «обратное» направление детализации.

Детализация ресурсов выполняется по принципу «снизу вверх», что соответствует структуре разделов стандарта: из простых ресурсов собираются составные. Это противоречит методологии SADT (в соответствии с которой ресурсы тоже должны детализироваться сверху вниз, вместе с процессами и по мере их детализации), но в данной задаче это допустимо, поскольку детализация процессов практически регламентирована.

Вторым существенным фактором является то, что для детализации ресурсов приходится использовать англоязычный оригинал стандарта Common Criteria [6–8] для более полного и точного построения модели. Дело в том, что в ОК изменена структура свидетельств оценки (при сохранении суммарного содержания), которые являются базовыми ресурсами модели, и эти изменения, к сожалению, проведены не в лучшую сторону. Например, в этих стандартах одному и тому же компоненту может соответствовать различное число элементов содержания и представления свидетельств, при этом их больше в англоязычном оригинале стандарта. Как и для процессов, для получения точного представления о структуре ресурсов семейства полезно составить таблицу, представляющую разбиение свидетельств оценки семейства по компонентам и элементам. Действия, касающиеся необходимых усиления и добавлений, используя англоязычный вариант, выполняются аналогично как для входящих, так и для исходящих ресурсов.

При декомпозиции выходных и внутренних ресурсов приходится в значительно большей степени использовать ОМО, поскольку ОК содержат недостаточно информации по содержанию и форме представления результатов деятельности оценщика. При этом следует учитывать различие в структурировании деятельности оценщика (в основном – терминологическое), имеющее место в этих документах.

В соответствии с подразделом 2.2.5 ОМО, оценщик выносит вердикты относительно выполнения требований ОК [9]. Наименьшая структурная единица ОК, по которой выносится вердикт, – элемент действий оценщика (явный или подразумеваемый). Вердикт выносится как результат выполнения соответствующего действия. В итоге результат оценки формируется в соответствии с подразделом 5.3 части 1 ОК: вердикты по элементам действий объединяются в вердикты по компонентам, затем – в вердикты по классам, и, наконец, по ОО в целом. Все вердикты сначала являются неокончательными и сопровождаются различного рода обоснованиями, образуя «Заключение». Вся совокупность заключений по выполненным действиям оценщика представляет собой «Технические результаты оценки», которые на завершающем этапе оценки преобразуются в два документа: «Сообщения о проблемах» и «Технический отчет об оценке» (при этом неокончательные вердикты преобразуются в окончательные).

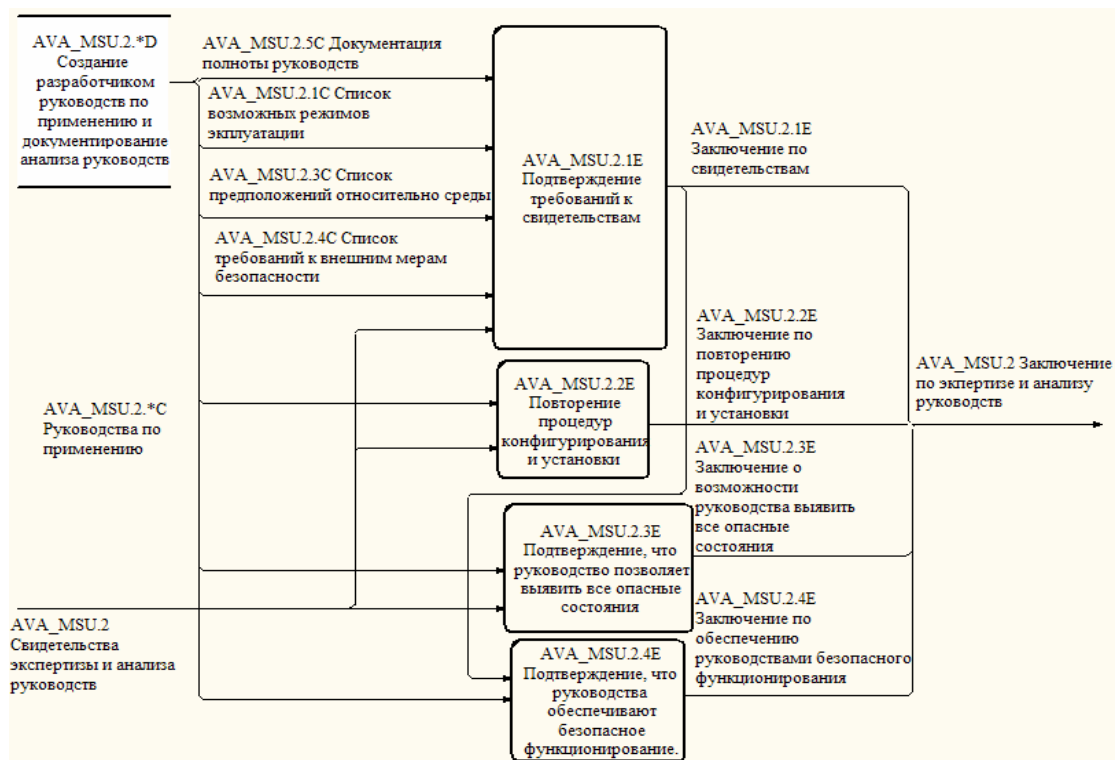


Рис. 1. Диаграмма детализации процесса «Экспертиза и анализ руководств (AVA_MSU.2)»

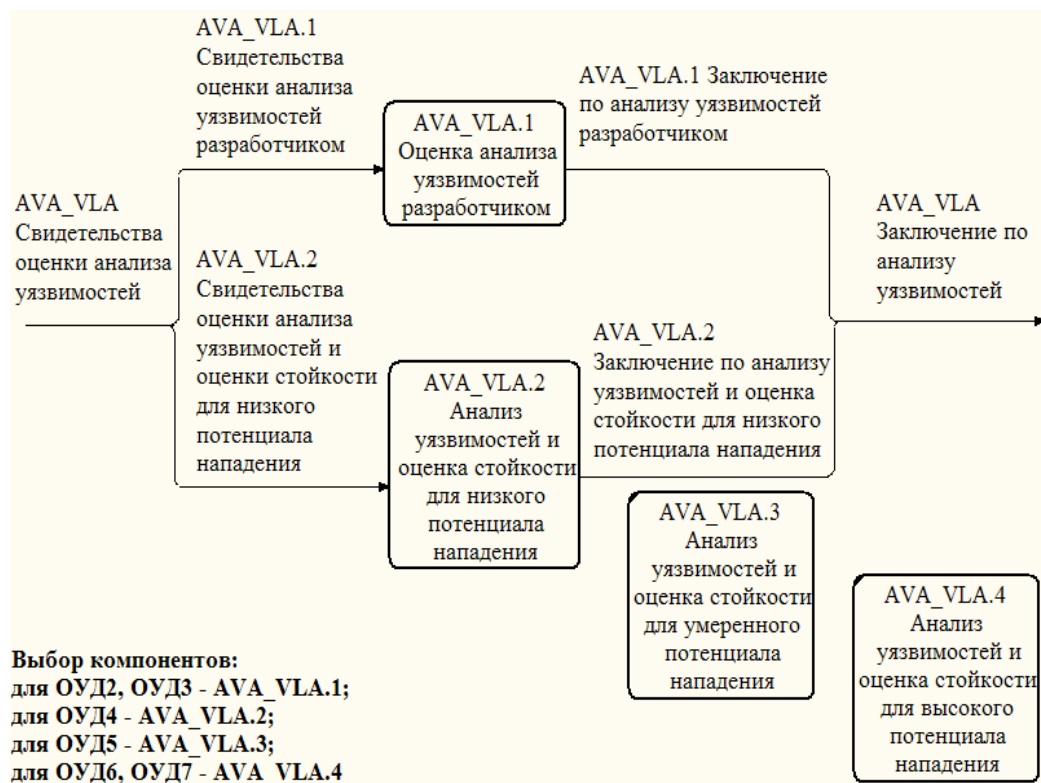


Рис. 2. Диаграмма детализации процесса "Оценка анализа уязвимостей (AVA_VLA)"

Таким образом, формируется следующий алгоритм детализации выходных ресурсов. Заключение по классу доверия ОК (виду деятельности ОМО) есть объединение (join) заключений по семействам доверия ОК (подвидам деятельности ОМО). Заключе-

ние по семейству совпадает с заключением по одному из компонентов ОК, а именно по тому компоненту, который предусматривается согласованным ОУД. Это должно быть явно отмечено в определении ресурса. Заключение по компоненту ОК (подвиду деятельности ОМО) есть объединение заключений по элементам ОК, а они, в свою очередь, формируются по описанию элементов действий оценщика в ОМО. Их именование производится по аналогии с именованием процессов. Отличие в детализации различных ресурсов заключается лишь в сложности декомпозиции, возрастающей благодаря наличию усиления компонентов, а также в том, что некоторые выходные ресурсы одних элементов становятся входными ресурсами других, т.е. являются внутренними для диаграммы детализации компонента. Результат описанных выше действий по детализации можно рассмотреть на рис. 1.

После вышеописанных действий детализация действий оценщика заканчивается, компоненты по определенным ОУД, не описанные в стандарте, не детализируются, и проведение оценки для таких ОУД по стандарту ОК невозможно. Далее объединение ресурсов проводится аналогичным образом для диаграмм более высоких уровней. Подобное объединение ресурсов продемонстрировано на рис. 2.

Заключение

Опыт использования ОК говорит о том, что применение методологии ОК способствует существенному повышению качества оценки и разработки продуктов и систем ИТ. Таким образом, оценка защищенности ИТ по ОК является перспективным направлением, а проблемы, связанные с использованием ОК, очень актуальными. При этом решению многих задач способствует представление ОК в виде формальных моделей. Как правило, функциональная модель является основной и наиболее применяемой на практике. Функциональная модель ОК может быть построена с помощью методики структурного анализа и проектирования систем SADT. Данная методика изначально не может должным образом отобразить модель ОК, поэтому для получения методики функционального моделирования ОК в методику SADT необходимо внести некоторые коррективы. В настоящей работе представлена методика построения функциональной модели деятельности оценщика и разработчика по оценке защищенности ИТ на основе методики SADT и нотации DFD, скорректированная с учетом специфики ОК.

Литература

1. Prieto-Diaz, R. The Common Criteria Evaluation Process. Process Explanation, Shortcomings, and Research Opportunities // Commonwealth Information Security Center Technical Report CISC-TR-2002-03, December 2002. – CISC, James Madison University, USA.
2. Любимов А.В. Функциональная структура общих критериев оценки безопасности информационных технологий // Труды 9-й н.-т. конф. «Теория и технология программирования и защиты информации. Применение вычислительной техники». – Санкт-Петербург, 18 мая 2005 г. – С. 20–24.
3. Зайцев О.Е. Базовые параметры формальных моделей оценки защищенности ИТ по «Общим критериям» // Научно-технический вестник СПбГУ ИТМО. – 2007. – Выпуск 39. Исследования в области информационных технологий. – С. 20 – 26.
4. Зайцев О.Е., Любимов А.В., Суханов А.В. Подходы к структурному моделированию основных компонентов безопасности ИТ «Общих критериев» // Труды 11-й н.-т. конф. «Теория и технология программирования и защиты информации. Применение вычислительной техники». – Санкт-Петербург, 18 мая 2007 г. – С. 56–60.
5. Любимов А.В. Структурное моделирование стандартов информационной безопасности // V Санкт-Петербургская межрегиональная конференция «Информационная

- безопасность регионов России (ИБРР-2007)». – Санкт-Петербург, 23–25 октября 2007 г. – Материалы конференции. – СПб, 2007. – С. 57–58.
6. Common Criteria for Information Technology Security Evaluation. Version 2.2. Revision 256. Part 1: Introduction and general model. – January 2004.
 7. Common Criteria for Information Technology Security Evaluation. Version 2.2. Revision 256. Part 2: Security functional requirements. – January 2004.
 8. Common Criteria for Information Technology Security Evaluation. Version 2.2. Revision 256. Part 3: Security Assurance Requirements. – January 2004.
 9. РД Безопасность информационных технологий. Общая методология оценки безопасности информационных технологий (проект). – ФСТЭК России, 2005.

Зайцев Олег Евгеньевич

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, zoe83@mail.ru

Любимов Александр Вилевич

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, a_v_l@inbox.ru

УДК 004.75, 004.772, 004.62

АНАЛИЗ ФАКТОРОВ, ВЛИЯЮЩИХ НА КАЧЕСТВЕННЫЕ И КОЛИЧЕСТВЕННЫЕ ПОКАЗАТЕЛИ ФУНКЦИОНИРОВАНИЯ СИСТЕМ РАСПРЕДЕЛЕННОГО ХРАНИЛИЩА ДАННЫХ

Н.М. Лукьянов, В.В. Кириллов

В работе проводится исследование распределенных хранилищ данных, анализ качественных и количественных показателей их функционирования. Результатом исследования является функциональная модель системы распределенного хранилища данных, построенная для работы в составе Интернет-сервисов.

Ключевые слова: хранилища, распределенные, показатели, Интернет, сервис.

Введение

Раньше создание распределенной системы хранения данных было специфической задачей, необходимость которой была не очевидна. Сегодня, на совершенно ином техническом уровне, необходимость в распределенных системах хранения очевидна. В сети Интернет широкое распространение получили различные сервисы, предоставляющие услуги размещения, накопления и обмена различной информации, такой как фотографии, видеоматериалы, текстовые документы, электронная почта и т.д. Эти сервисы так или иначе используют в своей работе хранилища данных, которые имеют оригинальные характеристики и свойства. Каждая подобная система имеет свою архитектуру и решает свою конкретную задачу.

Данное исследование направлено на построение и анализ распределенного хранилища данных, предназначенного для работы с интерактивными веб-сервисами.

Постановка задачи

Долгое время распределенные файловые системы использовались в основном в кластерных системах, которые занимались сбором и обработкой большого количества информации, возникающей при решении таких задач, как имитация физических процессов, анализ радиосигналов, генетический анализ, расчет компьютерной графики, финансовое моделирование и т.д. [1].

В области Интернет-приложений вопрос о применении распределенных систем хранения данных для он-лайн сервисов стал рассматриваться относительно недавно, примерно с 2001 г. Сейчас такие системы используются для электронной коммерции, для мультимедийных он-лайн сервисов и самых посещаемых, по данным статистики alexa.com, на сегодняшний день Интернет-сервисов – социальных сетей. В большинстве случаев архитектура и принципы построения таких систем держатся в секрете, так как являются коммерческой тайной.

В данном исследовании распределенное хранилище данных будет рассматриваться в качестве составного элемента в архитектуре крупной Интернет-системы, основная задача которой – предоставление большого количества пользовательских сервисов и агрегация больших объемов разнородных данных. Задачей же распределенного файлового хранилища как составного элемента централизованной системы является хранение программ и данных и предоставление доступа к ним по мере необходимости со стороны сетевых пользовательских сервисов.

Исходя из 1 000 000 планируемых пользователей за первый год работы системы, даже без учета резервирования данных суммарный объем файлового хранилища должен быть около 225 Тб. Для выбора наиболее подходящей из существующих или создания новой распределенной системы хранения необходимо определить, проанализировать и сравнить целый ряд характеристик. Каждая система может обладать всеми или частью этих характеристик, что и создает основу для сравнения различных архитектур между собой. В результате исследования необходимо получить функциональную модель системы с учетом следующих требований [2]:

- открытая архитектура,
- прозрачность размещения файлов,
- независимость размещения файлов,
- мобильность клиента,
- мобильность файлов,
- устойчивость к сбоям,
- расширяемость.

Существующие системы

За довольно большой срок существования такого направления, как распределенные вычисления, было разработано большое число разнообразных файловых систем с различными архитектурами, назначением и уникальными свойствами. Именно поэтому классификация и анализ существующих архитектур распределенных хранилищ позволит выявить их сильные и слабые стороны. Эти знания, с учетом специфики и назначения проектируемой системы, позволят построить ее по описанным требованиям.

По данным энциклопедии «Википедия», насчитывается около сотни распределенных файловых систем различного назначения и архитектуры, которые, впрочем, можно разделить на 3 группы – параллельные файловые системы, симметричные файловые системы с разделяемым диском, гибридные файловые системы с API интерфейсом.

Параллельная файловая система обладает большой производительностью, масштабируемостью и является наиболее успешным примером файловой системы, которая используется в вычислительных кластерах [10]. Система состоит из 3-х компонент:

- сервер метаданных, который отвечает за структуру директорий, а также указывает на положение «объектов», составляющих файл;
- сервер хранения;
- клиент, который находит или загружает необходимый ему файл.

Симметричные файловые системы с доступом к устройству хранения на блочном уровне характеризуются отсутствием функционального разделения узлов. В отличие от

параллельных файловых систем, в симметричных нет управляющего сервера метаданных. Примером симметричной файловой системы может служить Global File System (GFS) от компании RedHat, а также GlusterFS.

Параллельные и симметричные файловые системы являются POSIX-совместимыми и широко используются для различных задач – распределенных вычислений, моделирования, производственных задач [11]. Инженеры Google, Yahoo, разработчик Интернет-дневника LiveJournal и другие пошли несколько иным путем. Они разработали гибридную систему распределенного хранения данных с API доступом, т.е. не файловую систему в классическом понимании. Благодаря наличию API очень легко интегрировать такие файловые системы с любыми Интернет-системами. Гибридные системы построены по принципу, схожему с параллельными файловыми системами, где есть сервер метаданных и серверы хранилища.

Надежность

Надежность систем обработки данных, их способность ежесекундно обеспечивать пользователей оперативной и достоверной информацией – одно из важнейших условий эффективной работы. Одним из решений этой проблемы является кластеризация серверных систем, за счет которой поддерживается высокий уровень готовности (high availability).

Построение отказоустойчивых кластеров предполагает использование в качестве узловых машин высоконадежных серверов с двукратным, а иногда и многократным, дублированием всех основных блоков и компонентов [3]. Так, например, в параллельной файловой системе Lustre, в симметричной файловой системе GFS и GlusterFS надежность данных, хранимых на узлах, практически полностью должна обеспечиваться внутренними средствами резервирования самих узлов, что требует использования дорегостоящих компонентов – как узлов, так и средств коммуникации между ними.

Возможно, при реализации хорошо финансируемого проекта использование промышленных решений могло бы иметь ряд преимуществ. По данным сайтов Mendax.com и ebay.com, хранилище данных объемом 10 Пб можно приобрести менее чем за 10 миллионов долларов, значит, по приблизительным оценкам, хранилище объемом 1 Пб будет стоить примерно 1 миллион долларов.

Противоположный подход, привнесенный из области вычислительных кластерных систем, состоит в том, что сама кластерная архитектура признается единственным необходимым и достаточным условием для обеспечения высокой готовности систем обработки данных, а аппаратная база может не иметь высокой надежности, т.е. может быть представлена обычными бытовыми компонентами, но иметь возможность оперативной замены. Второй подход используют гибридные файловые системы с API GoogleFS, Nadoop и MogileFS [4]. В их окружении нет промышленных решений хранения и обработки данных, используются обычные, предназначенные для бытового использования устройства. Резервирование и обработка данных реализованы на программном уровне.

Для подтверждения рациональности такого подхода инженеры Google сравнили стоимость комплекта 88 серверов, которые суммарно давали 176 процессоров, 176 Гб ОЗУ и 7 Тб дискового пространства, со стоимостью многопроцессорного сервера из 8 процессоров, 64 Гб ОЗУ и 8 Тб дискового пространства. Несмотря на то, что многопроцессорный сервер содержит на 168 процессоров меньше, имеет в 2 раза меньший объем ОЗУ, его стоимость почти в 3 раза выше стоимости всех 88 серверов [5].

Использование бытовых серверов, а также продуманной архитектуры и программного обеспечения позволяет добиваться требуемых показателей, используя при этом значительно меньшие финансовые ресурсы.

Резервирование данных

Исходя из определенных ранее объемов файлового хранилища (225 Тб), общее количество дисков в проектируемой системе должно быть не менее 900 дисков по 250 Гб каждый. Для обеспечения минимальной сохранности данных рассмотрим возможность использования RAID уровня 5. Данные записываются последовательно на все диски, кроме одного, на который записывается контрольная сумма данных остальных дисков. Для такой избыточности будет необходимо еще 100 дисков.

Ожидаемое время отказа бытового жесткого диска – 1000 дней. Время восстановления (копирования) данных при 7% скорости диска займет 1 день (250 Гб на скорости 3,5 МБ/с). Ожидаемое число вышедших из строя дисков при таких условиях будет равно $1000 \times (1/1000) = 1$ диск в день. Если хранилище построено на избыточных массивах RAID 5 из 10 дисков, то вероятность выхода из строя второго диска во время восстановления массива будет $9 \times (1/1000) \approx 0,01$. Вероятность выхода из строя сразу 2 жестких дисков, в результате которого потеряются все данные массива, можно рассчитать как произведение вероятностей этих двух событий, т.е. в течение 100 дней может произойти выход из строя сразу двух жестких дисков массива, что приведет к полной потере данных [6].

Во избежание полной потери данных в результате выхода из строя нескольких жестких дисков одного избыточного массива необходимо принять меры, уменьшающие вероятность аварии. Можно, например, использовать алгоритм подсчета двойной контрольной суммы RAID 6, который, впрочем, не очень популярен и недоступен в большинстве аппаратных RAID контроллерах. Для наиболее эффективного резервирования данных, хотя и с 50% избыточностью, рекомендуется использовать RAID 10. Диски массива объединяются парами в «зеркала» (RAID 1), а затем все эти зеркальные пары, в свою очередь, объединяются в общий массив с чередованием (RAID 0). В результате от «зеркалирования» наследуется надежность, от «объединения с чередованием» – производительность как на чтение, так и на запись [7].

Масштабирование

Учитывая специфику работы Интернет-систем, всегда необходимо быть готовым к масштабному расширению как вычислительных, так и накопительных мощностей системы. В этом случае возможны два варианта масштабирования – горизонтальное и вертикальное. Увеличение мощности узлов называется вертикальным масштабированием и позволяет временно улучшить производительность системы. Однако рано или поздно настанет такой момент, когда наращивать мощность узлов станет невозможным [8]. С учетом представленных компанией Google отчетов о соотношении цены и производительности различных платформ становится очевидным, что вертикальное масштабирование – весьма затратное и неэффективное мероприятие [5].

Горизонтальное масштабирование предоставляет возможность добавлять в систему дополнительные компьютеры и распределять работу между ними. Таким образом, производительность новой системы в целом выходит за пределы предыдущей. Горизонтальное масштабирование с возможностью добавления машин в кластер не только устраняет узкие места при обработке электронных операций, но и одновременно предоставляет дополнительную гибкость для дальнейшего роста и адаптации архитектуры под конкретные требования проекта [9].

Параллельная файловая система Lustre изначально создавалась с учетом требований к горизонтальному масштабированию. Благодаря разделению ролей хранения и обработки данных и наличию сервера метаданных становится возможным легкое расширение системы за счет добавления новых узлов хранения. Горизонтальному масштаби-

рованию хорошо поддаются гибридные файловые системы с API GoogleFS и Hadoop. MogileFS, несмотря на ориентированность к расширению системы с помощью добавления узлов, очень часто имеет проблемы, связанные с недостаточно хорошо протестированной программной архитектурой. Симметричные файловые системы GFS и GlusterFS достаточно плохо показали свои возможности динамического расширения. Добавление узлов возможно только с условием остановки работы системы целиком, что неприемлемо с учетом необходимости обеспечения круглосуточного, бесперебойного доступа к данным. Также работа всего комплекса возможна только в пределах одного центра хранения данных, без географической удаленности узлов хранения друг от друга.

Производительность

Производительность системы в целом складывается из производительности отдельных узлов и компонентов. Различные архитектуры предназначены для различных условий эксплуатации, частоты запросов и видов данных. Однако, чтобы иметь единую шкалу оценки производительности, тестирование проводилось в одинаковых условиях и с одинаковыми типами данных для всех систем.

Для проведения испытаний различных файловых систем был собран тестовый стенд, состоящий из следующего оборудования:

- 4 файловых сервера Intel 2x Xeon, 2GB ОЗУ, 4 SATA HDD 250 Гб (использовались 2), ОС CentOS Linux 5;
- 1 сервер метаданных (для Lustre) Intel Xeon, 1 GB ОЗУ, 1 SATA 160 Гб, ОС CentOS Linux 5;
- 4 клиента Intel P4, 512GB ОЗУ, 1 SATA 40 Гб, ОС CentOS Linux 5.

Инструментом тестирования была свободно распространяемая программа Bonnie++. Тесты проводились на разном количестве серверов и клиентов. Результаты тестирования отражены в табл. 1, 2.

Количество серверов / клиентов	Скоростные показатели			
	Чтение (КБ/с)			
	NFS	Lustre	GlusterFS	GFS
1/1	49510	47753	54803	55741
2/1	49510	48783	46269	78857
4/1	49510	93473	95642	98224
4/4	35422	278265	261258	320282

Таблица 1. Зависимость пропускной способности файловой системы на чтение от количества серверов и клиентов

Количество серверов / клиентов	Скоростные показатели			
	Запись (КБ/с)			
	NFS	Lustre	GlusterFS	GFS
1/1	30419	35228	41763	45452
2/1	30419	45804	46010	67389
4/1	30419	85567	87674	88873
4/4	24369	210352	195473	298441

Таблица 2. Зависимость пропускной способности файловой системы на запись от количества серверов и клиентов

При доступе 1 клиента к 1 серверу все файловые системы показывают одинаковую производительность. Лишь GlusterFS и GFS немного превосходят остальные файловые системы. При доступе 1 клиента к 2 серверам наибольшую производительность показывает GFS – около 78 МБ/с при чтении и 67 МБ/с при записи. Отставание Lustre можно объяснить дополнительными вычислительными затратами, необходимыми для доступа к метаданным.

Использование 4 серверов данных существенно увеличивает скорость доступа к файлам. Здесь наибольшую скорость показывает GFS – около 320 МБ/с на чтение и 298 МБ/с на запись. На втором месте Lustre – 278 МБ/с на чтение и 210 МБ/с на запись. На третьем месте GlusterFS – 261 МБ/с на чтение и 195 МБ/с на запись.

Готовность к работе

В качестве еще одного критерия выбора распределенной системы хранения данных можно выделить такой фактор, как *готовность к работе* или *качество работы* системы. Большинство систем, принимающих участие в обзоре, не могут обеспечить высокий уровень надежности и готовности к работе. Так, файловая система MogileFS имеет слабую поддержку со стороны разработчиков, не имеет библиотеки взаимодействия с популярными языками программирования, такими как, например, PHP, и проявляет нестабильность в работе. Очень часто в системе появляются лишние, не удаленные копии файлов. Файловые системы GlusterFS и Hadoop также имеют проблемы с обеспечением стабильной работы системы. В почтовых рассылках и службе технической поддержки часто появляются сообщения о нестабильной работе или проблемах с чтением/записью данных.

Модель системы

Подводя итоги всестороннего обзора и изучения существующих распределенных систем, обратим внимание на сводную таблицу характеристик шести файловых систем (табл. 3). В первом столбце представлена самая распространенная сетевая файловая система NFS, которая служит лишь для иллюстрации отличия обычных файловых систем от систем распределенного хранения данных. По таблице можно увидеть, что наиболее приближенными, обеспечивающими специфические требования работы с Web-сервисами, системами хранения данных, являются Hadoop и MogileFS. На рисунке схематично представлена распределенная система хранения, разработанная с учетом архитектур этих двух систем, а также с учетом проведенных выше исследований.

Свойства	Файловые системы					
	NFS	Lustre	GlusterFS	GFS	Hadoop	MogileFS
Работа в составе распределенной системы	Да	Да	Да	Да	Да	Да
Наличие головного узла метаданных	Нет	Да	Нет	Нет	Да	Да
Необходимость резервирования головного узла	Нет	Да	Нет	Нет	Да	Да
Posix-совместимая система	Да	Да	Да	Да	Нет (через API)	Нет (через API)
Прозрачность доступа к данным	Нет	Да	Да	Да	Да	Да
Межузловое резервирование данных	Нет	Зеркалирование	Зеркалирование	Зеркалирование	Многократное	Многократное
Необходимость внутриузлового резервирования	Да	Да	Да	Да	Нет	Нет
Работа в WAN сетях	Нет	Да	Нет	Нет	Да	Да
Горизонтальное масштабирование	0	5	3	3	5	4
Производительность	3	4	5	5	4	2
Стабильность работы	5	5	3	4	3	3

Таблица 3. Характеристики файловых систем

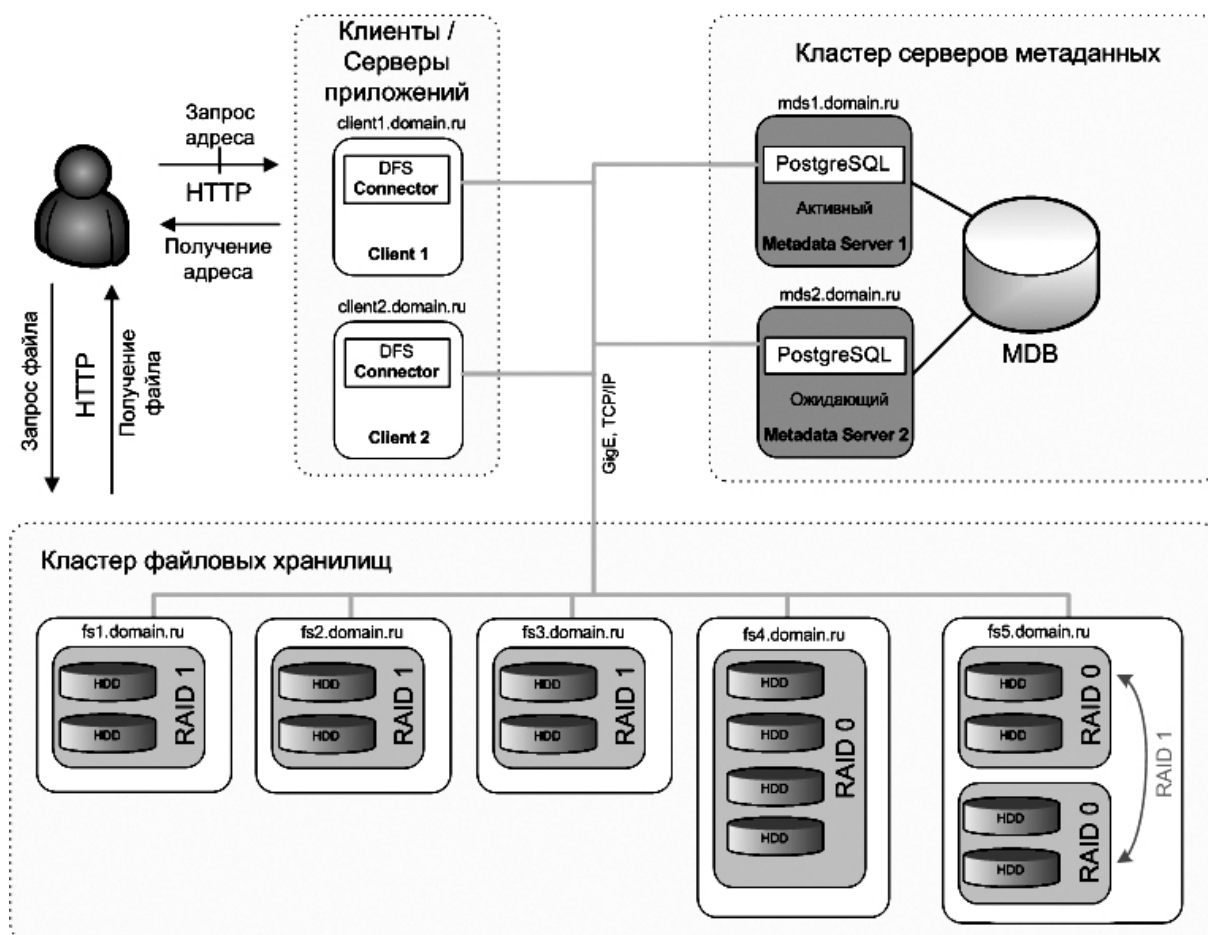


Рисунок. Модель распределенной системы хранения данных

Для совместной работы множества узлов, распределения нагрузки на узлы и контроля целостности системы необходимо присутствие сервера метаданных, содержащего в себе информационную карту распределения файлов по узлам хранения. Отсутствие такого компонента в системе не сможет обеспечить требуемой расширяемости и управляемости, что косвенно подтвердилось в процессе эксплуатации системы GFS без сервера метаданных.

Количество серверов метаданных должно быть более 1, чтобы данный компонент не стал единой точкой отказа системы. В процессе построения системы необходимо учитывать время переключения между основным и запасным сервером метаданных, а также нагрузку на сервер метаданных, которая возникает в процессе работы клиентов с системой хранения.

Минимальной единицей хранения объекта в системе является целый файл, а не его фрагменты. Такое решение увеличит надежность системы, так как в случае полного краха серверов метаданных можно без их участия восстановить файлы, хранимые на серверах хранения.

Следующим элементом системы будет сервер хранения данных, который будет непосредственно взаимодействовать с пользователем, предоставляя доступ к файлам. Внутреннее резервирование данных узла проводится не аппаратными средствами, а программными аналогами, обеспечивающими зеркалирование данных. В дальнейшем будут проведены испытания наиболее эффективного способа резервирования и ускорения чтения/записи данных на узлах хранения.

Средой передачи данных служит Gigabit Ethernet, а протоколом – TCP/IP. Такое решение существенно удешевляет всю систему, не ухудшая ее пропускную способность, позволяя географически перемещать узлы хранения и серверы метаданных.

Взаимодействие пользователя с системой хранения происходит по следующей схеме. Пользователь посылает запрос серверам приложений на получение конкретного файла. Серверы приложений, обращаясь к серверу метаданных, получают ответ о местонахождении требуемого файла и передают его пользователю. В момент формирования ответа сервер метаданных анализирует текущую доступность серверов, содержащих необходимый файл, выбирает менее нагруженный сервер и передает ссылку на этот сервер. Пользователь получает прямую ссылку на требуемый файл и обращается к конкретному серверу хранения.

Загрузка файла в распределенную систему хранения происходит по несколько иному сценарию. Пользователь передает загружаемый файл серверу приложений, который помещается в каталог распределителя. Сервер приложений обращается за списком 3 серверов данных, готовых принять файл. Сервер метаданных на основе информации о доступности, текущей загрузке и объема доступного пространства на серверах выдает требуемый список. Происходит загрузка файла на 3 свободных узла распределенного хранилища. После успешной загрузки информация о местоположении 3 копий файла сохраняется на сервере метаданных.

Обладая тремя копиями файла, можно производить балансировку нагрузки при большом числе запросов к узлам хранения данных, а также обеспечивать сохранность данных без использования аппаратных средств резервирования.

Заключение

Основной задачей исследования было определение и анализ основных факторов, влияющих на качественные и количественные показатели функционирования систем хранения данных. Для выполнения этой задачи были сформулированы основные требования к системе, рассмотрены наиболее распространенные методы построения распределенных систем, просчитаны необходимые объем и структура хранилища, вероятность выхода из строя носителей информации в процессе их эксплуатации, а также рассмотрены способы резервирования дискового пространства системы.

По результатам исследования можно сделать вывод, что ни один из рассмотренных в данной статье принципов построения распределенных хранилищ не смог без существенных доработок обеспечить требуемые характеристики. Однако в результате изучения основных характеристик подобных систем были выявлены сильные и слабые стороны архитектуры, определены ключевые параметры системы, которые могли бы повлиять на работу всего комплекса. Эти параметры необходимо учесть при построении функциональной модели новой распределенной системы хранения данных.

На следующем этапе исследования на основе выработанных требований и предложенной архитектуры системы планируется разработать действующий прототип системы с базовой функциональностью, а также провести разноплановое тестирование для подтверждения правильности выбранной архитектуры и логики работы системы.

Литература

1. Качуров В. Современные распределенные файловые системы для Linux: Основные сведения (перевод) // Компьютерра. – 2002. – № 9.
2. Шнитман В., Кузнецов С. Аппаратно-программные платформы корпоративных информационных систем // Информационно-аналитические материалы Центра информационных технологий. – Режим доступа: <http://www.citmgu.ru>, своб.
3. Николов А. Кластерные системы высокой готовности // BYTE:Россия. – 2005. – № 8.
4. Hoff T. Google Architecture. Материалы сайта HighScalability.com, 2008. – Режим доступа: <http://highscalability.com/google-architecture>, своб.

5. Hammonds K. H. How Google Grows...and Grows...and Grows // Журнал FastCompany. – 2003. – Режим доступа: <http://www.fastcompany.com/magazine/69/google.html>, своб.
6. Инструкция пользователя Lustre 1.6, 2007. – Режим доступа: http://manual.lustre.org/manual/LustreManual16_HTML/TOC.html, своб.
7. Егоров А. RAID 0, RAID 1, RAID 5, RAID 10, или что такое уровни RAID? Материалы компании «ТИМ», 2005. – Режим доступа: <http://www.timcompany.ru/article4.html>, своб.
8. Фильчаков А. Кластеры и кластеризация // КомпьютерПресс. – 2000. – № 10.
9. Пресс-релиз Red Hat Enterprise Linux 5 Оптимальное решение для финансовых организаций. Материалы компании Red Hat, 2007. – Режим доступа: <http://www.redhat.com>, своб.
10. Таненбаум Э., ван Стеен М. Распределенные системы. Принципы и парадигмы. – СПб: Питер, 2003. – 877 с.
11. Вахалия Ю. UNIX изнутри. – СПб: Питер, 2003. – 844 с.

Лукьянов Николай Михайлович

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, nikolay@lukianov.net, nikolay.lukianov@gmail.com

Кириллов Владимир Васильевич

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, профессор, kirillovv@gmail.com

УДК 681.3.06

МЕТОД РАЗБОРА ИСХОДНЫХ ТЕКСТОВ ПО УПРОЩЕННОЙ ГРАММАТИКЕ ЯЗЫКА

Д.Б. Арефьев, В.Л. Верещагин, А.И. Галанов, А.А. Молдовян

Статья рассматривает метод разбора исходных текстов, основанный на упрощенном множестве порождающих правил грамматики. Приводится концепция метода и сведения об эффективности применения на практике.

Ключевые слова: защита, информация, уязвимости, анализ, грамматика, исходный текст.

Введение

Опыты, приведенные в данной статье, были выполнены в рамках создания специального комплекса, предназначенного для контроля уязвимостей программ. Данный комплекс работает с исходными текстами и создан с целью повышения эффективности существующих методов контроля. Одни методы предполагают прямой поиск заданных выражений, другие используют отладочную информацию, третьи справедливо считают, что для корректного исследования программы необходимо выполнить полный лексический и синтаксический анализы. При разработке нового комплекса был учтен опыт практического применения этих методов, и в данной статье предлагается исследование метода разбора по упрощенной грамматике языка. Для эксперимента был выбран язык ANSI C (99) по нескольким причинам: язык широко применяется в промышленности, достаточно прост для разбора и обладает низкой защищенностью.

Программный комплекс предоставляет эксперту сведения о безопасности программы на основе данных, полученных в результате разбора исходных текстов по правилам грамматики языка. Формирование полной грамматики является достаточно трудоемким занятием. С другой стороны, ни одна из исследованных программ не использовала полную мощность языка, что позволило использовать только такое количество

грамматических правил, которое необходимо и достаточно для конкретного проекта. В работе рассматривается методика, которая позволяет корректно определить количество таких необходимых и достаточных правил, при условии, что дальнейший анализ ориентирован на оценку безопасности программного продукта.

Постановка задачи

Как уже было сказано, поиск программных закладок начинается со сбора информации об исследуемой программе. Если взять на исследование какой-либо проект средней величины, состоящий из нескольких десятков исходных файлов, то среди них обязательно окажется какое-то число избыточных. До разбора файлов проекта целесообразно выделить только те файлы, которые действительно компилируются и собираются. Для этого применяется способ [1], основанный на принципе маркировки каждого исходного файла проекта. После того, как в каждый файл будет вставлен уникальный маркер, проект компилируется, и в сформированных исполняемых файлах отыскиваются маркеры тех файлов, из которых он был собран. Этот простой способ позволяет составить список файлов, по которым необходимо собрать информацию о функциях, линейных участках, связях между ними и т.п.

В основном требования к тому, какая именно информация должна быть собрана, описаны в руководящем документе [2]. К сожалению, как показывает практика, для эффективного контроля программных закладок выполнение этих требований оказывается недостаточным, и необходимо вводить дополнительные проверки (например, методом выявления пассивных программных закладок), для которых необходима дополнительная информация.

Общие требования к информации, которая должна быть сформирована для каждого исполняемого файла после разбора, можно представить следующим образом:

- список исходных файлов;
- сведения о функциях, их связях с исходными файлами, переменными, другими функциями и линейными участками;
- сведения о начале функции в исходном файле и ее окончании;
- сведения о начале линейного участка и его окончании;
- сведения о переменных;
- сведения об аргументах функций;
- сведения о содержании проверки в условных операторах и операторах цикла;
- сведения о выражении в операторе выбора;
- сведения о передаче управления линейным участком;
- сведения об окончании файла (сдвиг от начала, завершающий оператор);
- сведения о метках для оператора безусловного перехода

и др. Весь список достаточно велик для приведения его здесь, и, видимо в этом нет никакой необходимости, так как при более подробном рассмотрении пришлось бы переписывать полную грамматику языка. Именно такой подход был использован в разработке анализатора, основывающегося на формальных грамматиках [3]. Технически он наиболее совершенен по сравнению с другими подходами, но имеет достаточно трудную реализацию в связи с тем, что для каждого языка (пусть даже такого специфического, как Avenue) необходимо написать полную грамматику, что может занять больше времени, чем даже ручной анализ. Чтобы не идти таким сложным путем, несколько лет назад в анализаторе ST (разработка одного из ведущих российских специалистов по безопасности В.Н. Стрельца) был реализован подход, основывающийся на отладочной информации [4]. Суть этого подхода заключается в следующем: Visual Studio формирует во время компиляции специальный файл, в котором содержится отладочная информация. Этой

информации достаточно для того, чтобы выполнить анализ по 3-му уровню контроля недекларированных возможностей соответственно руководящему документу.

Достаточно простой подход используют английские и американские исследователи Джон Виега (JohnViega), Гари МакГроу (Gary McGraw) и др. В своей статье о статическом сканере безопасности ITS4 [5] они приводят аргументы в пользу упрощенной грамматики языка и доказывают ее эффективность. Сканер безопасности ITS4 предназначен для контроля таких ошибок, как переполнения буфера, утечка памяти, ошибки в работе механизма указателей, ошибки инициализации создаваемых объектов и др. Он успешно находит некорректное использование стандартных функций вроде `strcpy`, `strcut` и `sprintf`, используя предложенную методику. Фактически грамматика языка, используемая сканером, должна содержать правила объявления функций, переменных, указателей и некоторые другие. Этот подход позволяет им безошибочно находить необходимые места текста и анализировать их степень уязвимости.

В наших исследованиях учтен опыт использования формальных грамматик, а также опыт зарубежных специалистов. Вернемся к тому, что перед разбором исходных текстов были выбраны только те файлы, которые действительно входят в проект. Отметим, что все отобранные файлы компилируются. Это обстоятельство позволяет нам сформулировать основополагающее условие опыта – для исходного текста, подлежащего лексическому и синтаксическому анализу, существует такая однозначная контекстно-свободная грамматика, которая порождает язык, являющийся либо подмножеством исходного языка, либо совпадающий с ним. Под исходным языком в нашем опыте понимается стандарт ANSI C (99), но это лишь частный случай, на самом деле им может быть любой существующий на сегодняшний день язык программирования. В таком случае мы можем построить такой синтаксический анализатор, который будет работать на основе только тех структурных правил языка, которые используются в текстах разбираемой программы.

Резюмируя сказанное, можно отметить, что к задачам исследования относится оценка предлагаемого метода анализа, в частности, важно ответить на вопрос: «Возможно ли использовать упрощенную грамматику для анализа исходных текстов?».

Методы исследований

Чтобы оценить метод анализа, необходимо провести эксперимент и интерпретировать полученные результаты применения. Общая концепция анализа основана на синтаксическом разборе исходных текстов, и для того, чтобы получить результаты, необходимо использовать средства автоматизации. В рамках исследования был создан инструмент для анализа, который будет рассмотрен ниже.

Перед тем, как рассмотреть концептуальную схему работы анализатора, целесообразно определить его местоположение в составе специального комплекса контроля уязвимостей и программных закладок. На рис. 1 представлена схема, отображающая основные взаимосвязи модулей комплекса.

Некоторые принципы взаимодействия модулей целесообразно рассмотреть отдельно. Исходные тексты (блок 12) исследуются на наличие избыточности в блоке 9. Те файлы, которые были отобраны как «используемые», участвуют в дальнейшем анализе, а избыточные файлы помечаются в базе данных (БД, блок 10). Файлы попадают на вход модуля разбора исходных текстов (блок 1). Именно методы, применяемые в данном модуле, исследуются в данной статье. Информация о файлах сохраняется после разбора в БД, после чего используется на стадии статического анализа (блок 2). Статический анализ предназначен для исследования исходного текста программы на соответствие стандартам программирования, на наличие дефектов и уязвимостей, в частности, программных закладок. Подготовка к динамическому анализу реализуется в модуле вставки

датчиков (блок 4). Датчики вставляются в каждый линейный участок, информация о которых хранится в БД, заполненной после разбора исходных текстов. Версия исходных текстов проекта со вставленными датчиками называется лабораторной (блок 6). Последовательность сработавших датчиков формирует «трассу» программы (блок 11), которая затем попадает на вход модуля динамического анализа (блок 14). Блок 13 формирует отчеты по результатам статического и динамического анализа (список функций, связей, переменных, и т.п.). В блоках 5, 7, 8 показаны некоторые модули, теоретически входящие в процесс статического анализа, но фактически реализованные как отдельные программные модули. Среди них, соответственно, модуль построителя блок-схем, модуль контроля уязвимостей из БД и модуль обнаружения пассивных программных закладок.

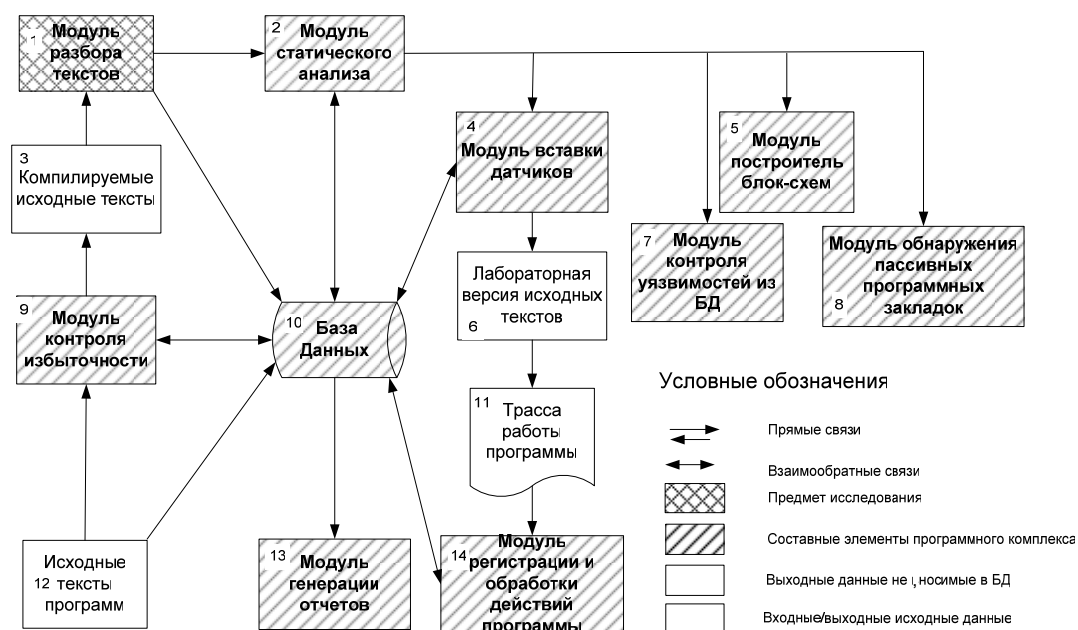


Рис. 1. Упрощенная схема комплекса контроля уязвимостей

Как видно из общего описания схемы, модуль разбора исходных текстов играет основополагающую роль. Алгоритм модуля преобразует исходные тексты к такому виду, который можно использовать для статического и динамического анализов. На концептуальной схеме (рис. 2) показано взаимодействие различных составляющих модуля разбора исходных текстов.

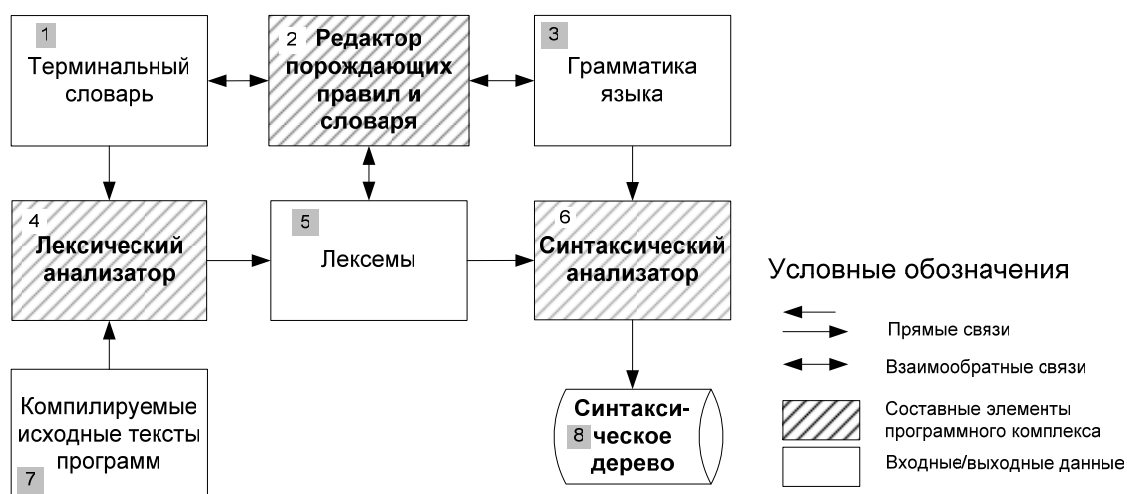


Рис. 2. Концептуальная схема разбора исходных текстов

Исходные тексты (блок 7), которые не являются избыточными и собираются в проект, попадают на вход лексического анализатора (блок 4). В результате лексического разбора исходные тексты представляются в виде множества лексем (блок 5) с соответствующими им атрибутами. Терминальный словарь (блок 1) на стадии лексического анализа способствует выделению лексемы в тексте. Словарь формируется исследователем самостоятельно, либо экспортируется ранее созданный. Инструмент, позволяющий сформировать словарь для какого-либо языка, – «Редактор порождающих правил и словаря» – показан на рис. 2 под номером 2. Данный инструмент позволяет по необходимости расширять или сокращать список терминальных символов. Теоретически необходимость может состоять в прикладном использовании. Если исследуемые исходные тексты написаны на узкоспециализированном языке, например, Avenue (скриптовый язык, используемый для геоинформационных систем), то при определенных условиях может отсутствовать необходимость вводить все терминальные символы, из которых формируется язык. Таким условием, например, может быть небольшой объем исходных текстов, в которых набирается десяток терминальных символов. В таком случае может оказаться целесообразным сформировать подобный список вручную и затем его использовать для создания множества порождающих правил (блок 3).

Множество правил построения предложений языка порождают схему грамматики. Полная схема включает в себя все правила, позволяющие сформировать множество конечных цепочек терминального алфавита (словарь), т.е. конечный язык. Редактор позволяет исследователю самостоятельно сформировать порождающие правила. На рис. 3 показана обобщенная схема работы модуля «Редактор порождающих правил и словаря».

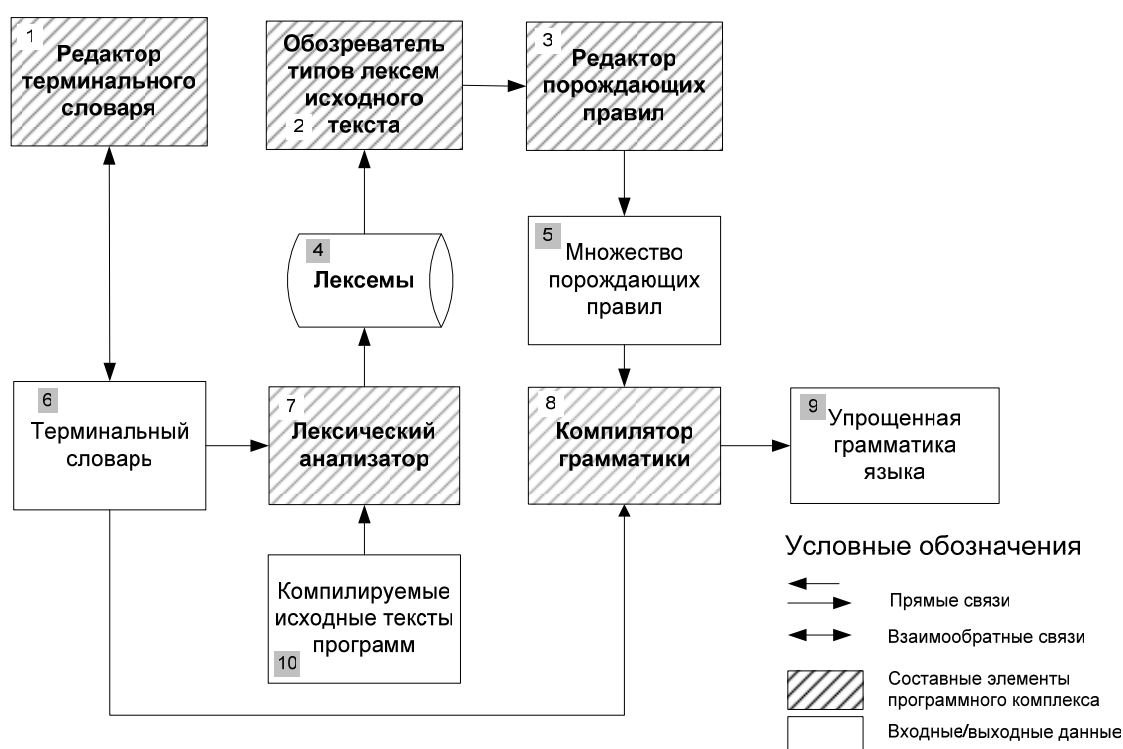


Рис. 3. Схема получения упрощенной грамматики языка

Как видно из схемы, принцип получения упрощенной грамматики несколько отличается от классических представлений о ее формировании. Показан подход не от грамматики к исходным текстам, а от исходных текстов к грамматике. Данный подход заключается в следующем.

- 1) Исследователь определяет язык программирования исходных текстов и формирует терминальный словарь (блок 6). Совокупность терминальных символов, характерных для конкретного языка, образует терминальный словарь. Типы данных, условные операторы, идентификаторы и пр. являются терминальными символами. Словарь может быть импортирован из уже существующего варианта либо создан заново. Чтобы словарь можно было дополнить, изменить или создать, создан специальный инструмент «Редактор терминального словаря» (блок 1).
- 2) Исследователь берет для анализа те файлы, которые успешно компилируются (блок 10). Это гарантирует то, что все цепочки терминальных символов (исходные тексты) сформированы посредством некоторого множества порождающих правил, которое, в свою очередь, является подмножеством правил полной грамматики языка. Допустим, в полной грамматике существует 1000 различных правил вывода цепочек, а в исходных текстах используется мощность языка всего лишь на 30%, т.е. реализовано всего 300 различных цепочек. Если по ним составить выводящие правила, то мы получим упрощенный вариант правил, характерный именно для этих исходных текстов. Данный способ может быть удобен в двух основных случаях: грамматика языка отсутствует, и необходимо формировать ее самого начала; имеется грамматика языка, но устаревшая, требующая доработки. В первом случае упрощенная грамматика позволяет избежать формирования полной схемы, а во втором случае – конфликтов новых правил со старыми правилами.
- 3) Формируются лексемы. Исходные тексты обрабатываются лексическим анализатором (блок 7), в результате чего формируется массив лексем (блок 4). Тип каждой лексемы определяется, и затем все типы выстраиваются в соответствии с последовательностью, описанной в исходных текстах. Например, если в программе есть строка объявления переменной: «int a;», то последовательность типов лексем будет выглядеть так: «Type ID;», где «Type» – это «тип данных», «ID» – последовательность латинских букв, не являющихся служебным словом и удовлетворяющая правилам составления идентификатора. Вся эта информация отображается в окне модуля «Обозреватель типов лексем исходного текста» (блок 2).
- 4) Формируется множество порождающих правил. Порождающее правило формируется исследователем вручную на основе последовательности типов лексем. «Редактор порождающих правил» (блок 3) позволяет создавать и изменять набор правил. Множество порождающих правил (блок 5) формируется по результатам проработки исходных текстов исследователем.
- 5) Формирование конечной грамматики. Модуль «Компилятор грамматики» (блок 8) приводит набранную грамматику к удобному для автоматической обработки виду и сохраняет в отдельный файл, который и формирует упрощенную грамматику языка (блок 8, рис. 3, и блок 3, рис. 2).

Компоненты модуля, выполняющего разбор исходных текстов (блок 1, рис. 1), можно представить в виде двух групп. Первая группа – это компоненты, изменяемые при исследовании нового языка, например, «терминальный словарь», «множество порождающих правил», «массив лексем» и «упрощенная грамматика языка». Фактически это входные данные, которые изменяются вместе с исследуемыми исходными текстами. Вторая группа – это алгоритмы, остающиеся неизменными при новых исследованиях, например, при исследовании программ, написанных на языках, для которых отсутствует сформированный словарь и набор порождающих правил. К таким компонентам относятся программы «Лексический анализатор», «Обозреватель типов лексем исходного текста», «Редактор терминального словаря», «Редактор порождающих правил» и «Компилятор грамматики». В результате получаем инструмент, который позволяет проводить разбор и анализ программ, написанных на различных языках.

Получается, что сложность анализа прямо пропорциональна мощности схемы грамматики, применяемой в данном конкретном проекте.

Эксперимент, проводимый при исследовании методики анализа, заключается в следующем. Цель – определить корректность методики анализа исходных текстов по упрощенной грамматике языка. Файлы с исходными текстами подаются на вход анализатора, над ними производятся лексический, синтаксический и статический анализы, в результате чего в БД сохраняется определенная информация¹. Среди этой информации имеются сведения о функциях, линейных участках, переменных и связях между ними. Критерием правильности выполнения анализа исходных текстов, является соответствие списка функций, линейных участков и переменных действительному состоянию в исходных текстах. Исходные условия состоят в следующем:

1. 1539 исходных файлов, написанных на языке ANSI C (99);
2. все файлы компилируются и собираются в конечные исполняемые файлы;
3. в исходных текстах отсутствуют² правила, не учтенные в грамматике;
4. компилятор «Watcom C 10.6»;
5. среда компиляции – операционная система реального времени «QNX» версии 4.25;
6. среда анализа – MS Windows XP (SP2) + драйвер поддержки файловой системы «ext2».

Методика проверки заключается в следующем. Для всех 1539 файлов производятся синтаксический и статический анализы. В результате формируется БД со списком функций, переменных и линейных участков. Для контроля формируется список файлов, которые будут проверены вручную. Из всех файлов выбираются такие, которые содержат максимальное число не повторяющихся порождающих правил. Из этих файлов формируется группа, которая суммарно содержит все порождающие правила упрощенной грамматики. Для сформированного списка файлов проводится ручной анализ, который сравнивается с результатами статического. В том случае, если результаты совпадают, считается, что к данным исходным текстам метод был применен корректно. Проверенные файлы удаляются из общего списка, и формируется новая группа для проверки. Общее количество таких проверок равно трем. По результатам делается вывод о корректности работы метода анализа на основе упрощенной грамматики языка.

Результаты исследований

Результаты исследования, в частности, проведенного эксперимента, целесообразно разложить на два основных этапа – анализ исходных текстов и выполнение ручного контроля полученных результатов. Результаты, полученные на первом этапе, были записаны в БД в виде таблиц. Концептуальное представление таблиц можно видеть на примере (см. таблицу).

№ эл-та	Имя эл-та	№ исх. текста	№ исп. файла	Начало эл-та, в байтах	Оконч. эл-та, в байтах	Связь с др. эл-ми	Флаг 1	...	Флаг N
0	func()	27	03	1982	2401	1	1	...	0
1	if(a>b)	12	03	1863	1945	82	1	...	1
...	...	...	...	...	...	...	...	...	...
L	i	T	M	0	004	2008	X	...	Y

Таблица. Пример формата представления данных в БД

¹ В соответствии с требованиями РД.

² Для достижения этого условия используется механизм выявления нераспознанных конструкций в инструменте «Редактор порождающих правил» (блок 3, рис. 3)

БД содержит таблицы со сведениями об исполняемых и исходных файлах, о функциях, о линейных участках, переменных и связях между ними, а также необходимую дополнительную информацию.

Ручной анализ показал, что при автоматическом подходе были верно сформированы таблицы функций, переменных и линейных участков, а также связи между ними. На основании этого можно сделать вывод о возможном применении упрощенной грамматики при анализе исходных текстов.

К недостаткам данного подхода проверки можно отнести индукционный подход доказательства. В связи с тем, что полностью исходные тексты проверить вручную не представляется возможным, целесообразно найти эталонный вариант контроля функций, переменных и линейных участков и разработать автоматизированное средство сравнения полученных результатов. В качестве такого эталонного варианта анализа может быть использован, например, компилятор и его отладочная информация.

Заключение

Проведенные исследования позволили сформировать представление о достоинствах и недостатках метода анализа исходных текстов, основанного на использовании упрощенной грамматики языка. Упрощенная грамматика определяется на основе множества правил вывода конечных цепочек языка, характерных только для исследуемых исходных текстов. За счет такого подхода процесс формирования полной грамматики языка упрощается пропорционально мощности схемы грамматики. Исследования показали, что весь метод позволяет корректно обрабатывать исходные тексты и получать сведения о таких конструкциях языка, как переменные, функции и линейные участки. Среди достоинств метода следует отметить:

- гибкость (одними и теми же инструментами можно исследовать программы, написанные на различных языках программирования);
- достаточность (исходный текст разбирается с необходимой для анализа точностью);
- эффективность (результаты исследования превышают менее точные разборы, например, на основе отладочной информации, и не уступают анализу, основанному на применении полной грамматики языка).

К недостаткам подхода следует отнести прикладные качества – метод можно использовать при достаточном уровне знаний, касающихся теории компиляции и программирования. Другой недостаток – высокая трудоемкость формирования порождающих правил по исходным текстам.

Литература

1. Котенко Д.А. Контроль избыточности исходных файлов программ на основе вставки программных датчиков // V Санкт-Петербургская межрегион. конф. «Информационная безопасность регионов России (ИБРР-2007)», Санкт-Петербург, 23–25 октября 2007 г. – СПб: СПОИСУ, 2007. – 153 с.
2. Руководящий документ «Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недеklarированных возможностей» по 3 уровню контроля. – Гостехкомиссия России (в настоящее время ФСТЭК), Москва, 1999 г.
3. Котенко Д.А., Солодяников А.В. Метод использования формальных грамматик языков программирования при проведении сертификационных испытаний на отсутствие недеklarированных возможностей // V Санкт-Петербургская межрегион. конф. «Информационная безопасность регионов России (ИБРР-2007)», Санкт-Петербург, 23–25 октября 2007 г. – СПб: СПОИСУ, 2007. – 153 с.

4. Верещагин В.Л., Солодяников А.В. Анализатор исходного текста, основанный на обработке отладочной информации // V Санкт-Петербургская межрегион. конф. «Информационная безопасность регионов России (ИБРР-2007)», Санкт-Петербург, 23–25 октября 2007 г. – СПб: СПОИСУ, 2007. – 153 с.
5. J. Viega et al.. ITS4: A Static Vulnerability Scanner for C and C++ Code. // Ann. Computer Security Applications Conf. (ACSAC), Applied Computer Security Assoc., 2000.
6. Фомичев В.С. Теория автоматов. – СПб: СПбГЭТУ, 2007. – Режим доступа: http://www.eltech.ru/misc/LGA_2007_FINAL/Allpage/Section6/part_6.1_.html, своб.
7. Ахо А.В., Сети Р., Ульман Д. Компиляторы: принципы, технологии и инструменты. – М: Издательский дом «Вильямс», 2003.

Арефьев Дмитрий Борисович

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, av_dmitry@mail.ru

Верещагин Владимир Леонидович

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, seckey@yandex.ru

Галанов Алексей Игоревич

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, daiverddd@yandex.ru

Молдовян Александр Андреевич

— Научный филиал Федерального государственного унитарного предприятия "НИИ"Вектор" специального центра программных систем "Спектр", директор, доктор технических наук, профессор, ma@cobra.ru

УДК 004.032.2

АВТОМАТИЗИРОВАННАЯ СИСТЕМА СБОРА И ОБРАБОТКИ ДАННЫХ СКВАЖИННЫХ ИЗМЕРЕНИЙ С ПОМОЩЬЮ АКУСТИЧЕСКОГО РАСХОДОМЕРА

Т.Е. Войтюк, А.В. Демин

В статье представлены новые концепции сбора и обработки данных скважинных измерений с использованием акустического расходомера. Автоматизированная система позволяет повысить рентабельность разработки ранее неперспективных месторождений.

Ключевые слова: нефть, добыча, месторождение, акустический расходомер, волоконно-оптический.

Введение

Использование новейших информационных технологий в процессе разработки нефтяных и газовых месторождений России становится первостепенной задачей на пути увеличения и контроля эффективности месторождений. Все большее число нефтедобывающих компаний сталкивается с необходимостью вскрытия нескольких продуктивных зон одной скважиной и, в то же время, обеспечения возможности замера дебитов из каждой отдельной зоны, что связано с поддержанием баланса между существующим подходом к разработке недр, с одной стороны, и экономическими соображениями, с другой. Затраты на строительство дополнительных эксплуатационных скважин на каждом кустовом основании в большинстве случаев превышают стоимость закачивания одной скважины по схеме, позволяющей вести одновременную добычу сразу из нескольких продуктивных горизонтов.

Для решения этой задачи применяют технологию «интеллектуальное месторождение», которая дает возможность оперативно принимать решения по оптимизации режи-

мов работы скважин, пластов и системы разработки месторождения. Эта технология реализует автоматизированную систему многоуровневого сбора и обработки информации.

Представленный программно-аппаратный комплекс сбора и обработки данных скважинных измерений с помощью акустического расходомера является составной частью системы «интеллектуальное месторождение». По данным, получаемым с волоконно-оптических гидрофонов, расположенных в стволе скважины, и хранимым в текстовых файлах в формате CSV (Commit Separator Value), комплекс позволяет производить расчет и отображение временных и спектральных характеристик регистрируемых сигналов, а также анализ с использованием разнообразных цифровых фильтров. Полученные сведения используются для управления режимами работы скважины и оптимизации разработки залежей нефти.

Интеллектуальное месторождение

Интеллектуальное месторождение – это месторождение, на котором имеется автоматизированная система многоуровневого сбора и обработки информации, которая, в свою очередь, служит базой для оперативного принятия решений по оптимизации режимов работы скважин, пластов и, в итоге, системы разработки месторождения. Создание интеллектуальных месторождений является перспективным и экономически целесообразным направлением, которое позволяет повысить рентабельность разработки многопластового месторождения за счет строительства одной «интеллектуальной» скважины вместо нескольких обычных.

Одной из инновационных составляющих системы разработки для низкодебитных многопластовых месторождений является оптимальное объединение различных пластов в один объект, что позволяет резко увеличить средние дебиты скважин, текущие уровни добычи нефти и ввести в разработку ранее нерентабельные месторождения. Однако возникает проблема качественного и постоянного контроля притока из каждого пласта [1]. В связи с ужесточением правил разработки, допускающих одновременную эксплуатацию нескольких пластов единой сеткой скважин только при условии организации достоверного учета и постоянного контроля притока из каждого объекта разработки, только налаженный мониторинг работы каждого пласта позволит создать четкое представление о состоянии месторождения. Контроль пластового и межпластового давления, температуры зон перфорации, определение депрессии на добывающей скважине, газового фактора, состава и расхода флюида позволят создать четкое представление о состоянии месторождения – о степени выработки, зонах остаточных запасов и других параметрах, которые способствуют принятию адекватных решений для достижения максимального коэффициента извлечения нефти (КИН) из каждого объекта разработки.

Создание инструмента постоянного мониторинга количественных и качественных параметров работы каждого объекта в скважине является фундаментальным и наиболее сложным этапом реализации интеллектуального месторождения.

Главными проблемами мониторинга нефтяных скважин в северных широтах являются большие глубины (4 км и более), высокие температуры (свыше 130 °С) и давление (400 атм и более). Такие скважины содержат залежи высоковязких нефтей и природных битумов, которые являются трудноизвлекаемыми. Традиционные методы измерения, с применением электронных средств регистрации давления, температур и расхода (имеющие в составе измерительной части усилительные, преобразующие и др. устройства), не могут эксплуатироваться в таких условиях или имеют малую точность и большую погрешность в показаниях [2]. Создание нетрадиционных измерительных комплексов позволяет решить данную проблему. Одним из перспективных направлений является применение волоконно-оптических измерительных систем, не имеющих электронных преобразователей и активных элементов и слабо зависящих от температу-

ры и давления. Предлагаемая технология дает возможность существенно увеличить экономическую эффективность освоения месторождений, содержащих высоковязкие нефти и битумы.

Основными элементами волоконно-оптической измерительной системы являются [3] когерентный источник света (лазер), фотоприемное устройство (фотодиоды различных конструкций, фотоэлектронные преобразователи), волоконно-оптический кабель и преобразователь измеряемого физического поля в виде чувствительного волоконно-оптического элемента (датчик). Электрический сигнал в светоизлучающем устройстве преобразуется в световой и по волоконно-оптическому кабелю поступает к датчику. Световой сигнал от датчика (в виде отраженного или рассеянного света) поступает в фотоприемник по тому же или специальному кабелю. В фотоприемном устройстве сигнал преобразуется в электрический сигнал, несущий информацию об измеряемом объекте.

Измерительная часть системы

Автоматизированная система сбора и обработки данных с помощью акустического расходомера состоит из 2 частей – скважинной и наземной.

Скважинная часть разработанной оптоволоконной измерительной системы состоит из распределенного датчика температуры, точечного датчика температуры и давления, датчика расхода-состава. Датчики объединены спускаемым в скважину волоконно-оптическим кабелем. С помощью кабеля осуществляется передача результатов измерений физических параметров от датчиков к наземному оптоэлектронному блоку обработки информации для обработки, хранения и отображения сигнала в наземной части.

Распределенный датчик температуры относится к типу устройств, в которых измеряемая температура действует непосредственно на оптическое волокно, так что оно само служит сенсором на всем своем протяжении, а выбор точки измерения определяется интервалом времени между моментом излучения сигнала лазером и моментом прихода отраженного от торца кабеля сигнала. На распределенный датчик температуры возложена функция определения зон термической аномалии по всему стволу скважины с целью определения зон негерметичности колонны и заколонной циркуляции.

Датчики давления и температуры, выполненные на основе волоконно-оптических брэгговских решеток (ВОБР), относятся к типу точечных датчиков. ВОБР представляет собой небольшой участок световода, в котором с заданным шагом чередуются участки с меньшим и большим коэффициентом преломления. Принцип действия ВОБР основан на том, что при распространении по оптическому волокну света в широком диапазоне длин волн отражаются только волны того участка спектра, который соответствует шагу решетки. Если под действием температуры или давления оптическое волокно с нанесенной решеткой Брэгга удлиняется, то меняется шаг решетки и, соответственно, длина отражающегося луча, что и преобразуется в значение температуры и давления. Точность точечных датчиков температуры кратно превосходит точность распределенного датчика, что позволяет с высокой достоверностью оценить относительный приток из каждого интервала, над которым расположен точечный датчик температуры.

Для измерения расхода и состава флюида в зоне притока применен пассивный волоконно-оптический акустический расходомер [4], который обеспечивает регистрацию гидроакустических сигналов, создаваемых различными источниками в скважине. По характеру гидроакустического воздействия, с учетом предварительных калибровок, определяется состав флюида, а также скорость потока. Скорость потока характеризуется параметрами флуктуаций флюида в зоне перфорации. Акустический расходомер обладает уникальными эксплуатационными свойствами: высокая чувствительность в общем акустическом спектре частот (1–100 000 Гц), большой коэффициент усиления (140–180 Дб), равномерная амплитудно-частотная характеристика (0,1–2 Дб), малые габариты и масса,

отсутствие активных приборов (т.е. полная электрическая и магнитная пассивность), работоспособность при высоких температурах (до 200 °С) и давлениях (до 500 атм).

Структурная схема акустического расходомера представлена на рис. 1. Расходомер состоит из 5 блоков: лазерный источник (ЛИ), формирователь модулирующих сигналов (ФМС), волоконный тракт с чувствительным элементом (ЧЭ), фотоприемник (ФП) и детектор фазомодулированных (ФМ) сигналов.

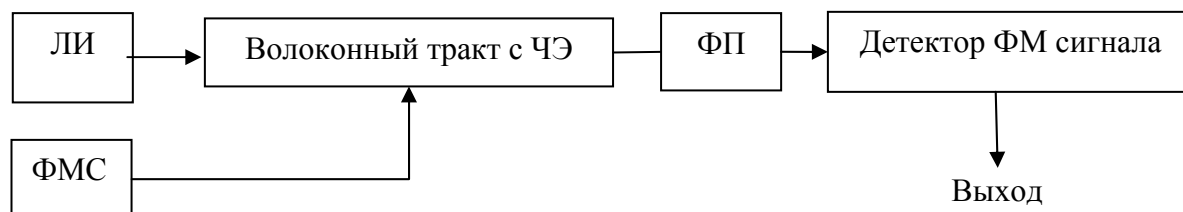


Рис. 1. Схема акустического расходомера

Лазерный источник содержит лазерный диод и драйверы к нему. Лазерный диод представляет собой полупроводниковый лазер с распределенной обратной связью, с длиной волны излучения 1550 нм, мощностью излучения в одномодовом волокне 5 мВт, шириной спектра излучения 1 нм. Управление лазером осуществляется при помощи специальной электронной платы.

Назначением формирователя модулирующего сигнала является создание сигнала специальной формы, необходимого для вспомогательной модуляции интерферометра при осуществлении псевдогетеродинного метода регистрации интерференционного сигнала. Модулирующий сигнал содержит две гармонические компоненты с частотами 30 кГц и 60 кГц, которые синхронизированы между собой и имеют определенное соотношение амплитуд и фаз. Фотоприемник содержит фотодиод, специализированный усилитель и блок питания. Фотодиод имеет волоконный вход со стандартной FC вилкой и выход с разъемом «BNC». Фотодиод рассчитан на диапазон длин волн в области 1,3–1,5 мкм и имеет чувствительность 0,05 В/мкВт. Детектор фазомодулированных сигналов предназначен для работы в волоконно-интерферометрическом датчике переменных воздействий с использованием псевдогетеродинного метода регистрации, когда на выходе интерферометра создается сигнал поднесущей частоты, модулированный по фазе внешним воздействием.

Волоконный тракт [5], собранный по схеме волоконного интерферометра Маха–Цендера, представлен на рис. 2.

Интерферометрический чувствительный элемент реализован в относительно простом варианте с опорным плечом, изолированным от внешних воздействий, и чувствительным плечом, на которое воздействует гидроакустический сигнал. Опорное плечо (волокно), часть которого длиной 2–3 м намотана на пьезокерамический цилиндрический модулятор, смотано в катушку. Чувствительное плечо размещено на специальной опорной конструкции. Плечи интерферометра выполнены из двух отрезков стандартного одномодового волокна одинаковой длины.

Сигнал с выхода когерентного источника света подается на два оптоволоконных плеча – измерительное и опорное. Первое размещено на катушке гидрофона, сигналы которого подлежат регистрации, второе, равное по длине первому, остается на поверхности. В результате воздействия акустического сигнала происходит изменение длины волокна гидрофона. Сигналы с обоих волоконных плеч через коммутатор поступают по волоконно-оптическому кабелю на блок оптоэлектронной обработки информации (сигнала). Блок оптоэлектронной обработки фиксирует оптический сигнал и преобразует разность фаз в электрический сигнал и далее в цифровой код. Коды сигнала записываются в стандартной форме и затем обрабатываются специальными программными средствами.

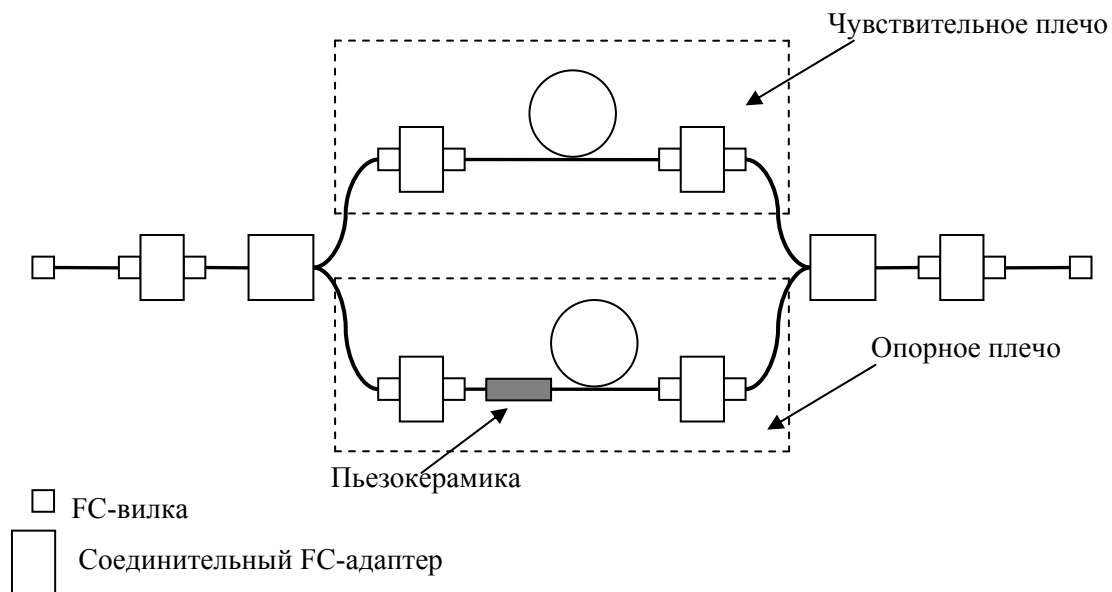


Рис. 2. Схема волоконного тракта

Приемник оптических сигналов, блок их обработки и программный комплекс образуют наземную часть разработанной автоматизированной системы, обеспечивающей обработку (интерпретацию), хранение и отображение информации гидродинамического контроля работы скважины.

Обработка данных

Обработку и отображение данных, полученных по показаниям акустического расходомера, выполняет программный комплекс, обеспечивающий:

- отображение временных реализаций и спектров сигналов, полученных с выхода акустического расходомера;
- применение высокочастотных, низкочастотных и полосовых цифровых фильтров при анализе регистрируемых сигналов;
- возможность изменения масштаба представленных для визуализации характеристик сигналов;
- комбинированное отображение характеристик сигналов от различных источников в разных временных интервалах;
- вывод на печать и сохранение преобразованной информации.

Программный комплекс имеет удобный пользовательский интерфейс. В интерфейсе предусмотрена возможность выбора файлов, содержащих информацию, в различных директориях, а также возможность выделения части информационных массивов для ускорения обработки при больших объемах данных. Предусмотрено представление характеристик сигналов в режимах «сигнал–сигнал» и «сигнал–спектр». В режиме «сигнал–сигнал» для пользователя предоставляется информация о сигналах с выхода расходомеров, установленных на различных глубинах, в различных интервалах перфорации. Эти сигналы для удобства отображаются на разных графиках. Режим «сигнал–спектр» характеризуется представлением временной реализации сигнала и его спектра. В этом режиме пользователю дана возможность выбора типа и параметров применяемых цифровых фильтров для обработки сигналов. В случае возникновения ошибки, связанной с некорректной работой библиотеки, открытием файла, не содержащего нужную информацию, и вводом некорректных данных пользователь получает сообщение о причине невыполнения требуемой функции.

Программная обработка гидроакустического сигнала позволяет по соответствующей методике определить расход флюида в скважине и его состав. Исследования, проведенные в ГАНиГ им. Губкина и СПбГТУ, показали, что наиболее информативными являются спектральные и статистические характеристики сигналов, регистрируемые с помощью акустических датчиков, помещенных в скважине над зонами перфораций.

С целью исключения ошибок при реализации методов цифровой обработки (определение спектральных и статистических характеристик сигналов, формирование цифровых фильтров) и ускорения процесса отладки программного комплекса применено взаимодействие разноразовых программ, таких как математический пакет MatLab и язык Java. Основные вычисления в программном комплексе производятся с использованием готовых математических пакетов. Это позволяет упростить и увеличить скорость математических расчетов и придать приложению удобный пользовательский интерфейс. Создание независимых приложений на основе математических библиотек пакета MatLab основано на разработке автономных модулей в этой среде [6]. Для получения модулей используются входящие в состав пакета утилиты – MatLab Compiler и MatLab Builder For Java. Для получения независимого компонента используется утилита DeploymentTool. Основные шаги по созданию компонента представлены на рис. 3.

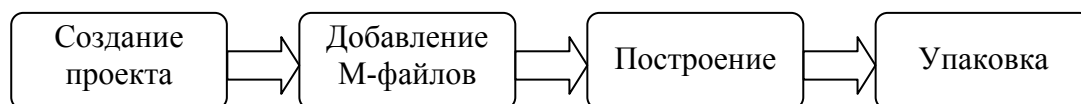


Рис. 3. Создание независимого компонента в пакете MatLab

Утилита DeploymentTool позволяет выбирать тип автономного модуля (C++, .Net, Java) и в соответствии с выбранным типом создает проект, добавляет в проект разработанные М-файл-функции, проводит сборку и упаковку компонента.

Так как в работе создавался модуль для работы с языком Java, то после перечисленных действий в корневом каталоге проекта создаются подкаталоги *distrib* и *src*. В папке *distrib* находится *.jar-файл и *.ctf-файл проекта. В папке *src* находятся *.java-файлы, которые позволяют посмотреть исходный код созданного пакета. Файл с расширением «*.ctf» – «component technology file» – это самораспаковывающийся файл, который обеспечивает доступ к предустановленному ядру пакета MatLab. Этот файл необходимо помещать в каталог, куда производится установка приложения, использующего библиотеки ядра MatLab. Файл с расширением «*.ctf» содержит в себе файлы пакета MatLab в кодированном виде, которые получаются в результате работы утилиты Java Builder. Созданный *.jar-файл можно использовать в любых приложениях, написанных на Java.

Для каждого m-файла, требующего преобразования в код на языке Java, создаются два java-класса: *myclass.java* и *mycomponentMCR.java*. Класс *mycomponentMCR* содержит ключи декодирования CTF-файла и код, инициализирующий MCR для работы выбранного компонента. Класс *myclass.java* включает методы доступа к М-файл-функции, ассоциирующейся с данным классом.

MCR – MatLab Component Runtime Environment – это средство, необходимое для установки ядра пакета MatLab на машину пользователя без установки самого пакета. MCR входит в пакет MatLab и может быть получен при создании проекта с помощью утилиты DeploymentTool. MCR требует однократной установки на машину пользователя. Для установки MCR используется инсталлятор – MCRinstaller.exe (для Windows) или MCRinstaller.zip (для Unix). После этого любое приложение, использующее в своей работе функции MatLab, может получать доступ к ядру, содержащему основные математические библиотеки.

М-файл-функция является типичным объектом языка программирования системы MatLab [7]. Одновременно она является полноценным модулем с точки зрения структурного программирования, поскольку содержит входные и выходные параметры и использует аппарат локальных переменных. М-файл-функция разрабатывается на М-языке, внутреннем языке программирования MatLab, код m-функции сохраняется в текстовых файлах, имеющих расширение «*.m». В разработанном программном комплексе применяются функции пакета Signal Processing Toolbox, являющегося одним из первых специализированных пакетов системы MatLab [8].

Расчет спектра регистрируемого сигнала производится непараметрическими методами. Это позволяет получить достоверную информацию о спектральных характеристиках без использования знаний о плотности распределения случайного процесса. Спектральный анализ переводит описание сигнала из временной области в частотную. Таким образом, спектральное представление сигналов позволяет изучать их частотный состав, т.е. судить о том, какой вклад в формирование сигнала вносят колебания определенных частот.

Для определения спектральных характеристик дискретного случайного процесса вычисляется средний спектр мощности его ограниченного по длине фрагмента (1), а затем длина фрагмента устремляется к бесконечности:

$$W(\omega) = \lim_{n \rightarrow \infty} \frac{1}{2n+1} \left| \sum_{k=-n}^n x(k) e^{-j\omega kT} \right|^2. \quad (1)$$

Здесь $x(k)$ – отсчеты случайного процесса, T – период дискретизации, черта сверху обозначает усреднение по ансамблю реализаций. Кроме того, этот спектр можно выразить через корреляционную функцию случайного процесса (2):

$$W(\omega) = \sum_{k=-\infty}^{\infty} R_x(k) e^{-j\omega kT}. \quad (2)$$

При использовании непараметрических методов расчета спектра случайного процесса используется только информация, заключенная в отсчетах сигнала, без каких-либо дополнительных предположений. Основными методами расчета спектра в пакете Signal Processing Toolbox являются периодограмма и метод Уэлча (Welch). Периодограмма (periodogram) – оценка спектральной плотности мощности, полученная по N отсчетам одной реализации случайного процесса. Периодограмма рассчитывается по следующей формуле:

$$\hat{W}(\omega) = \frac{1}{Nf_d} \left| \sum_{k=0}^{N-1} x(k) e^{-j\omega kT} \right|^2, \quad (3)$$

где f_d – частота дискретизации. Деление на частоту дискретизации необходимо для получения оценки спектральной плотности мощности случайного аналогового процесса, восстановленного по отсчетам $x(k)$.

При разработке программного комплекса был применен метод Уэлча, так как при вычислении периодограммы по длинному фрагменту случайного сигнала она оказывается весьма изрезанной. Для уменьшения этой изрезанности необходимо применить какое-либо усреднение, что и используется в методе Уэлча. Этот метод был разработан на основе метода Бартлетта (Bartlett), который предложил анализируемый сигнал делить на сегменты без перекрытия, для каждого сегмента вычислять периодограмму, а затем эти периодограммы усреднять. Если корреляционная функция сигнала на длительности сегмента затухает до пренебрежимо малых значений, то периодограммы отдельных сегментов можно считать независимыми. В этом случае дисперсия периодограммы Бартлетта обратно пропорциональна числу используемых сегментов, однако с ростом числа сегментов при фиксированном общем числе отсчетов сигнала падает

спектральное разрешение (за счет того, что сегменты становятся короче). Уэлч внес в этот метод два усовершенствования: использование весовой функции и разбиение сигнала на перекрывающиеся фрагменты [9]. Применение весовой функции позволяет ослабить растекание спектра (spectrum leakage) и уменьшить смещение получаемой оценки спектра плотности мощности ценой незначительного ухудшения разрешающей способности. Перекрывание сегментов введено для того, чтобы увеличить их число и уменьшить дисперсию оценки [10].

Так же как и для периодограммы Бартлетта, дисперсия оценки, получаемой методом Уэлча, уменьшается примерно пропорционально числу сегментов. Благодаря перекрытию в методе Уэлча используется больше сегментов, поэтому дисперсия оценки спектра плотности мощности оказывается меньше, чем для метода Бартлетта. Метод Уэлча является наиболее популярным периодограммным методом спектрального анализа. Вычисление по методу Уэлча производится функцией «pwelch» в пакете MatLab.

Для анализа временных реализаций используется дискретная фильтрация сигнала. Операция линейной дискретной фильтрации в общем случае имеет вид

$$y(k) = b_0 x(k) + b_1 x(k-1) + \dots + b_m x(k-m) - a_1 y(k-1) - a_2 y(k-2) - \dots - a_n y(k-n). \quad (4)$$

Здесь $x(k)$ – отсчеты входного сигнала, $y(k)$ – отсчеты выходного сигнала, a_i и b_j – постоянные коэффициенты. Максимальное из чисел m и n называется порядком фильтра. Предыдущие выходные отсчеты могут не использоваться при расчетах, тогда все $a_i = 0$, и фильтр называется нерекурсивным или трансверсальным. Если предыдущие выходные отсчеты используются, фильтр называется рекурсивным.

Линейная дискретная фильтрация реализована в функции filter [11] пакета MatLab. Входными данными этой функции являются наборы коэффициентов $a = \{a_i\}$ и $b = \{b_j\}$, одномерный массив данных X ; выходными данными является массив значений (после применения фильтра) y . Линейная дискретная фильтрация относится к технологиям обработки произвольных данных, поэтому функция filter не принадлежит пакету Signal Processing, а является встроенной в ядро MATLAB. В пакете Signal Processing расположены функции, реализующие разнообразные алгоритмы синтеза дискретных фильтров.

Под синтезом дискретного фильтра понимается выбор таких наборов коэффициентов $\{a_i\}$ и $\{b_j\}$, при которых характеристики получающегося фильтра удовлетворяют заданным требованиям. Строго говоря, в задачу проектирования входит выбор подходящей структуры фильтра с учетом конечной точности вычислений. В программе применяются эллиптические фильтры, обеспечивающие более крутой (по сравнению с фильтрами Баттерворта и Чебышева) спад АЧХ в переходной зоне между полосами пропускания и задерживания. Платой за это является наличие равномерных пульсаций АЧХ как в полосе пропускания, так и в полосе задерживания. Как правило, эллиптические фильтры позволяют выполнить требования, предъявляемые к АЧХ, при минимальном порядке фильтра. Для синтеза аналоговых и дискретных эллиптических фильтров нижних частот, верхних частот и полосовых фильтров используется функция ellip [12] пакета MatLab.

Функция возвращает описание фильтра в виде векторов-строк b и a (6), имеющих длину $n+1$ и содержащих коэффициенты полиномов числителя и знаменателя функции передачи в порядке убывания степеней переменной z :

$$H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \dots + b(n+1)z^{-n}}{1 + a(2)z^{-1} + \dots + a(n+1)z^{-n}}. \quad (5)$$

Перечисленные математические функции используются в разрабатываемых М-файл-функциях. Для взаимодействия созданного приложения и скомпилированных М-файл-функций, а также ядра системы MatLab в java-приложении необходимо создать метод, который инициализирует используемые математические функции [13].

Любая программа, написанная в системе MatLab, в качестве операндов использует n -мерные матрицы, имеющие специальную структуру `mxArray`. Отдельные поля этой структуры хранят следующую информацию: имя переменной среды MatLab в текстовом виде, размерность созданной переменной, тип переменной, является ли она вещественной или комплексной (в случае комплексной переменной создаются отдельно векторы вещественной и мнимой частей), является ли переменная разреженной матрицей (разреженные матрицы имеют отличную от обычных схему хранения данных).

Для использования функций, написанных в системе MatLab, необходимо преобразовать стандартные `java`-типы в типы MatLab [14]. С этой целью используются содержащиеся в пакете `com.mathworks.toolbox.javabuilder` классы и методы, облегчающие работу с тем или иным типом данных внутри расширения MATLAB.

В разработанном программном комплексе как входные, так и выходные данные для функции должны иметь тип `MWNumericArray`. `MWNumericArray` – это структура `mxArray`, применяемая для чисел. Конструктор `new MWNumericArray`(«переменная», «тип переменной в MatLab») преобразует стандартный `java`-тип в тип, который можно использовать в функции. Для присвоения результата работы `M`-функции `java`-переменной в классе `MWNumericArray` созданы методы `*.get` «Стандартный `java`-тип».

Заключение

Автоматизированная система сбора и обработки данных скважинных измерений с помощью акустического расходомера обеспечивает регистрацию параметров работы продуктивных пластов, вскрытых перфорацией, с целью определения количественного и качественного состава притока по каждому интервалу перфорации и технического состояния эксплуатационной колонны скважины. Использование современных волоконно-оптических технологий для контроля и оптимизации работы скважины позволяет резко увеличить нефтеотдачу месторождения.

Программная часть системы производит обработку и отображение данных, полученных с оптоволоконных датчиков. Применение полосовых, высокочастотных и низкочастотных фильтров позволяет модифицировать полученный спектр сигнала с целью дальнейшего анализа. Одной из особенностей данного программного комплекса является использование математических библиотек среды MatLab, что дает возможность применять сложные математические функции без создания их алгоритмов, уменьшает количество ошибок на стадии отладки приложений, повышает быстродействие комплекса, так как эти функции оптимизированы.

Разработанная автоматизированная система сбора и обработки данных скважинных измерений с помощью акустического расходомера обеспечивает получение необходимой информации для принятия решений по управлению работой скважины в соответствии с существующими геофизическими методиками и способствует повышению рентабельности разработки.

Литература

1. Форест Г. Добыча нефти. – М.: ЗАО «Олимп-Бизнес», 2003. – 416 с.
2. Крылов Д.Н. Детальный прогноз геологического разреза в сейсморазведке. – М.: ООО «Недра-Бизнесцентр», 2007. – 195 с.
3. Кульчин Ю.Н. Распределительные волоконно-оптические измерительные системы. – М.: ФИЗМАТЛИТ, 2001. – 272 с.
4. Янг М. Оптика и лазеры, включая волоконную оптику и оптические волноводы. – М.: Мир, 2005. – 541 с.

5. Назаров А.В., Козырев Г.И. Современная телеметрия в теории и на практике: Учебный курс. – СПб: Наука и Техника, 2007. – 672 с.
6. Бей И. Взаимодействие разноразовых программ. Руководство программиста. – М.: Издательский дом «Вильямс», 2005. – 880 с.
7. Ануфриев И.Е. Самоучитель MatLab 5.3/6.x. – СПб: БХВ-Петербург, 2002. – 736 с.
8. Дьяконов В.П. MATLAB 6.5 SP1/7 + Simulink 5/6®. Основы применения. Серия «Библиотека профессионала». – М.: СОЛОН-Пресс, 2005. – 800 с.
9. Куприянов М.С., Матюшкин Б.Д. Цифровая обработка сигналов: процессоры, алгоритмы, средства проектирования. – СПб: Политехника, 1999.
10. Percival D.B., Walden A.T. Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques. – Cambridge: Cambridge University Press, 1993.
11. Welch P.D. The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms. // IEEE Trans. Audio Electroacoust. – June 1967. – Vol. AU-15. – P. 70–73.
12. Бондарев В.Н., Трестер Г., Чернега В.С. Цифровая обработка сигналов: методы и средства. – Севастополь: СевГТУ, 1999. – 398 с.
13. Хорстманн К.С., Корнелл Г. Java 2. Библиотека профессионала, том II. Тонкости программирования. – М.: Издательский дом «Вильямс», 2006. – 1168 с.
14. Ноутон П., Шилдт Г., Java™ 2: Пер. с англ. – СПб.: БХВ-Петербург, 2005. – 1072 с.

Войтюк Татьяна Евгеньевна

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, voytuk@i-teco.ru, tanya_4ever@mail.ru

Демин Анатолий Владимирович

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, dav_60@mail.ru

УДК 004.021

РЕАЛИЗАЦИЯ МОДЕЛИ ОСВЕЩЕНИЯ КУКА-ТОРРЕНСА С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ DEFERRED SHADING

А.А. Безгодов, Э.В. Стародубцев

Рассматриваются достоинства и недостатки модели освещения Кука–Торренса, особенности технологии Deferred Shading и способ реализации данной модели освещения с использованием технологии Deferred Shading. Предлагается способ размещения данных в геометрическом буфере и ряд методов оптимизации.

Ключевые слова: компьютерная графика, модели освещения, deferred shading

Введение

Освещение в игровых системах является важным фактором, который влияет на восприятие виртуального мира пользователем. Пользователь должен в полной мере идентифицировать материалы и сопоставлять синтезируемое изображение с тем, что он видит в реальной жизни: металл должен выглядеть как металл, пластмасса – как пластмасса, и т.д. Во многом восприятие зависит от выбранной модели освещения. Некоторые модели годятся для описания лишь узкого круга материалов, в то время как другие модели позволяют описывать более широкий круг материалов. Для обеспечения гибкости в некоторых проектах используют несколько моделей освещения.

В данной работе рассматриваются особенности модели освещения Кука–Торренса, способ реализации с использованием технологии Deferred Shading, а также ряд методов оптимизации, которые позволяют ускорить построение изображения, несмотря на высокую вычислительную трудоемкость данной модели освещения.

Модель освещения Кука–Торренса

Модель освещения Кука–Торренса была представлена в [1]. Поверхность, отражающая свет, моделируется как совокупность блестящих микрограней, ориентированных в различных направлениях. Каждая микрогрань отражает падающий свет, и только те грани, которые ориентированы должным образом, отражают свет в сторону наблюдателя и вносят вклад в освещенность. Согласно [1], общая освещенность R рассчитывается как

$$R = sR_s + dR_d, \quad (1)$$

где s – весовой коэффициент зеркального отражения, d – весовой коэффициент диффузного отражения, причем $s + d = 1$, R_s – зеркальная составляющая освещенности, R_d – диффузная составляющая освещенности. Диффузная составляющая может быть рассчитана по закону Ламберта, а зеркальная составляющая выражается следующим образом (рис. 1):

$$R_s = \frac{FDG}{(\mathbf{n}, \mathbf{v})}, \quad (2)$$

где F – коэффициент Френеля, D – доля граней, ориентированных по направлению к наблюдателю, G – коэффициент затенения и экранирования микрограней, $\mathbf{n}$ – усредненная нормаль к поверхности, $\mathbf{v}$ – единичный вектор, направленный в сторону наблюдателя. Коэффициент Френеля зависит от коэффициента преломления материала для длины волны, равной длине волны падающего света:

$$F = \frac{1}{2} \frac{(g - c)^2}{(g + c)^2} \left\{ 1 + \left[\frac{c(g + c) - 1}{c(g - c) + 1} \right]^2 \right\}, \quad (3)$$

где $c = \cos(\theta) = (\mathbf{h}, \mathbf{v})$, $g = \eta^2 + c^2 + 1$. Здесь η – коэффициент преломления, c – косинус угла между вектором, направленным к наблюдателю, $\mathbf{v}$ и половинным вектором $\mathbf{h}$,

$$\mathbf{h} = \frac{\mathbf{l} + \mathbf{v}}{\|\mathbf{l} + \mathbf{v}\|}, \quad (4)$$

$\mathbf{l}$ – единичный вектор, направленный в сторону источника света.

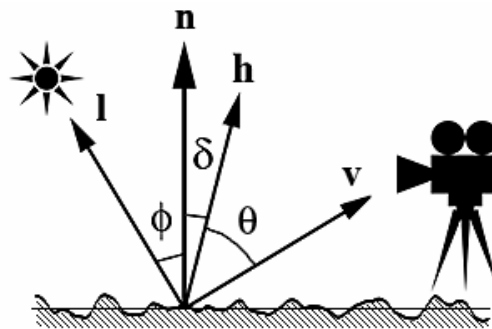


Рис. 1. Расчет освещения

Чтобы вычислить коэффициент преломления, достаточно знать коэффициент Френеля (F_0) для нормального падения света. В этом случае коэффициент преломления может быть выражен как

$$\eta = \frac{1 + \sqrt{F_0}}{1 - \sqrt{F_0}} \quad (5)$$

и подставлен в выражение (4). Данные вычисления выполняются для всех используемых длин волн, например для красного, зеленого и синего цвета. Коэффициент экранирования и затенения выражается как

$$G = \min \left[1, \frac{2(\mathbf{n}, \mathbf{h})(\mathbf{n}, \mathbf{v})}{(\mathbf{v}, \mathbf{h})}, \frac{2(\mathbf{n}, \mathbf{h})(\mathbf{n}, \mathbf{l})}{(\mathbf{v}, \mathbf{h})} \right]. \quad (6)$$

Доля граней, ориентированных по направлению к наблюдателю, может быть вычислена по одной из функций распределения, например, по функции распределения Гаусса или Бекмана. Согласно [1], распределение Бекмана хорошо согласуется с экспериментом для многих материалов и длин волн. Распределение может быть вычислено как

$$D = \frac{1}{m^2 \cos^4 \delta} e^{-\left[\frac{\tan \delta}{m}\right]^2}, \quad (7)$$

где m обозначает среднеквадратичный наклон микрограней и варьируется от 0,2 для гладких материалов до 0,6 для шероховатых.

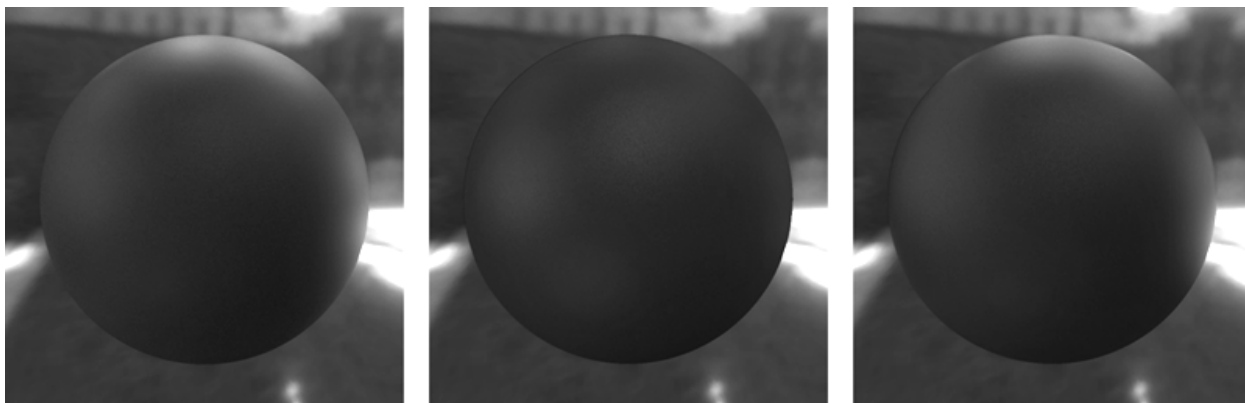


Рис. 2. Сравнение результатов расчета освещения на основе экспериментальных данных (слева), с использованием модели Блинна (в центре) и модели Кука-Торренса (справа)



Рис. 3. Сравнение моделей освещений Блинна (слева) и Кука-Торренса (справа)

Как видно из приведенных выше формул, модель освещения Кука-Торренса является весьма трудоемкой с вычислительной точки зрения. Поэтому в первую очередь возникает вопрос, что дает эта модель освещения по сравнению с другими моделями освещения. Согласно [2], модель освещения Кука-Торренса наиболее точно описывает взаимодействие света и материала для широкого круга материалов (рис. 2).

В ходе проведения исследовательской работы на базе игровой системы Quake 2, созданной фирмой id Software, была разработана собственная графическая система, реализующая обе модели освещения с использованием технологии Deferred Shading. Все приведенные ниже изображения получены в результате работы разработанной графической системы. Сцены созданы в редакторе Radiant для Quake 2, а текстуры взяты из игровой системы Doom 3, разработанной также фирмой id Software.

На рис. 3 представлен результат работы графической системы, работающей в режиме сравнения моделей освещения.

Технология Deferred Shading

Технология Deferred Shading была предложена Т. Saito и Т. Takahashi в работе «Comprehensible rendering of 3-D shapes» (1990). Особенностью данного подхода является расчет освещенности в пространстве изображения, а не в мировом пространстве или пространстве объектов, как это делается с использованием классического подхода. Фундаментальным объектом данной технологии является так называемый G-buffer (геометрический буфер). Он хранит все необходимые для расчета освещения атрибуты, не зависящие от конкретного источника света. Как правило, это информация о позиции, нормали и параметрах материала. Также геометрический буфер может содержать дополнительную информацию, которая может быть использована для различных эффектов пост-обработки, таких как размытие движения (Motion Blur) или свечение (Glow).

Обобщенный алгоритм расчет освещения выглядит следующим образом.

- | |
|---|
| <ol style="list-style-type: none">1. для каждого объекта в сцене:2. отобразить параметры материалов в G-buffer3. для каждого источника света:4. буфер_кадра += расчет_освещения (G-buffer, источник_света) |
|---|

Данная технология обладает рядом преимуществ и недостатков [3], поэтому решение о ее использовании во многом определяется спецификой проекта. Так, если присутствуют закрытые пространства и много небольших динамических источников света, то технология Deferred Shading может дать серьезный прирост производительности. Особо следует отметить случаи, когда проект нацелен на использование техник построения глобальной освещенности в реальном масштабе времени [4].

Распределение атрибутов в геометрическом буфере

Одной из важнейших задач разработки системы, использующей технологию Deferred Shading, является выбор формата и способа размещения атрибутов в геометрическом буфере. Современные видеокарты, поддерживающие Direct3D 9, поддерживают максимум четыре внеэкранные поверхности, в которые может осуществляться одновременная запись (Multiple Render Targets, MRT). Каждый MRT должен иметь одинаковую с другими MRT разрядность, например, 32, 64 или 128 бит. Возможные форматы представлены в табл. 1. Каждый формат описывается следующим образом: A, R, G, B обозначают альфа, красный, зеленый и синий каналы соответственно. Число рядом с обозначением канала означает количество бит в данном канале, F – формат с плавающей запятой.

Использование 64- и 128-битных форматов позволяет хранить данные без каких бы то ни было способов сжатия [2]. Однако, чем больше размер буфера, тем дольше выполняются операции записи и чтения. Таким образом, в первую очередь следует попытаться разместить данные в буфере с наименьшей разрядностью. Для этого следует выделить атрибуты, которые требуется записать в геометрический буфер – позиция точки в пространстве вида, нормаль в точке, цвет диффузного отражения, параметры зеркального отражения, вектора размытия движения, коэффициент «свечения», переключатель материалов.

	32 битные форматы	64 битные форматы	128 битные форматы
Форматы	A8R8G8B8 A2R10G10B10 G16R16 R32F	A16R16G16B16 A16R16G16B16F G32R32F	A32B32G32R32F
Размер геометрического буфера (1024x768).	12 Mb	24 Mb	48 Mb

Таблица 1. Форматы и размеры геометрического буфера

Так как в момент расчета освещения всегда известны экранные координаты точки, то восстановить видовые координаты X и Y можно, зная глубину и матрицу проецирования. Таким образом, в буфере достаточно хранить только глубину. Для обеспечения точности следует использовать не менее 24 бит для представления значения глубины (D). В противном случае возможны искажения изображения (рис. 4). Для 24 бит сгенерированное изображение не имеет искажений, для 16 бит присутствуют незначительные искажения, а для 8 бит расчет освещения дает полностью неправильные результаты.

Как показали эксперименты, для хранения нормали (N_x, N_y, N_z) достаточно 1 байта для представления каждой компоненты, а для хранения цвета диффузного отражения, (D_r, D_g, D_b), требуется 3 байта, т.е. 1 байт на каждую из трех компонент.

Параметры зеркального отражения обычно задаются как цвет зеркального отражения и дополнительный коэффициент (коэффициент шероховатости), определяющий форму блика. Однако такой подход требует 4 байта для хранения и является избыточным. Как правило, цвет зеркального отражения выбирают либо белым (для неметаллов), либо равным цвету диффузного отражения (для металлов), либо промежуточным. Таким образом, можно сократить представление за счет указания соотношения диффузной и зеркальных составляющих отраженного света, коэффициента шероховатости (m) и степени «металлизации» (k_{metal}), определяющей цвет зеркального отражения (k_s).

Переключатель материалов (s) необходим для обеспечения возможности использования нескольких моделей освещения. Обычно он представляет собой число, а диапазон, в котором оно находится, определяет модель освещения. Например, при значении от 0 до 0,5 используется модель Блинна, а от 0,5 до 1 – модель Кука–Торренса. Распределение данных в буфере представлено в табл. 2.

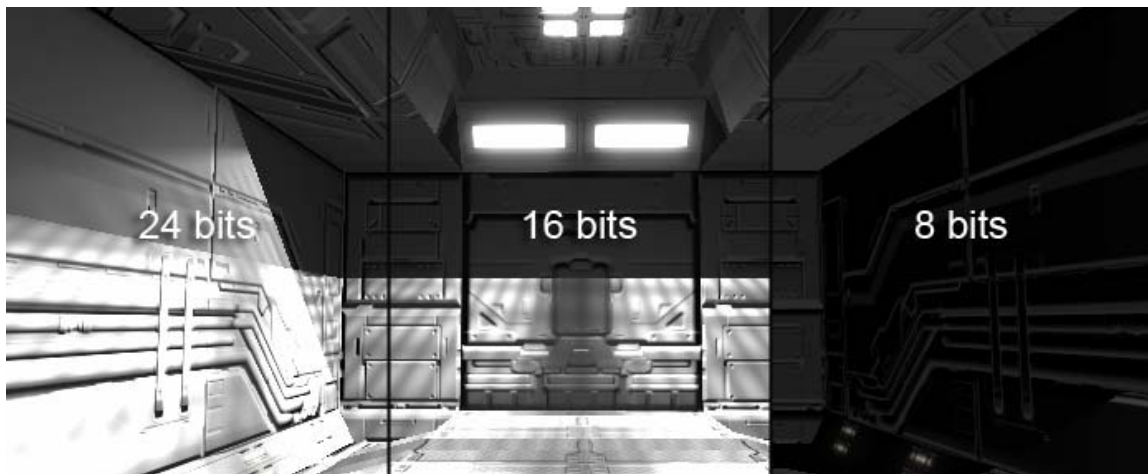


Рис. 4. Искажения изображения для различных разрешений буфера глубины

Коэффициент свечения (k_e) определяет, какая доля диффузного цвета не освещается и постоянно излучает. Этот параметр необходим для задания светящихся поверхностей, таких как поверхности ламп, мониторов или неба, которые могут встречаться в игровом пространстве. Вектора размытия (M_x, M_y) задают направления размытия изображения при

построении эффекта размытия движения. Для их задания достаточно 2 байт для компонент X и Y . Хранение векторов движения в геометрическом буфере было предложено в [5]. Светящиеся поверхности и эффект размытия представлены на рис. 5.

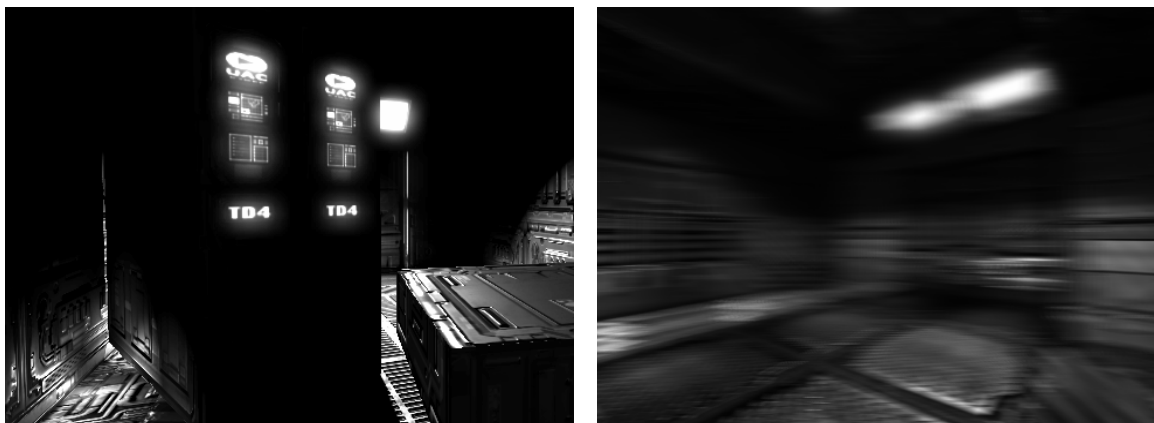


Рис. 5. Светящиеся поверхности и эффект размытия движения при повороте камеры

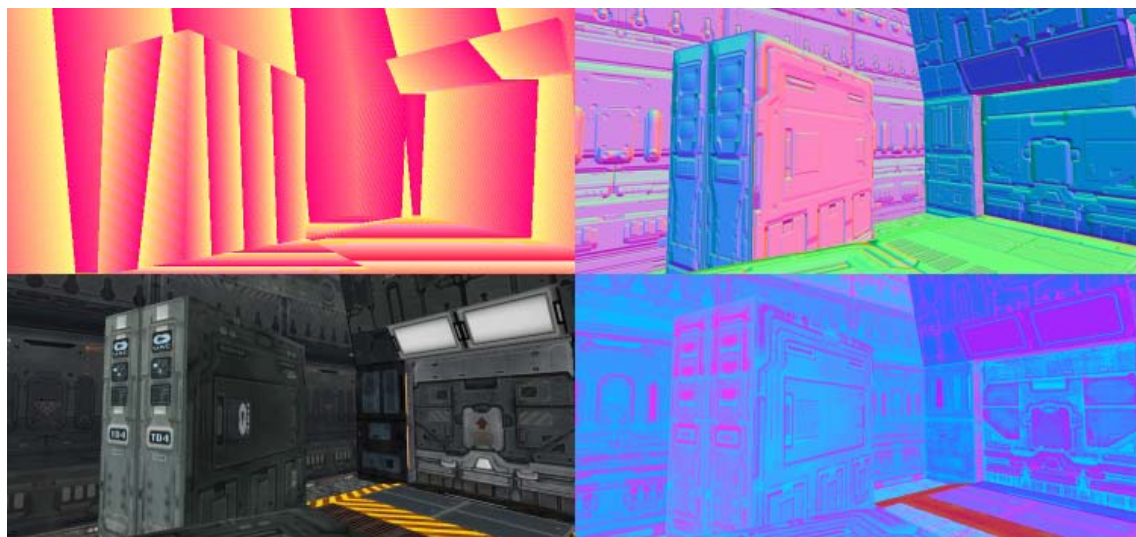


Рис. 6. Визуализация геометрического буфера



Рис. 7. Результат расчета освещения

	R	G	B	A
MRT0	D			M_x
MRT1	D_r	D_g	D_b	M_y
MRT2	N_x	N_y	N_z	k_e
MRT3	k_s	m	k_{metal}	s

Таблица 2. Распределение данных в геометрическом буфере

На рис. 6 представлен визуализированный геометрический буфер (RGB-компоненты), а на рис. 7 – результат расчета освещения по приведенному геометрическому буферу.

Использование таблиц для ускорения расчета освещения

Одним из основных недостатков модели освещения Кука–Торренса является ее высокая вычислительная сложность и, как следствие, низкая производительность приложений, которые используют данную модель освещения.

С помощью ShaderPerf¹ – программы, разработанной nVidia – можно оценить производительность шейдера. Был осуществлен замер производительности шейдера, выполняющего расчет освещения по модели освещения Блинна и Кука–Торренса. Результат представлен в табл. 3, строки 1 и 2, соответственно.

Шейдер	Производительность, Мpix/s
Модель освещения Блинна	505
Модель освещения Кука–Торренса	240
КТ, без расчета коэффициентов Френеля	471
КТ, без учета самозатенения микрограней	356
КТ, без учета распределения Бекмана	400

Таблица 3. Замер производительности шейдеров

Несмотря на то, что математически модель освещения Кука–Торренса сложнее модели Блинна почти на порядок, шейдер, реализующий эту модель освещения, уступает в производительности лишь в два раза. Это обуславливается тем, что, помимо расчета освещения, шейдер осуществляет чтение из текстур и выполняет дополнительные вычисления, общие для обеих моделей. Для выявления наиболее трудоемких операций были поочередно отключены расчет коэффициентов Френеля, учет самозатенения микрограней и учет распределения Бекмана. Результат оценки производительности представлен в табл. 3, строки 3–5 соответственно. Из таблицы видно, что наиболее трудоемкой операцией является расчет коэффициентов Френеля и распределения Бекмана.

Одним из эффективных способов ускорения сложных вычислений является использование таблиц. Таблицы хранятся в текстурах (Lookup texture, LUT), что, помимо высокой скорости вычислений (за один такт, если текстура находится в кэш-памяти), позволяет осуществлять практически бесплатную аппаратную линейную интерполяцию значений. Текстуры могут быть 1-, 2- и 3-мерным, т.е. возможно задание функции максимум от трех переменных. Для расчета коэффициентов Френеля требуются два параметра – косинус угла падения и коэффициент Френеля для нормального падения. Эти параметры, а также множество значений функции лежат в диапазоне $[0...1]$, таким образом, путем использования двумерной текстуры можно задать функцию в табличном виде.

В случае функции распределения Бекмана параметрами являются коэффициент шероховатости и косинус угла падения. Однако множество значений функции распре-

¹ Данная программа позволяет оценить производительность шейдеров для различных моделей графических ускорителей, здесь и далее приведены данные для ускорителя G70.

деления Бекмана не лежит в пределах диапазона $[0...1]$. В связи с этим имеет смысл использовать масштабирование значений или формат текстур с плавающей запятой. Запаковка значений в ARGB каналы текстуры неэффективна, так как в этом случае нельзя использовать аппаратную билинейную фильтрацию. Использование таблиц для расчета самозатенения микрограней невозможно, так как функция зависит от четырех переменных – всех скалярных произведений в выражении (6).

На рис. 8 представлены визуализированные таблицы для расчета коэффициентов Френеля и распределения Бекмана.

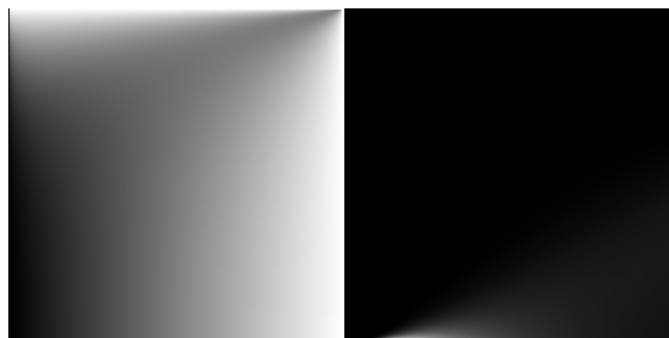


Рис. 8. Таблицы для расчета коэффициентов Френеля и распределения Бекмана

Результаты оценки производительности шейдера представлены в табл. 4. Как видно из результатов замера производительности, для шейдера, реализующего модель освещения Кука–Торренса, удастся достичь лишь 70% от производительности шейдера, реализующего модель Блинна. Таким образом, снижение производительности – это цена, которую приходится платить за качество генерируемого изображения.

	Шейдер	Производительность, Mpix/s
1	Модель освещения Кука–Торренса, использование таблицы для коэффициентов Френеля	331
2	Модель освещения Кука–Торренса, использование таблиц для коэффициентов Френеля и распределения Бекмана	355

Таблица 4. Производительность шейдеров, использующих табличное представление функций

Использование световых объемов для ускорения расчета освещения

В больших сценах может присутствовать огромное количество источников света. В классической интерпретации технологии Deferred Shading расчет освещения осуществляется для всех точек на экране. Таким образом, количество вычислений для каждой точки на экране будет расти пропорционально количеству источников света в сцене. Однако этого можно избежать, осуществляя расчет освещения только для тех точек, для которых вклад освещения от источника света является значимым. Как правило, вклад освещения зависит от формы и типа источника света, а также закона убывания силы света. Существует множество законов убывания силы света, однако для многих из них можно указать область, где вклад в освещение данным источником света значим, и области, где вклад незначим. Таким образом, световым объемом можно назвать область пространства, где вклад в освещенность данным источником света является значимым.

На рис. 9 представлены наиболее употребительные формы световых объемов: сферическая (для точечных источников света), коническая (для источников типа «прожектор») и цилиндрическая (для направленных источников света). Сферическая форма

светового объема образуется вокруг точечного источника света, а радиус выбирается равным расстоянию, на котором свет полностью затухает или вклад в освещение не является значительным. Коническая форма используется для источников света типа «прожектор», угол раствора конуса определяется углом рассеяния света, а высота – расстоянием, на котором свет затухает. Для направленных источников света (солнечный или дневной свет) затухание обычно отсутствует, и в этом случае размеры светового объема определяются на этапе создания сцены.

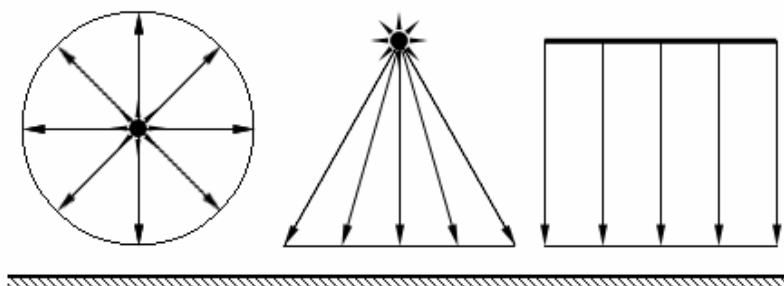


Рис. 9. Формы источников света

Таким образом, вместо расчета освещения для всех точек на экране осуществляется расчет освещения только для тех точек, в которые отобразился световой объем. Однако существует ряд случаев, в которых данный подход неэффективен (рис. 10). Значимый вклад в освещенность присутствует только в областях, отмеченных жирной линией, но отображены будут все три световых объема, что приведет к лишним затратам ресурсов.

Тем не менее, такие случаи можно обрабатывать и отбрасывать ненужные точки на экране с помощью буфера трафарета. Алгоритм выглядит следующим образом.

1. Буфер трафарета заполняется нулями.
2. Выставляется следующий режим работы буфер трафарета.
 - a. При отображении граней, обращенных к наблюдателю, если они не проходят буфер глубины, значение в буфере трафарета уменьшается на единицу.
 - b. При отображении граней, обращенных от наблюдателя, если они не проходят буфер глубины, значение в буфере трафарета увеличивается на единицу.
 - c. Для всех остальных случаев значение в буфере трафарета сохраняется.
3. Рисуется световой объем с выключенной записью в буфер цвета и глубины.
4. Выставляется следующий режим работы буфер трафарета.
 - a. Прimitives растеризуется только в тех точках, где значение в буфера трафарета равно 1.
 - b. Если тест трафарета проходит успешно (в буфере было значение 1), значение в буфере трафарета уменьшается на единицу. Таким образом, после отображения светового объема буфер будет опять заполнен нулями.
5. Рисуется световой объем с включенной записью в буфер цвета, используя шейдер, реализующий расчет освещения.
6. Перейти к пункту 2 и выполнить действия для следующего источника света.

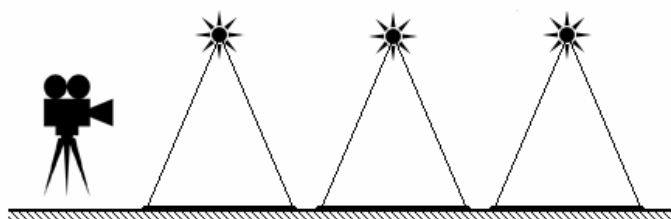


Рис. 10. Случай неэффективного использования световых объемов

На рис. 11 представлены этапы приведенного выше алгоритма. Серыми блоками обозначены элементы сцены. Линия слева – буфер трафарета, жирными линиями обозначены точки в буфере трафарета, где значение равно 1. Темно-серым обозначены области в сцене, для которых осуществляется расчет освещения.

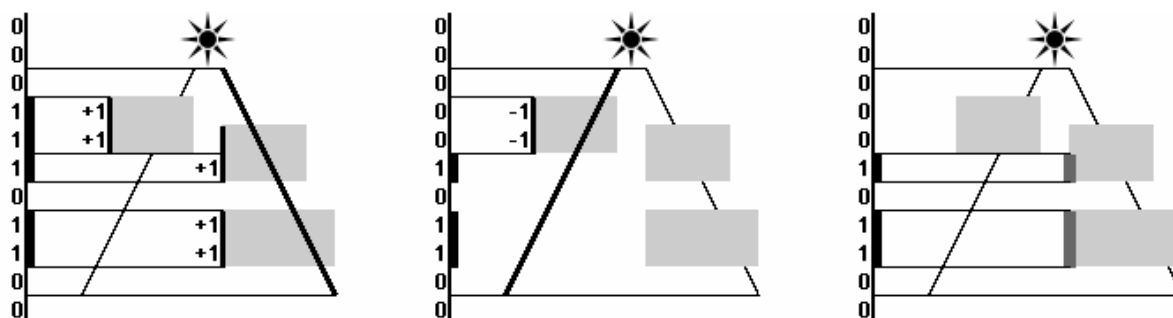


Рис. 11. Использование буфера трафарета для ускорения расчета освещения

Следует отметить, что приведенный алгоритм не требует очистки буфера трафарета для каждого источника света, а также работает для всех возможных случаев взаимного расположения наблюдателя и светового объема: наблюдатель вне объема; наблюдатель внутри объема; наблюдатель расположен так, что экранная плоскость пересекает световой объем. Такой подход обеспечивает стабильную производительность для больших сцен с большим количеством равномерно расположенных по сцене источников света. На рис. 12 представлены визуализированные световые объемы для четырех источников света типа «прожектор».

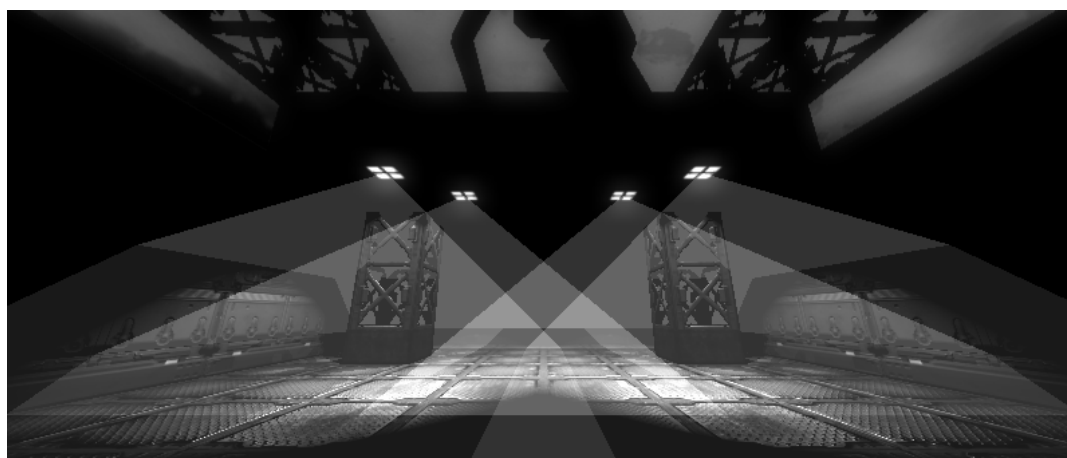


Рис. 12. Визуализированные световые объемы

Заключение

Рассмотрена модель освещения Кука–Торренса. Разработана графическая система на базе игровой системы Quake 2 (исходные тексты Quake 2 доступны по лицензии GPL), в которой реализованы модели освещения Кука–Торренса и Блинна. Обе модели освещения реализованы с использованием технологии Deferred Shading.

Предложен метод размещения данных в геометрическом буфере минимальной разрядности (128 бит), позволяющий задавать разнообразные материалы и размещать данные, необходимые для эффектов постобработки. Предложен ряд методов оптимизации расчета освещения, осуществлен ряд действий в направлении увеличения производительности разработанной графической системы.

По сравнению с использованием модели освещения Блинна использование модели освещения Кука–Торренса дает незначительное снижение производительности (скорость заполнения 505 Mpix/s против 355 Mpix/s), а взамен дает более естественный вид бликов на поверхности различных материалов. Как показывает опыт разработки таких систем, расчет освещения занимает лишь часть от всего времени обработки кадра (в приведенных сценах эта величина колеблется от 20% до 50% от общего времени расчета кадра), и, как следствие, падение производительности становится еще менее значимым.

Особый интерес для дальнейших исследований в технологии Deferred Shading представляют материалы, имеющие несколько градаций неровностей; анизотропные материалы; многослойные материалы; материалы, обладающие свойством подповерхностного рассеивания (subsurface scattering).

Литература

1. Cook R.L., Torrance K.E. A Reflectance Model for Computer Graphics. – 1982.
2. Ngan A., Durand F., Matusik W. Experimental Analysis of BRDF Models. – 2004.
3. Harris M., Hargreaves S. Deferred Shading. Presentation. – 2004.
4. Shishkovtsov O. Deferred shading in S.T.A.L.K.E.R. – 2005.
5. Michiel van der Leeuw. Deferred Rendering in Killzone 2. Presentation. – 2007.

Безгодов Алексей Алексеевич

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, demiurghg@gmail.com

Стародубцев Эдуард Владимирович

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, sevedw@dl.ifmo.ru

УДК 004.49; 629.7.08

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ КОНТРОЛЯ СИСТЕМ ПОСАДКИ ВОЗДУШНЫХ СУДОВ

А.О. Саута, В.И. Поляков

Приводится краткое описание спутниковой системы посадки и дифференциального режима работы спутниковой навигационной системы. С учетом этих разработок рассматривается проблема контроля точности существующих систем посадки воздушных судов и предлагается вариант ее решения с применением конкретной аппаратуры, имеющейся на рынке авиационного оборудования.

Ключевые слова: посадка воздушных судов, спутниковая навигационная система, спутниковая система посадки.

Введение

Необходимость оснащения аэропортов и воздушных судов (ВС) оборудованием, облегчающим пилотам посадку на взлетно-посадочную полосу (ВПП), была осознана довольно давно. С середины XX века установка радиотехнических посадочных систем производится почти повсеместно. За время развития авиации было разработано несколько типов систем посадки, таких как MLS (Microwave Landing System – микроволновая система посадки), спутниковая система посадки (ССП) и КГС (курсо-глиссадная система, англ. ILS – Instrumental Landing System). Однако международным стандартом на данный момент является именно последняя, поскольку она была разработана раньше других и внедрена во многих аэропортах. Также следует отметить возможность приме-

нения для посадки системы VOR/DME или ее российского аналога РСБН¹, которые являются системами ближней навигации. Тем не менее, для гражданской авиации считаться специализированными посадочными системами они не могут.

КГС является довольно сложным и дорогостоящим комплексом аппаратуры, состоящим из нескольких наземных радиомаяков и бортового приемника, обеспечивающего индикацию параметров для экипажа ВС. Он требует постоянного квалифицированного обслуживания и наблюдения. В связи с тем, что излучаемый маяками радиосигнал распределен в довольно большом объеме пространства, на точность работы КГС влияет геометрия местности, наличие источников электромагнитного излучения определенной частоты и другие факторы. Чтобы определить, насколько точно функционирует система, необходимо сравнить полученные с ее помощью данные в нескольких точках (отклонения ВС по курсу и глиссаде от заданной траектории) с эталонными значениями.

В данной работе предлагается решение этой проблемы на основе спутниковой системы посадки (ССП), работающей в дифференциальном режиме и обеспечивающей на данный момент точность, достаточную для посадки по I категории ИКАО (реально точность СПП намного выше, может обеспечиваться посадка и по III категории, но такая возможность не сертифицирована на данный момент).

1. Курсо-глиссадная система посадки

КГС является самой распространенной системой посадки, используемой в современной гражданской авиации. В связи с тем, что она была разработана довольно давно, в ней применены морально устаревшие технологии. Однако система зарекомендовала себя как высоконадежная и, как правило, не вызывает нареканий. Рассмотрим кратко ее устройство и принцип работы.

КГС состоит из наземной и бортовой частей. Наземная часть состоит из пары глиссадных и пары курсовых радиомаяков, а также нескольких (как правило, 2–3) маркерных радиомаяков. На борту ВС устанавливается приемник, осуществляющий выдачу на панель пилотажно-навигационного прибора (ПНП) отклонений ВС по курсу и глиссаде. Таким образом, экипаж ВС может контролировать текущее отклонение от заданной траектории захода на посадку. Общая схема КГС приведена на рис. 1.

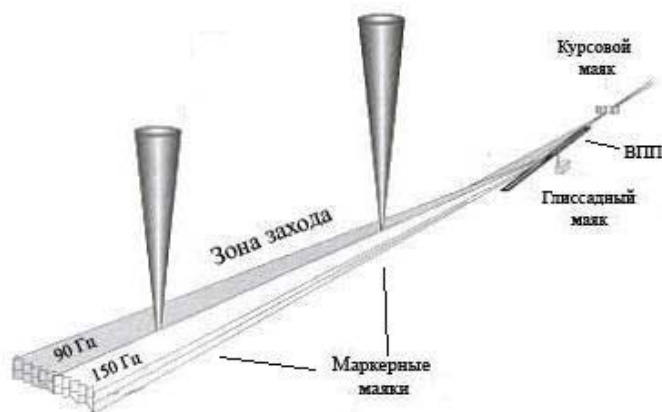


Рис. 1. Общая схема КГС

1.1. Принцип работы КГС

Пара курсовых радиомаяков устанавливается за взлетно-посадочной полосой (ВПП) и излучает вдоль нее направленный радиосигнал. Излучение маяков происходит под небольшим углом друг к другу на частоте около 108 МГц. Однако излучение каж-

¹ Слово «аналог» употреблено здесь в смысле общности решаемых задач. Аппаратура VOR/DME и РСБН несовместимы друг с другом.

дого маяка модулируется разной частотой, 80 и 150 Гц. Ровно посередине, на осевой линии ВПП, интенсивность сигналов обоих маяков одинакова. Это позволяет делать вывод о том, что при равенстве амплитуд принимаемого сигнала приемник находится точно на заданном курсе. При различии интенсивности принимаемых сигналов можно определить, в какую сторону произошло отклонение от курса и насколько далеко. Кроме навигационных сигналов, курсовой маяк передает свой идентификационный код – две или три буквы азбукой Морзе. Это позволяет пилоту или штурману удостовериться, что он настроился на нужную КГС.

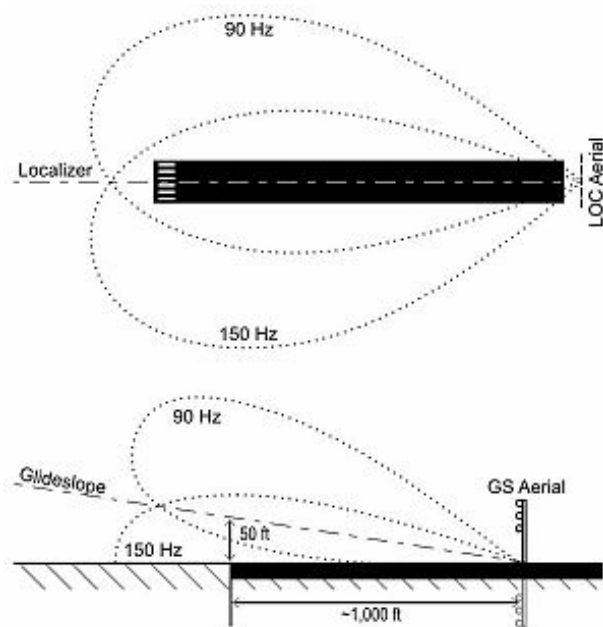


Рис. 2. Схема действия курсовых и глиссадных радиомаяков

Пара глиссадных радиомаяков устанавливается в районе приземления и, аналогично курсовым маякам, излучает пару направленных сигналов. Однако, в отличие от курсовых маяков, излучение происходит в плоскости горизонта. Это позволяет получить информацию об отклонении приемника от траектории в вертикальной плоскости (по глиссаде). Схема действия курсовых и глиссадных радиомаяков приведена на рис. 2.

Угол наклона линии глиссады определяется взаимной ориентацией глиссадных маяков и составляет, как правило, около 3° . Чем меньше угол наклона, тем удобнее садиться самолету. Однако в зависимости от конкретной местности он может изменяться и составлять 4° и более (например, в горной местности).

Помимо курсового и глиссадного, в состав КГС входят также маркерные радиомаяки. Их назначение – проинформировать экипаж о правильной работе посадочной системы и позволить контролировать правильность захода. Маркерные маяки работают на частоте 75 МГц, излучая узконаправленный сигнал вверх. При нахождении ВС над маркерным маяком экипаж оповещается специальным индикатором на приборной панели и звуковым сигналом.

1.2. Ограничения КГС

Из-за применяемых частот и других особенностей системы она имеет свои ограничения по применению. Директорные системы в самолетах чувствительны к отражениям сигналов КГС, способным вызвать серьезные ошибки в определении отклонения от заданной траектории захода. Форма земной поверхности, различные крупные объекты, металлические конструкции и пр. могут отражать сигнал, искажая тем самым пространственную картину излучения КГС. С целью устранения указанных факторов при-

нимаются различные меры, такие как выравнивание поверхности земли, устранение крупных объектов из прилегающей к КГС зоны, ограничение руления самолетов при использовании системы и пр. Из других ограничений следует отметить возможность захода на посадку лишь по прямой траектории.

С точки зрения потребителя сигналов КГС (бортовой аппаратуры) означенные искажения вызывают искривление заданной траектории захода, которые необходимо удерживать в пределах, допустимых заявленной категорией КГС. Однако на данный момент не существует простого и эффективного метода контроля точности работы посадочной системы. В работе предлагается в качестве контрольной использовать спутниковую систему посадки, основные принципы работы которой описаны в следующем разделе.

2. Спутниковая система посадки

ССП – сравнительно новая разработка, имеющая большие перспективы. Для определения местоположения ВС и других навигационных параметров она использует глобальную спутниковую навигационную систему (ГНСС) любого типа, NAVSTAR и/или ГЛОНАСС. Основные принципы функционирования ГНСС хорошо документированы и изложены во множестве источников, например в [1–4]. В данной работе остановимся на вопросах, непосредственно связанных с обеспечением посадки ВС.

Главные проблемы, которые должны быть решены при построении ССП, это ее точность и надежность. Для решения этих задач используется дифференциальный режим работы ГНСС. При этом ГНСС дополняется дифференциальными подсистемами, снижающими погрешность навигационных измерений.

2.1. Дифференциальная подсистема ГНСС

Основным фактором, позволяющим повышать точность ГНСС за счет дополнения ее дифференциальными подсистемами, является относительное постоянство значительной части погрешностей спутниковой радионавигационной системы в пространстве и времени. Построение дифференциального режима предполагает наличие минимум двух спутниковых приемоизмерителей (ПИ). Измеритель ПИ1, расположенный на контрольно-корректирующей станции (ККС), имеет точную геодезическую привязку и формирует кодовые коррекции PRC_i и скорости их изменения для каждого спутника наблюдаемого созвездия:

$$PRC_i(k) = R_i(k) - PR_i(k) - t_{sv_gnss,n,i}(k),$$

$$RRC_i(k) = \frac{PRC_i(k) - PRC_i(k-1)}{2},$$

где PR_i – псевдодальность, измеренная ПИ1 для i -го спутника, $i = 1 \dots M$, M – число сопровождаемых ПИ1 спутников, R_i – расчетная дальность, $t_{sv_gnss,i}(k)$ – временная поправка шкалы спутника относительно шкалы спутниковой системы, k – временной отсчет, RRC_i – скорость изменения кодовой коррекции. При расчете дальности до спутника R_i или эфемеридном расчете ККС по линии связи передает сформированные коррекции и скорость их изменения в ПИ потребителя ПИ2, расположенные на борту воздушного судна. Из измеренных в ПИ2 псевдодальностей и полученных коррекций формируется уточненное измерение псевдодальности:

$$PRA_i(k) = PRA_i(k) + PRC_i(k) + (k - k_{zcount})RRC_i(k) + TC_i,$$

где $PRA_i(k)$ – измеренная псевдодальность до i -го источника дальности, k_{zcount} – дискретный момент времени формирования дифференциальной поправки, TC_i – тропо-

сферная коррекция. Тропосферная коррекция i -го спутника рассчитывается в соответствии с [5]:

$$TC_i = N_R h_0 \frac{10^{-6}}{\sqrt{0,002 + \sin^2(El_i)}} (1 - \exp(-\frac{\Delta h}{h_0})),$$

где N_R и h_0 – индекс рефракции и высота по тропосферной шкале, передаваемые наземной станцией, Δh – высота воздушного судна над опорной точкой ЛККС, El_i – угол возвышения спутника. Таким образом, дифференциальный режим работы ГНСС позволяет компенсировать ошибки измерения псевдодалности, обусловленные задержкой радиосигнала в верхних слоях атмосферы (ионосфере, тропосфере).

В зависимости от радиуса рабочей зоны выделяют широкозонные, региональные и локальные дифференциальные подсистемы. Последние позволяют добиться наибольшей точности, поэтому именно их целесообразно использовать при посадке ВС.

В задачи локальной ККС авиационного назначения (ЛККС-А) входит также контроль целостности, достоверности и доступности всех передаваемых данных. Это достигается за счет применения стандартных методов цифрового отказоустойчивого кодирования, резервирования всех функциональных блоков, использования нескольких (3–4) приемоизмерителей (ПИ) ГНСС, дополнительных систем контроля.

Кроме того, для решения задачи посадки необходимо также передавать на борт ВС информацию о параметрах траектории захода (координаты начала ВПП, смещение относительно нее по широте и долготе конца ВПП, угол наклона глиссады по отношению к горизонту). Эти данные также передаются в цифровом виде по тому же радиоканалу, что и дифференциальные поправки.

2.2. Принцип работы ССП

Общий принцип работы ССП с использованием дифференциального режима ГНСС и ЛККС можно проиллюстрировать схемой (рис. 3).

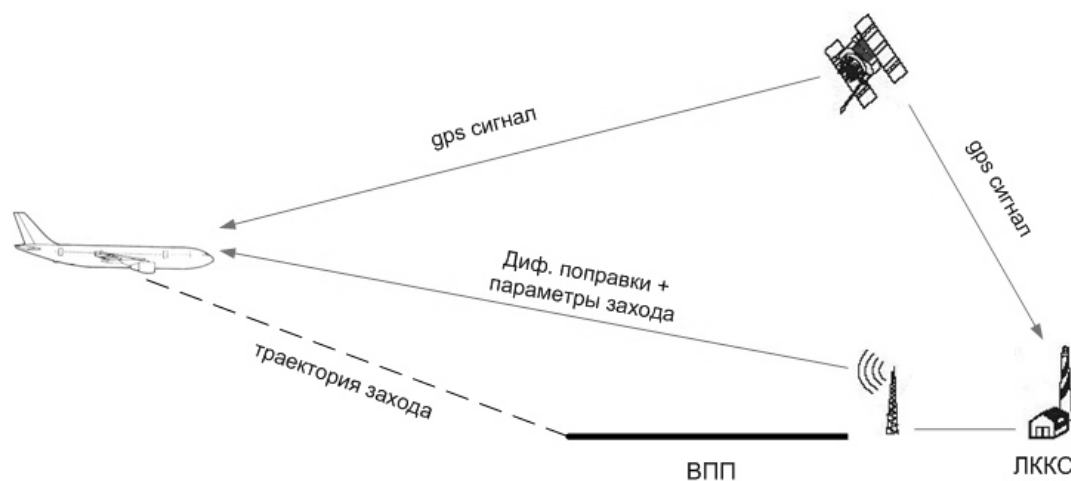


Рис. 3. Схема спутниковой системы посадки

Как видно из схемы, наземная часть ССП, представленная функциональным дополнением ЛККС-А, принимает сигналы ГНСС и производит вычисление дифференциальных поправок. Следует отметить, что для этого должна быть выполнена геодезическая привязка ЛККС с достаточно высокой точностью.

Вычисленные поправки и данные о траектории захода упаковываются в формат ICAO RTCA, после чего передаются по радиоканалу всем ВС, находящимся в зоне действия ЛККС. Бортовая часть ССП также оснащена ПИ ГНСС, на вход которого пода-

ются дифференциальные поправки, предварительно принятые специальной бортовой аппаратурой и преобразованные во внутренний формат приемоизмерителя. При решении навигационной задачи ПИ ГНСС учитывает указанные поправки и выдает внешним потребителям уточненные координаты. Одним из потребителей координат является бортовой прибор, способный на их основе рассчитать в реальном времени текущее отклонение ВС от заданной траектории захода, также принятой от ЛККС-А.

ЛККС передает дифференциальные поправки, которые могут использоваться всеми ВС (или другими потребителями, оснащенными ПИ ГНСС, способными работать в дифференциальном режиме), находящимися в зоне ее действия. Однако аэропорт, как правило, имеет несколько ВПП, параметры траектории захода на которые различаются. Поэтому ЛККС передает одновременно параметры для всех обслуживаемых полос, а бортовые приемники должны выбрать нужную.

3. Контроль КГС

С учетом вышеизложенных принципов может быть предложена структура программно-аппаратного комплекса, предназначенного для контроля точности работы КГС. В основе его функционирования лежит получение значений отклонений от траектории захода от исследуемой (КГС) и эталонной (ССП) систем посадки и последующего их сравнения. В зависимости от результатов сравнения делается вывод о необходимости коррекции и дополнительной настройки КГС.

Однако при разработке комплекса необходимо учесть, что в России в настоящее время практически нет аэропортов, оборудованных ССП. А это значит, что необходимо предусмотреть возможность ее быстрого временного развертывания. Для этого наземная часть комплекса выполняется в виде мобильной станции. Так как в данном случае собственно посадка с использованием данной системы производиться не будет, то нет необходимости сертификации оборудования как посадочного. Также возможно некоторое смягчение требований по надежности и резервированию аппаратной части.

3.1. Структура и принцип работы контрольного комплекса

Структура контрольного комплекса (КК) приведена на рис. 4. Наземная часть представляет собой систему, состоящую из двух блоков – «Архат» и РСБН. Аппаратура «Архат» представляет собой систему навигации и управления полетом и предназначена для решения таких задач, как зональная навигация, планирование полета в горизонтальной и вертикальной плоскостях, инженерно-штурманский расчет, выдача сигналов в систему автоматизированного управления (САУ) и директорные приборы. Однако в рамках данной задачи используется его встроенный ПИ ГНСС и вычислитель, позволяющий рассчитать дифференциальные поправки.

Второй компонент наземной части комплекса – аппаратура РСБН-85. Она представляет собой прибор для обеспечения ближней навигации, способный решать множество задач. В рамках КК предполагается использовать данное устройство для приема и передачи данных по радиоканалу. Иначе говоря, устройство используется в качестве приемопередатчика. В наземной части аппаратура РСБН предназначена для передачи рассчитанных в первом блоке поправок и параметров траектории захода на борт ВС.

Бортовая часть состоит из трех блоков, два из которых идентичны наземным (по крайней мере, в аппаратной части). Это приборы «Архат» и РСБН-85. Аппаратура РСБН обеспечивает прием дифференциальных поправок, передаваемых наземной частью, которые затем передаются устройству «Архат», которое использует их во встроенном ПИ ГНСС для обеспечения дифференциального режима. Это позволяет получить уточненные координаты ВС и, с использованием принятых от наземной части параметров траектории захода, рассчитать отклонения от нее по курсу и глиссаде.

Третьим блоком бортовой части комплекса является аппаратура VIM-95, принимающая сигналы контролируемой КГС и выдающая в «Архат» отклонения ВС по курсу и глиссаде, которые затем сравниваются с вычисленными вышеописанным способом отклонениями. Результаты выводятся на дисплей «Архат» и могут быть зарегистрированы несколькими способами для последующего анализа.

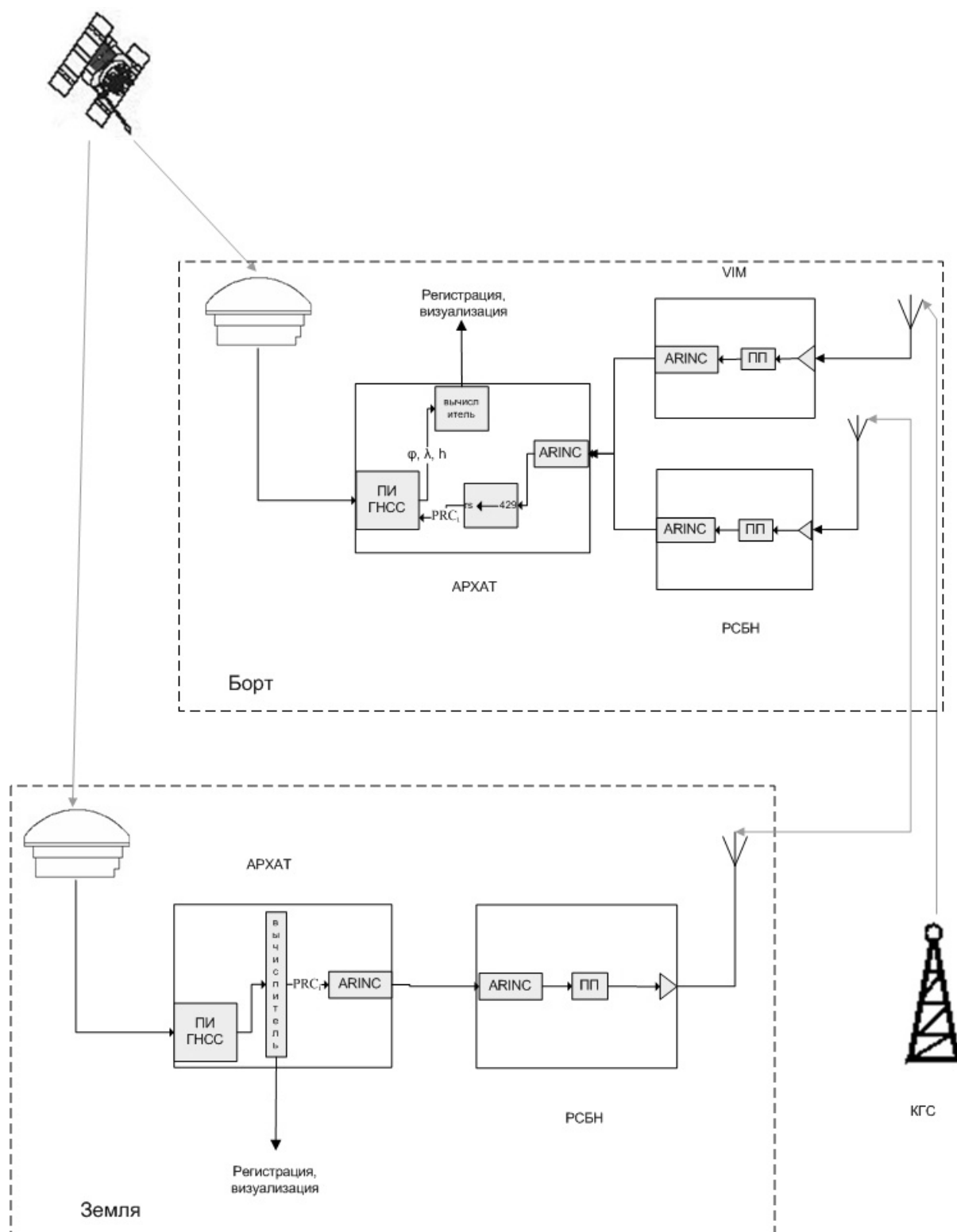


Рис. 4. Структура контрольного комплекса

Рассмотрим алгоритм работы комплекса и маршруты передачи данных в нем. Наземная часть осуществляет прием сигнала ГНСС с помощью антенны, подключенной к встроенному в «Архат» ПИ ГНСС, который осуществляет необходимые измерения и предоставляет необработанные измерения псевдодальности для каждого спутника вычислительного устройства. Так как предполагается, что геодезическая привязка выполнена заранее и географические координаты антенны известны с большой точностью, то можно с применением упомянутых выше алгоритмов сформировать кодовые коррекции для каждого спутника (PRC_i). Далее коррекции, упакованные в формат RTCA, передаются в цифровой канал стандарта ARINC-429, с помощью которого осуществляются информационный обмен между всеми блоками в составе комплекса. Поправки принимаются прибором РСБН-85, преобразуются внутри него в аналоговую форму, усиливаются и подаются на антенну прибора для выдачи в эфир. Дополнительно «Архат» передает в РСБН заранее заданные параметры траектории захода.

На борту ВС аппаратура РСБН осуществляет прием сигнала от наземной части. Она производит его перевод в цифровой вид и преобразование в формат ARINC-429 для выдачи в «Архат». Внутри аппаратуры «Архат» принятые дифференциальные поправки преобразуются в формат интерфейса RS-232 и подаются на вход встроенного ПИ ГНСС. ПИ ГНСС осуществляет прием спутникового навигационного сигнала с помощью антенны, после чего решает навигационную задачу с использованием принятых дифференциальных поправок. Полученные таким образом уточненные координаты ВС используются вычислителем для расчета отклонений по курсу и глиссаде от принятой от наземной части траектории захода. Аппаратура VIM-95 осуществляет прием сигнала от контролируемой КГС. Полученные с ее помощью отклонения преобразуются в формат ARINC-429 и передаются в «Архат» для сравнения с рассчитанными в нем параметрами.

3.2. Программная часть комплекса

Поскольку решаемая комплексом задача довольно специфична, требуется разработка специализированного программного обеспечения (ПО), устанавливаемого в прибор «Архат», что и позволяет использовать его в вышеописанном качестве. Темп расчета отклонений ВС от заданной траектории должен составлять около 20 Гц, что является довольно высоким требованием. «Архат» содержит в своем составе высокоинтегрированную ЭВМ архитектуры x86, имеющую оперативную память объемом 16 Мб и тактовую частоту процессора 33 МГц. Это позволяет использовать на нем операционную систему реального времени QNX, хорошо зарекомендовавшую себя при решении подобных задач. ПО, разрабатываемое под QNX, должно пройти тщательное тестирование и удовлетворять требованиям к производительности. Кроме того, в бортовой и наземной части его функции несколько разнятся.

ПО прибора «Архат» должно позволять оператору на борту ВС в реальном времени проводить экспресс-анализ работы обеих посадочных систем, а в наземной части осуществлять слежение за полетом ВС, для чего может потребоваться передача данных с борта к наземной части. Также данные от обеих систем должны выдаваться по каналу ARINC для обеспечения возможности регистрации другими бортовыми системами.

Заключение

Поскольку во многих аэропортах России мероприятия по контролю точности работы посадочной системы проводятся нерегулярно и явно недостаточны по объему, необходимо в ближайшее время ввести контрольный комплекс в эксплуатацию. Данные, полученные с его помощью, помогут повысить безопасность воздушных перевозок.

Предложенное решение может быть реализовано в ближайшее время и использовано на специальном ВС, предназначенном для исследовательских и контрольных ра-

бот. При реализации комплекса необходимо использовать современные вычислительные средства со специально разработанным программным обеспечением.

К достоинствам данного решения можно отнести такие качества, как мобильность (наземная часть может переноситься даже одним человеком), точность выполняемых расчетов, относительную дешевизну оборудования, а также использование оборудования, которое имеется в свободной продаже на рынке авиационных приборов. К недостаткам комплекса можно отнести необходимость выполнения геодезической привязки во время развертывания на аэродроме.

Представляется целесообразным изготовление опытного образца в ближайшее время и его введение в эксплуатацию после завершения соответствующих испытаний.

Литература

1. Радиотехнические системы: Учебное пособие для вузов по спец. «Радиотехника» / Под ред. Ю.М. Казаринова. – М.: Высш. шк., 1990. – 496 с.
2. ГЛОНАСС. Принципы построения и функционирования. / Под ред. А.И. Перова, В.Н. Харисова. – Изд. 3-е, перераб. – М.: Радиотехника, 2005. – 288 с.
3. Соловьев Ю.А. Системы спутниковой навигации. – М.: Эко-Трендз, 2000. – 270 с.
4. Соловьев Ю.А. Спутниковая навигация и ее приложения. – М.: Эко-Трендз, 2003. – 326 с.
5. RTCA/DO-253A

Саута Александр Олегович

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, demosito@gmail.com

Поляков Владимир Иванович

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, v_i_polyakov@mail.ru

УДК 004.93

СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ ПЕЧАТНЫХ ДОКУМЕНТОВ

З. Хоанг, А.Г. Коробейников

Разработан алгоритм многошкальной сегментации на базе динамической локальной карты связи (ДЛКС) изображения для получения иерархического дерева блоков документов, имеющих сложную структуру (присутствие рисунков, пятен, текстовых символов разных шрифтов и т.д.).

Ключевые слова: конференция, распознавание символов, сегментация изображения.

Введение

Хотя коммерческое программное обеспечение для распознавания печатных документов давно появилось и достигло высокой точности, все оно представлено закрытыми системами, используемые в них алгоритмы скрываются.

Как известно, процесс преобразования документов, хранимых в традиционных форматах (газета, книга и т.д.), в текстовый вид производится через многие этапы (дискретизация, бинаризация, подавление шумов, блок-сегментация, извлечение строк и символов, распознавание и т.д.). Цель данной работы – решить одну из задач, встречающихся при обработке изображения для распознавания, а именно – задачу сегментации изображения. В данный момент большинство исследований в области распознавания текста

ориентируется на поиск хороших алгоритмов для распознавания символов разных языков. Процесс сегментации мало исследуется, и методы сегментации, описанные в литературе, недостаточно подробны и/или имеют недостатки при применении.

Сегментация изображения – один из важных этапов в системе оптического распознавания текста. Цель этого этапа – выделение на введенном документе той области, в которой представлен текст, и отделение текстовой информации от графической. Сегментация включает в себя геометрический и логический анализ макета страницы. Геометрический анализ определяет максимально гомогенные области и их классификации (текст, таблица, изображение, фигуры и т. д.). Задача логического анализа – определение схемы страницы, типа каждого текстового блока (заголовок, логотип, сноска, колонки, основной текст), а также определение связи и порядка этих блоков [1].

1. Краткий обзор методов сегментации и анализа страницы

Методы сегментации и анализа страницы разделяются на 3 главные категории – нисходящие, восходящие и гибридные [2].

При применении нисходящих методов сначала находится самый высокий уровень структур (обычно это колонки и графика), потом колонки разбиваются на параграфы, параграфы – на строки, и т.д. Процесс продолжается до тех пор, пока не достигается самый нижний уровень – символы. При использовании этого подхода необходима априорная информация о макете страницы. К этой категории также относятся неравномерное размазывание [4], методы проекции профиля [5], белые потоки [6], преобразование Фурье [7] и т.д.

Восходящие методы начинаются с поиска минимального элемента (например, пиксель). Затем пиксели комбинируются и формируют связные компоненты. Дальше из этих компонентов создаются символы, слова, фразы, параграфы и колонки. На первый взгляд, это более гибкий подход, но на практике он трудно применим из-за накапливания ошибок каждого этапа обработки. К этой категории относятся такие методы, как метод анализа связных компонентов [1], метод неравномерного сглаживания [4], метод наращивания областей [3] и нейронные сети [8].

Среди известных методов интересен метод «X-Y дерево», разработанный Наги (Nagy) [9]. В этом методе документ сегментируется через многошаговый процесс, и в результате строится дерево вложенных прямоугольных блоков. Сначала документ сегментируется на большие блоки путем горизонтального и вертикального разрезания. Дальше этот процесс повторяется для каждого блока. Особенность этого метода – полученное дерево соответствует логической структуре документа. Основной недостаток метода – допущение, что текстовые блоки имеют прямоугольную форму и хорошо разделяются прямоугольными полосами.

Метод неравномерного сглаживания [4] популярен и часто используется вместе с другими методами. В основном это – простая обработка изображений, которая выполняется следующим образом: изображение сканируется горизонтально и (или) вертикально, и цвет небольших фоновых областей изменяется на цвет изображения (цвет текста). В результате получается эффект размытия, при котором объединяются близкие символы, элементы документа. Это служит основанием и для других методов, таких как профильная проекция, группирование связных компонентов, нейронные сети и т.д. Главная недостаток метода – определение порогового размера малой области, цвет которой необходимо обновить. Если этот размер слишком маленький, размытие может быть недостаточным, чтобы получить ожидаемый эффект. На практике этот размер обычно определяется эмпирическим путем или через приблизительную оценку размеров зазора между текстовыми блоками, строками или размера шрифта печатных символов. Вычисление этих параметров также является недостатком других методов сегментации.

2. Разработанный метод

В данной работе разработан метод многошкальной сегментации страницы с использованием понятия *динамическая локальная карта связи* (ДЛКС).

ДЛКС изображения является результатом трансформации изображения в матрицу, размер которой равен размеру изображения. ДЛКС можно представлять как изображение. ДЛКС строится следующим образом: в каждой позиции пикселя исходного изображения элемент матрицы принимает значение 0, если пиксель черный (текстовый цвет). В противном случае элементу матрицы присваивается значение, равное расстоянию между двумя соседними черными пикселями в исходном изображении. В качестве расстояния можно выбрать минимальное или среднее значение вертикального и горизонтального измерения. Эксперименты показывают, что лучший результат получается при использовании минимального значения вертикального и горизонтального расстояния. Далее ДЛКС преобразуется в изображение оттенков серого цвета путем нормализации полученной матрицы так, чтобы ее элементы имели значения в интервале от 0 до 255. На рис. 1 представлен пример отображения ДЛКС.

Каждый ненулевой пиксель в ДЛКС представляет собой размер ближайшего локального зазора между двумя черными пикселями исходного изображения. Если выполнить бинаризацию ДЛКС с пороговым значением t , то все пиксели, имеющие значение меньше t , принимают значение 0. Эти пиксели, объединившись друг с другом, создают блоки (текстовые или графические). Так как ДЛКС – изображение серого оттенка, то любой пиксель может принимать значение от 0 до 255.

Бинаризация на каждом из этих значений разбивает исходное изображение на блоки. Чем меньше пороговое значение t , тем больше количество блоков. Если $t = 255$, все пиксели ДЛКС будут черными, а все изображение – одним блоком, который определяется как корневой блок. Далее производится бинаризация с пороговым значением меньше 255, и тогда получаются дочерние блоки. Если процесс повторяется для каждого блока, то в результате получается дерево разделения исходного документа. Корневой блок разбивается на некоторые блоки нижнего уровня. Каждый из получаемых блоков, в свою очередь, разбивается на блоки более низкого уровня. Блоки одного уровня можно упорядочить с помощью их относительного положения. Можно использовать, например, позиции ограничивающих их прямоугольников для определения их упорядоченных положений.

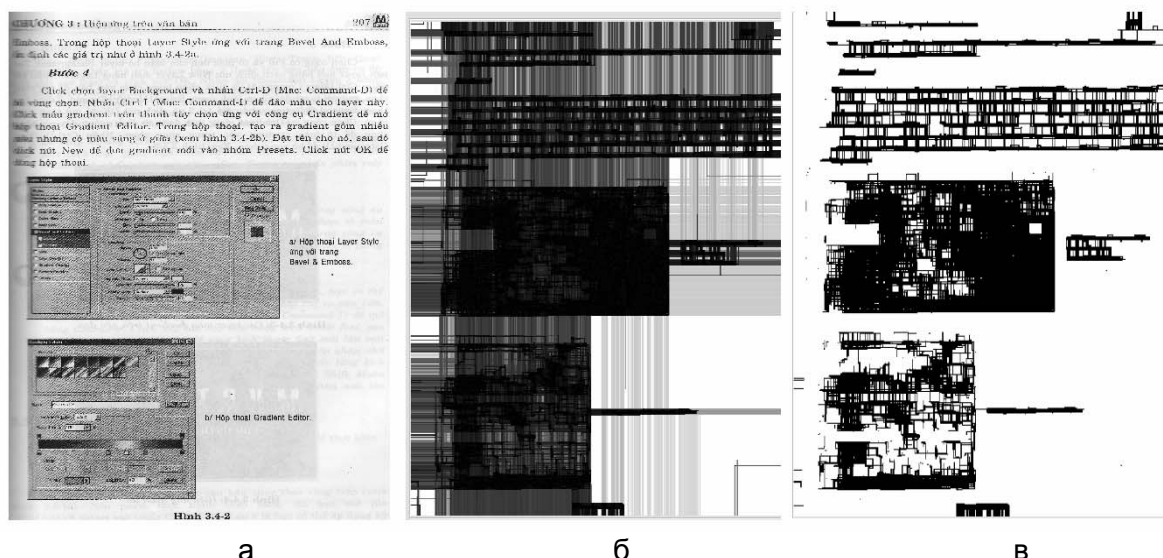


Рис. 1. Пример ДЛКС-изображения: а – исходное изображение; б – ДЛКС-изображение серого оттенка; в – ДЛКС-изображение после бинаризации

3. Результаты и их обсуждение

3.1. Исследование свойств порогового значения

Как было сказано выше, мы имеем 256 возможных выборов для пороговых значений. Поскольку процессы преобразования исходного изображения в ДЛКС и бинаризации – тяжеловесные, выполнение бинаризации со всеми возможными разными пороговыми значениями занимает большое вычислительное время. Поэтому возникает задача выбора потенциальных пороговых значений.

При анализе этой задачи было просканировано 15 печатных документов из разных источников (в том числе научные статьи, газеты, книги). Для каждого полученного изображения была построена диаграмма зависимости количества связанных компонентов (блоков) от порогового значения (рис. 2).

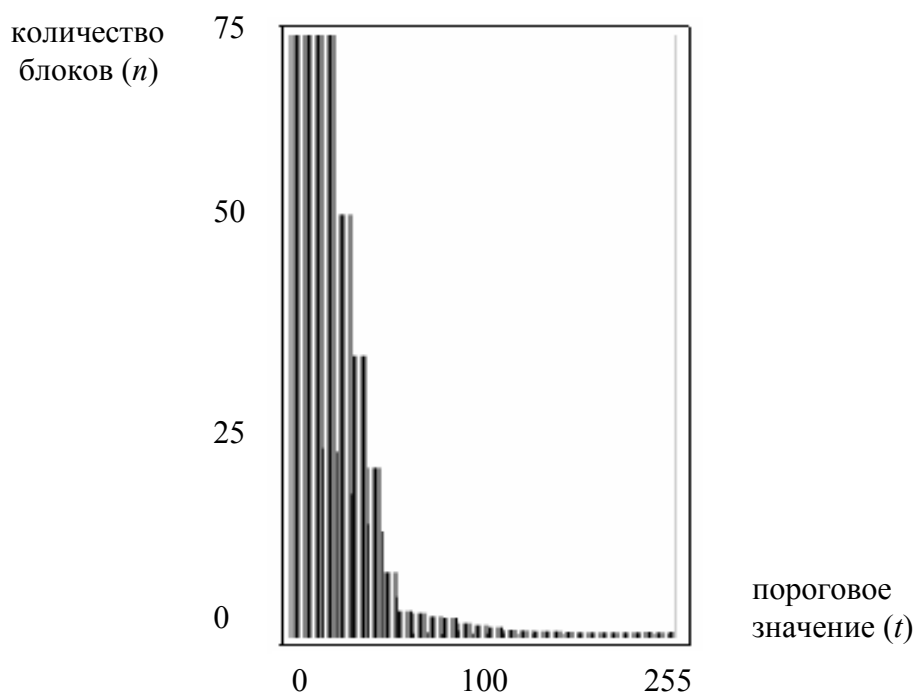


Рис. 2. Зависимость количества блоков от порогового значения

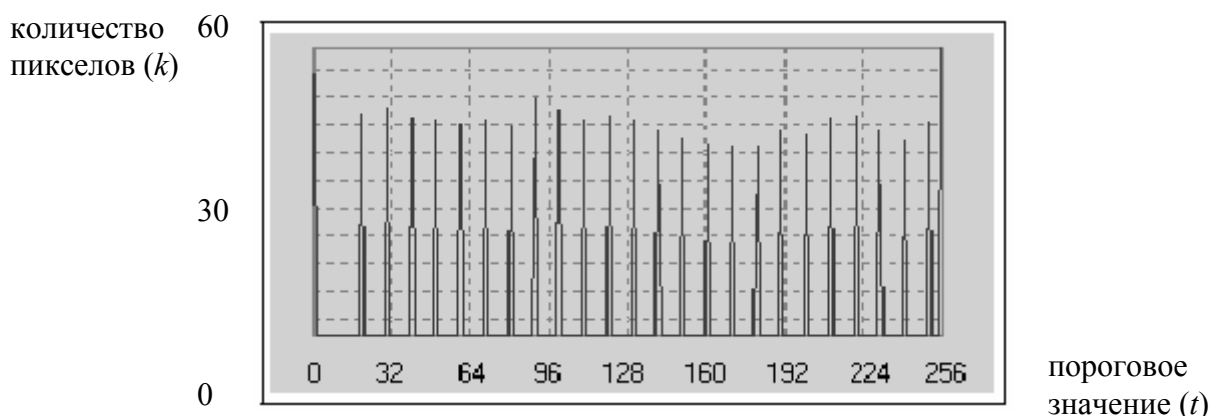


Рис. 3. Распределение значений пикселей ДЛКС

Оказалось, что количество блоков не изменяется непрерывно. Иначе говоря, в некоторой окрестности порогового значения t количество блоков изменяется незначительно или вообще не изменяется. Другой вывод из диаграммы – в начале (в окрестности $t = 0$) количество блоков всегда велико, но очень быстро уменьшается до некоторо-

го разумного значения. Это свойство можно использовать для нахождения нижней границы порогового значения t .

Второе исследование – это построение диаграммы распределения расстояний. Так как значения пикселей ДЛКС прямо связаны с расстояниями, то было вычислено распределение значений пикселей ДЛКС (рис. 3). Из диаграммы видно, что расстояния распределяются неравномерно и существуют некоторые пиковые значения.

Еще один интересный результат из экспериментов – количество пиков всегда находится в окрестности 25 значений, хотя документы, используемые в экспериментах, относятся к различным областям. Значения, находящиеся между пиковыми значениями, могут быть пороговыми.

3.2. Построение иерархического дерева сегментации

Блоки документа сегментируются в иерархическом порядке. Сначала необходимо выбрать набор пороговых значений t . Эксперименты показывают, что 10 пороговых значений достаточно для сегментации всех документов. Первое пороговое значение в наборе – это минимальное пороговое значение, при котором наблюдается резкое уменьшение количество блоков (см. рис. 2). В экспериментах это значение всегда находится в окрестности 50. Потом выбирается минимальное значение, находящееся между пиковыми значениями и большее первого порогового значения, в качестве потенциального порогового значения. Если в окрестности этого значения наблюдается резкое изменения количества блоков, то это значение подходит для порогового значения. Процесс повторяется для всех долин между пиковыми значениями.

Бинаризации с помощью минимального значения из набора пороговых значений генерирует блоки высшего уровня. Далее для каждого блока выполняется бинаризация, используя следующее пороговое значение из набора. В результате этого процесса получается иерархическое дерево сегментации, пример которого приведен на рис. 4.

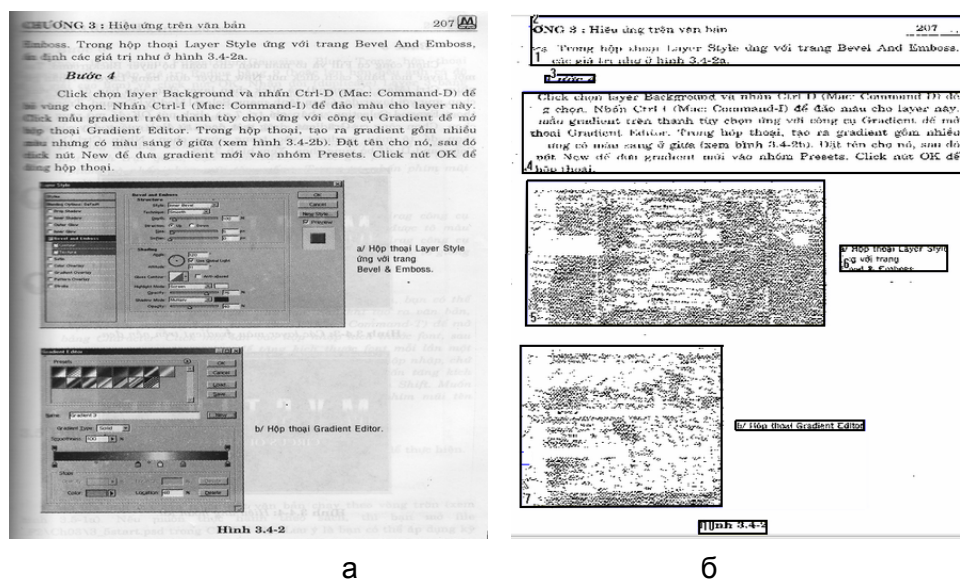


Рис. 4. Пример многошкальной сегментации: а – исходное изображение; б – результат сегментации и фильтрации

Заключение

На основе обзора методов сегментации и анализа страницы при решении задач оптимального распознавания текста представлен разработанный авторами метод многошкальной сегментации страницы на базе понятия «Динамическая локальная карта связи». Приведены результаты использования этого метода.

Литература

1. Jain A., Yu B. Document representation and its application to page decomposition // Pattern Analysis and Machine Intelligence. – 1998. – № 3. – С. 294–308.
2. Okun O., Doermann, Pietikainen M. Page Segmentation and Zone Classification: The State of the Art. LAMP-TR-036 (Technical Report). – University of Maryland, 1999.
3. Jain A. Fundamentals of digital image processing. – Prentice Hall, 1990. – 421 с.
4. Wong K.Y., Casey R.G., Wahl F.M. Document analysis system // IBM Journal of Research and Development. – 1982. – V. 26. – №6. – P. 647–656.
5. Wang D., Srihari S. Classification of newspaper image blocks using texture analysis // Computer Vision, Graphics, and Image Processing. – 1989. – V. 47. – С. 327–352.
6. Pavlidis T., Zhou J. Page segmentation by white streams // Proceedings of 1st International Conference Document Analysis and Recognition (ICDAR), International Association Pattern Recognition, 1991. – P. 945–953.
7. Hose M., Hoshino Y. Segmentation method of document images by two-dimensional fourier transformation // System and Computers in Japan. – 1985. – V. 3. – С. 16–20.
8. Tan C., Zhang Z. Text block segmentation using pyramid structure // SPIE Document Recognition and Retrieval, San Jose, USA. – 2001. – V. 8. – С. 297–306.
9. Nagy S.S.G., Stoddard S. Document analysis with expert system // Proceedings of Pattern Recognition in Practice II, 1985. – P. 31–33.

Хоанг Зянг

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, giangdebug@yahoo.com

Коробейников Анатолий Григорьевич

— Институт земного магнетизма, ионосферы и распространения радиоволн им. Н.В. Пушкова, зам. директора, доктор технических наук, профессор, Korobeynikov_A_G@mail.ru

УДК 681.4

ВЛИЯНИЕ ВНЕШНИХ ВОЗДЕЙСТВИЙ НА DC-КОЭФФИЦИЕНТ МАТРИЦЫ ДИСКРЕТНО-КОСИНУСНОГО ПРЕОБРАЗОВАНИЯ В ПОЛУТОНОВЫХ ИЗОБРАЖЕНИЯХ

Н.Н. Прохожев, О.В. Михайличенко, А.Г. Коробейников

Цель работы – постановка и анализ результатов экспериментов, проводимых с целью выявления влияния различных внешних воздействий на коэффициент отображающий общую яркость блока (DC-коэффициент) матрицы дискретно-косинусного преобразования (ДКП) в неподвижных полутоновых изображениях. Целью анализа устойчивости DC-коэффициентов является определение их теоретической пригодности для использования в качестве основы для разработки стеганографических алгоритмов повышенной устойчивости. В качестве внешних воздействий используются JPEG сжатие с потерями, гауссовский белый шум, масштабирование и фильтрация.

Ключевые слова: стеганография, алгоритмы частотной области, DC-коэффициенты.

Введение

Стеганографические алгоритмы, производящие встраивание скрываемой информации в частотную область изображений, получили широкое распространение в силу некоторых выгодных отличий от остальных стеганографических алгоритмов. К сильным сторонам данного вида алгоритмов, прежде всего, следует отнести возможность

встраивать информацию в изображения-контейнеры, сжатые форматом JPEG, который является одним из наиболее распространенных форматов хранения и передачи мультимедиа-контента на сегодняшний день. Также к преимуществам данного вида алгоритмов можно отнести и достаточно хорошую устойчивость к различного рода внешним воздействиям или атакам на изображение-контейнер.

С основным недостатком данного вида алгоритмов, а именно необходимостью значительной вычислительной мощности для реализации достаточно большого объема сложных вычислений, успешно справляется обычный современный офисный компьютер. Предметом исследований данной работы является анализ ДС коэффициента матрицы ДКП с целью выявления его устойчивости к внешним воздействиям и определение его теоретической пригодности для использования в стеганосистемах с высокой вероятностью внешних атак.

Частотные свойства матрицы ДКП коэффициентов

В основе большинства стеганографических алгоритмов частотной области лежит дискретно-косинусное преобразование (ДКП). Такие алгоритмы реализуют одинаковую последовательность действий с небольшими, специфическими для конкретного алгоритма, отличиями. Исходное изображение-контейнер предварительно разбивается на блоки, как правило, размером 8×8 пикселей, которые в дальнейшем и подвергаются дискретно-косинусному преобразованию. Результатом такого преобразования является матрица коэффициентов, представленная на рис. 1.

Коэффициент в левом верхнем углу матрицы содержит информацию о яркости всего сегмента (так называемый ДС-коэффициент), остальные коэффициенты матрицы называются АС-коэффициентами. Следует отметить, что коэффициенты низкочастотных компонент располагаются ближе к верхнему левому углу, в то время как коэффициенты высокочастотных компонент сгруппированы в правой нижней части матрицы. Низкочастотные коэффициенты содержат преобладающую часть энергии изображения, в то время как высокочастотные компоненты наиболее уязвимы для внешних воздействий [1]. В связи с изложенным авторы большинства алгоритмов считают пригодными для встраивания только среднечастотные коэффициенты, т.е. второстепенную диагональ матрицы ДКП коэффициентов.

	1	2	3	4	5	6	7	8
1	1603	203	11	45	-30	-14	-14	-7
2	108	-93	10	49	27	6	8	2
3	-42	-20	-6	16	17	9	3	2
4	56	69	7	-25	-10	-5	-2	-2
5	-33	-21	17	8	3	-4	-5	-3
6	-16	-14	8	2	-4	-2	1	1
7	0	-5	-6	-1	2	3	0	1
8	9	5	-6	-9	0	3	3	1

	- НЧ компоненты;
	- СЧ компоненты;
	- ВЧ компоненты

Рис. 1. Матрица ДКП коэффициентов

DC коэффициент как объект для встраивания информации не рассматривается во все в силу некоторых его особенностей. Первой и наиболее веской причиной, делающей DC коэффициент не столь привлекательным с точки зрения стеганографии, является то, что система человеческого зрения наиболее чувствительна к изменению яркости, вторая причина заключается в том, что этот коэффициент является уникальным в пределах ДКП матрицы, что усложняет задачу устойчивого внедрения встраиваемой информации. Столь же затруднительно оперировать коэффициентами соседних блоков, используя для встраивания их пары или тройки, в силу того, что они могут очень сильно отличаться по своим значениям. Тем не менее, абстрагируясь от деталей алгоритма встраивания, проведем анализ изменений DC коэффициентов в результате различных внешних воздействий на изображение, которые можно рассматривать как атаки.

Оценка устойчивости DC коэффициентов к внешним воздействиям

Влияние внешних воздействий на DC коэффициенты можно выявить, следуя приведенному ниже алгоритму.

- (1) Исходное изображение разбивается на блоки размером 8×8 пикселей.
- (2) Производится ДКП и определяются значения всех DC коэффициентов изображения.
- (3) Изображение подвергается внешнему воздействию.
- (4) Производится ДКП и определяются значения всех DC коэффициентов искаженного изображения.
- (5) Вычисляется разница между DC коэффициентами оригинального и искаженного изображений.

В силу того, что изменения в результате одного и того же внешнего воздействия у DC коэффициентов различных блоков могут сильно различаться, целесообразно оценивать максимальное изменение и усредненное по всем DC коэффициентам изображения.

Внешние воздействия на изображение

В качестве внешних воздействий применим сжатие JPEG с потерями, зашумление изображения белым гауссовским шумом, масштабирование и фильтрацию. Все эти воздействия довольно сильно и, что более важно, необратимо искажают изображение. Исследования проводились на 10-ти естественных полутоновых изображениях размера 512×512 пикселей. Среднее значение DC коэффициентов по всем изображениям составило 981.

JPEG сжатие с потерями

Для анализа изменений DC коэффициентов при JPEG сжатии изображения подвергались JPEG сжатию во всем диапазоне значений коэффициента качества JPEG. Результат представлен на рис. 2. Как можно видеть из графика, среднее изменение DC коэффициентов почти на всем диапазоне не превышает 20, что составляет 2 %, и только на низких значениях коэффициента качества (ниже 10) имеется существенный рост. Следует подчеркнуть что изображение, сжатое с коэффициентом качества ниже 20, будет сильно деградировано и вызывает большое сомнение в возможности его использования в коммерческих целях.

Гауссовский белый шум

Для анализа изменений DC коэффициентов при зашумлении изображения в последнее вносился белый гауссовский шум с нулевым средним значением и разными значениями отклонения, изменяющимися от 0 в сторону возрастания до тех пор, пока деградация изображения не достигла уровня, совершенно неприемлемого для исполь-

зования в коммерческих целях. Результат представлен на рис. 3. Следует отметить очень незначительные изменения значений DC коэффициентов.

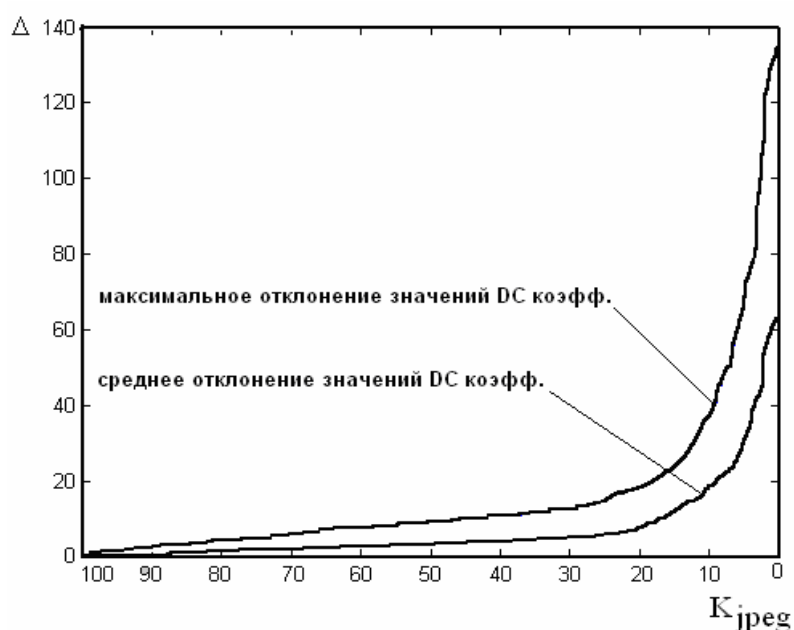


Рис. 2. Изменение DC коэффициентов изображения после сжатия JPEG с потерями

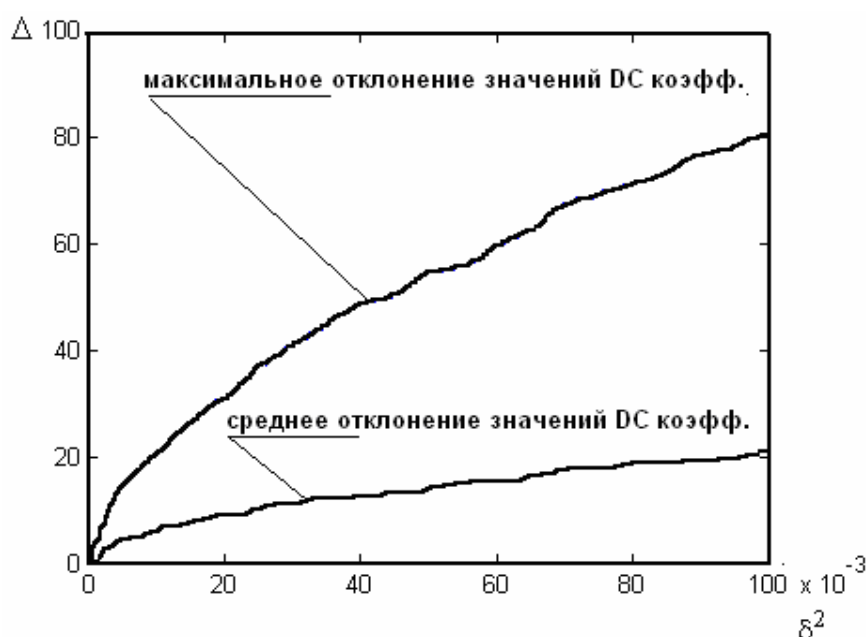


Рис. 3. Изменение DC коэффициентов изображения, подвергнутого зашумлению

Масштабирование

В ходе эксперимента с масштабированием изображение-контейнер сжималось до различных размеров вплоть до 80%-го сжатия, т.е. в 5 раз. Совершенно очевидно, что вычислять DC коэффициенты сжатого изображения и сравнивать их с коэффициентами оригинального изображения бессмысленно, по крайней мере, не уменьшив пропорционально размер блоков, на которые разбивается сжатое изображение. Однако такой задачи и не ставилось, поскольку изображение всегда можно восстановить до размеров исходного, а уж затем вычислять значения DC коэффициентов. Это и было проделано в ходе эксперимента, результаты которого представлены на рис. 4.

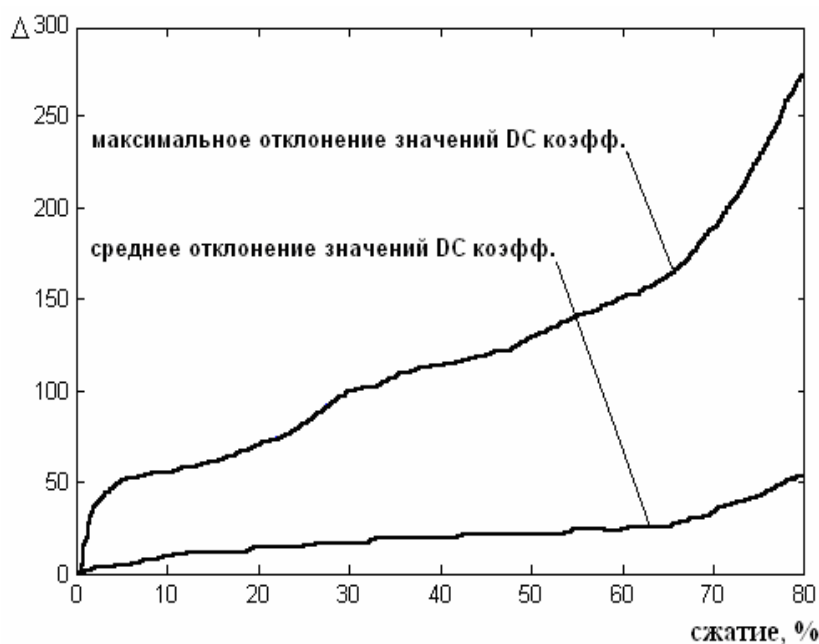


Рис. 4. Изменение DC коэффициентов изображения, подвергнутого масштабированию

Применение фильтров

Фильтрацию также можно отнести к разряду наиболее вероятных внешних воздействий на стеганосистему. Из большого многообразия фильтров для цифровых изображений были выбраны три вида фильтров: низкочастотный гауссовский фильтр, усредняющий фильтр с размером окна 3×3 пикселя и контрастный фильтр, повышающий резкость изображения, с тем же размером окна. Результаты эксперимента представлены в таблице.

Фильтр	Максимальное изменение DC коэффициента, Δ	Среднее изменение DC коэффициента, Δ
Низкочастотный	41	5.5
Усредняющий	126	17
Контрастный	211	34

Таблица. Изменение DC-коэффициентов изображения подвергнутого фильтрации

Выводы

Можно утверждать, что DC-коэффициенты имеют высокую устойчивость к внешним воздействиям, чем явно превосходят AC-коэффициенты. Среднее значение изменений DC-коэффициентов практически для всех внешних воздействий с большой интенсивностью, рассмотренных в данной работе, не превышает 2%, что гарантирует устойчивость внедренной информации. Визуальные искажения изображения контейнера при таких незначительных изменениях, скорее всего, останутся неразличимы для системы человеческого зрения. Следует отметить, что для успешного внедрения информации в DC-коэффициенты необходимо осуществлять качественный отбор коэффициентов, наименее подверженных внешним воздействиям, что, хоть и повлечет некоторое снижение пропускной способности, может значительно повысить устойчивость и скрытность внедрения.

Литература

1. Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография: Теория и практика. – М.: МК-Пресс, 2006. – 283 с.

Прохожеев Николай Николаевич

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, karaja2@yandex.ru

Михайличенко Ольга Викторовна

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, ст. преподаватель, 19791109@list.ru

Коробейников Анатолий Григорьевич

— Институт земного магнетизма, ионосферы и распространения радиоволн им. Н.В. Пушкова, зам. директора, доктор технических наук, профессор, Korobeynikov_A_G@mail.ru

УДК 004.932

ИССЛЕДОВАНИЕ ДИНАМИКИ ПРОЦЕССОВ МЕТОДОМ ВЫЧИСЛЕНИЯ МЕР КОЛИЧЕСТВЕННОГО РЕКУРРЕНТНОГО АНАЛИЗА В ОКНЕ, СМЕЩАЕМОМ ВДОЛЬ ГЛАВНОЙ ДИАГОНАЛИ РЕКУРРЕНТНОЙ ДИАГРАММЫ

В.Б. Киселев, Б.А. Крылов

Рассматривается вычисление мер количественного анализа рекуррентной диаграммы в окне, смещаемом вдоль главной диагонали. Приводятся примеры применения методики к модельным и природным данным. Дается оценка результатов и возможностей применения.

Ключевые слова: рекуррентный анализ, рекуррентные диаграммы, размер окна.

Введение

Исследования сложных систем, как природных, так и искусственных, показали, что в их основе лежат нелинейные процессы, тщательное изучение которых необходимо для понимания и моделирования сложных систем.

Рекуррентный анализ [1,2,3] — достаточно молодой и динамично развивающийся подход к анализу сложных систем, не требующий длинных или стационарных временных рядов. Рекуррентные диаграммы позволяют судить о характере протекающих в системе процессов, наличии и влиянии шума, дрейфа, наличии состояний повторения и замирания (ламинарность), совершении экстремальных событий, наличии скрытой периодичности и цикличности. Количественный анализ рекуррентных диаграмм позволяет сопоставить диаграмме некоторые численные меры, основанные на плотности рекуррентных точек, распределениях длин диагональных и горизонтальных (вертикальных) линий. Следует отметить, что пока не создано удовлетворительной теории применения рекуррентных диаграмм и их количественных мер; этот метод сам по себе представляет собой поле для исследований.

Вычисление мер в окне, смещаемом вдоль главной диагонали рекуррентной диаграммы, позволит рассмотреть эволюцию изучаемого процесса через эволюцию изменения выбранных мер рекуррентной диаграммы, и таким образом сделать основанные на количественных показателях выводы об изменениях в динамике исследуемой системы. Приведены примеры вычисления оконных мер для диаграмм, построенных по модельным и реальным системам, дана оценка возможностей использования.

Рекуррентные диаграммы

Рекуррентные диаграммы были предложены для отображения траектории $\vec{x}_t \in \mathcal{R}^m$ ($i = 1 \dots N$) в m -мерном фазовом пространстве на двумерную двоичную матрицу размером $N \times N$. Единица в ячейке матрицы соответствует повторению состояния

(проход траектории через одну и ту же точку фазового пространства) при некотором времени i в некоторое другое время j , а обе координатные оси диаграммы являются осями времени. Математически это выражается следующим образом:

$$\mathbf{R}_{i,j}^{m,\varepsilon_i} = \Theta(\varepsilon_i - \|\tilde{x}_i - \tilde{x}_j\|), \quad \tilde{x} \in \mathfrak{R}^m, \quad i, j = 1 \dots N, \quad (1)$$

где N – количество рассматриваемых состояний x_i , ε_i – размер окрестности точки $\tilde{x}$ в момент i , $\|\cdot\|$ – норма (расстояние) и $\Theta(\cdot)$ – функция Хэвисайда.

Графически рекуррентная диаграмма может быть представлена монохромным изображением, где единице соответствует черная точка. При наличии только одномерного ряда u_t эквивалентная траектория в m -мерном фазовом пространстве может быть восстановлена по методу временных задержек Такенса [4], $\mathbf{x}(t) = (u_i, u_{i+\tau}, \dots, u_{i+(m-1)\tau})$, где m – размерность вложения, τ – временная задержка (реальная временная задержка определяется как $\tau \cdot \Delta t$). Топологические структуры восстановленной траектории сохраняются, если $m \leq 2d + 1$, где d – размерность аттрактора. На практике оказывается, что в большинстве случаев аттрактор может быть восстановлен и при $m \leq 2d$ [5]. Задержка τ , как правило, выбирается априорно.

Размер окрестности ε_i определяет радиус окрестности в фазовом пространстве с центром в точке x_i . Если точка x_j попадает внутрь данной окрестности, то такое состояние считается подобным состоянию x_i , и, таким образом, на диаграмме устанавливается $\mathbf{R}_{i,j} = 1$. Радиус ε_i может быть постоянным для всех x_i либо определяться для каждой точки индивидуально, чтобы в получаемую окрестность всегда попадало определенное количество подобных состояний. В этой работе используется постоянное значение ε_i , что приводит к получению симметричной рекуррентной диаграммы относительно линии $\mathbf{R}_{i,j} = 1$ ($i = j$). Для вычисления расстояний между точками фазового пространства используется максимальная норма $L_\infty \equiv \sup |v_k|$.

Рисунок рекуррентной диаграммы отображает поведение процесса во времени и позволяет сделать выводы о его характере по ее топологии и текстуре.

Количественный анализ рекуррентных диаграмм

Збилут (Zbilut) и Вебер (Webber) разработали количественный анализ рекуррентных диаграмм (recurrence quantification analysis, RQA) [6] для определения численных показателей рекуррентной диаграммы. Они предложили меры, использующие плотность рекуррентных точек и диагональные структуры диаграммы: показатель подобия (RR), детерминизм (DET), максимальная длина диагональных линий (L), энтропия (ENTR), тренд (TREND). Несколько позже Марван (Marwan) предложил [7] меры, основанные на горизонтальных (вертикальных) структурах рекуррентных диаграмм – замиране (LAM) и показатель задержки (TT).

Как правило, для вычисления мер используются рекуррентные диаграммы с постоянным значением порога ε . Мера рекуррентности (recurrence rate, RR),

$$RR = \frac{1}{N^2} \sum_{i,j=1}^N \mathbf{R}_{i,j}^{m,\varepsilon}, \quad (2)$$

показывает плотность рекуррентных точек, просто подсчитывая их, включая линию идентичности. В пределе,

$$P_{\bullet} = \lim \frac{1}{N^2} \sum_{i,j=1}^N \mathbf{R}_{i,j}^{m,\varepsilon},$$

данная мера показывает вероятность нахождения рекуррентной точки в рекуррентной диаграмме (вероятность повторения состояния).

Следующая мера рассматривает диагональные линии. Частотное распределение длин l диагональных линий рекуррентной диаграммы $P^{\varepsilon}(l) = \bigcup_{i=1}^N P_i(l)$, где N – абсолютное количество диагональных линий (каждая линия считается только один раз). Процессы со стохастическим поведением могут порождать очень короткие диагонали либо вообще не порождать их, в то время как детерминистские процессы дают длинные диагонали и малое количество отдельных рекуррентных точек. Таким образом, отношение рекуррентных точек, составляющих диагональные структуры, к общему количеству рекуррентных точек,

$$DET = \frac{\sum_{l=l_{\min}}^N l P^{\varepsilon}(l)}{\sum_{i,j}^N \mathbf{R}_{i,j}^{m,\varepsilon}}, \quad (3)$$

называется мерой детерминизма (determinism, DET), или предсказуемости, системы. Следует отметить, что эта мера не имеет значения реального детерминизма процесса. Пороговое значение минимальной длины $l_{\min}$ исключает диагональные линии, образованные тангенциальным движением траектории в фазовом пространстве. Очевидно, что $l_{\min} = 1 \Rightarrow DET = 1$. Диагональные структуры показывают время, в течение которого участок траектории подходит достаточно близко к другому участку траектории. Таким образом, эти линии позволяют судить о расхождении элементов траектории. Средняя длина диагональных линий

$$L = \frac{\sum_{l=l_{\min}}^N l P^{\varepsilon}(l)}{\sum_{l=l_{\min}}^N P^{\varepsilon}(l)} \quad (4)$$

– это среднее время, в течение которого два участка траектории проходят близко один к другому, и может рассматриваться как среднее время предсказуемости. Находят применение максимальная длина диагональных структур, а также ее инверсия – дивергенция (divergence, DIV):

$$L_{\max} = \max(\{l_i; i = 1 \dots N_l\}), \quad DIV = \frac{1}{L_{\max}}. \quad (5)$$

Установлено, что длины диагональных линий соотносятся с наибольшим положительным показателем Ляпунова, если он существует для рассматриваемой системы [1]. Разными авторами были предложены методы оценки максимального положительного показателя Ляпунова с использованием длин диагональных линий. Мера энтропии (entropy, ENTR) соотносится с энтропией Шеннона (Shannon) частотного распределения длин диагональных линий

$$ENTR = - \sum_{l=l_{\min}}^N p(l) \ln p(l), \quad \text{где } p(l) = \frac{P^{\varepsilon}(l)}{\sum_{l=l_{\min}}^N P^{\varepsilon}(l)}, \quad (6)$$

и отражает сложность детерминистской составляющей в системе. Следующая мера представляет собой, по сути, отношение между DET и RR и может быть вычислена из частотного распределения длин диагональных линий:

$$RATIO = N^2 \frac{\sum_{l=l_{\min}}^N l P^{\varepsilon}(l)}{\left(\sum_{l=1}^N l P^{\varepsilon}(l) \right)^2}. \quad (7)$$

Эвристическое изучение физиологических систем показало [8], что эта мера может быть использована для распознавания фазовых переходов в случаях, когда RR уменьшается, а DET остается постоянной.

Рассмотрим точку траектории $\vec{x}_i$ и множество ассоциированных с ней рекуррентных точек $R_i = \{\vec{x}_j : \mathbf{R}_{i,j} = 1; j \in [1 \dots N]\}$. Определим подмножество точек $r_i = \{\vec{x}_k \in \mathbf{R}_i : (\mathbf{R}_{i,k} \cdot \mathbf{R}_{i,k+1}) + (\mathbf{R}_{i,k} \cdot \mathbf{R}_{i,k-1}) > 0; l \in [1 \dots N]; \mathbf{R}_{i,0} = \mathbf{R}_{i,N+1} = 0\}$, которое содержит рекуррентные точки, формирующие горизонтальные линии в строке i . Определим теперь длины v всех соединенных в s_i подмножеств $\{\vec{x}_j \notin s_i; \vec{x}_{j+1}, \dots, \vec{x}_{j+v} \in s_i; \vec{x}_{j+v+1} \notin s_i\}$. Обозначим $P_i(v) = \{v_k; k = 1, 2, \dots, K\}$ – множество длин присутствующих в s_i подмножеств; таким образом, выражение $P^\varepsilon(v) = \bigcup_{i=1}^N P_i(v)$ дает нам распределение длин горизонтальных линий $P^\varepsilon(v)$ для всей рекуррентной диаграммы. Мера замирания (laminarity, LAM)

$$LAM = \frac{\sum_{v=v_{\min}}^N v P^\varepsilon(v)}{\sum_{i,j}^N \mathbf{R}_{i,j}^{m,\varepsilon}} \quad (8)$$

определяется отношением количества рекуррентных точек, образующих горизонтальные линии, к общему количеству рекуррентных точек. LAM характеризует наличие состояний замирания системы (т.е. когда движение системы по фазовой траектории останавливается или продвигается очень медленно). Средняя длина горизонтальных структур

$$TT = \frac{\sum_{v=v_{\min}}^N v P^\varepsilon(v)}{\sum_{v=v_{\min}}^N P^\varepsilon(v)} \quad (9)$$

называется показателем задержки (trapping time, TT) и характеризует среднее время, которое система может провести в более-менее неизменном состоянии.

Мера чистоты сигнала

Влияние стохастической составляющей процесса приводит к появлению на диаграмме отдельно стоящих точек и очень коротких диагональных линий. В основном стохастические процессы (например, обобщенное броуновское движение), как уже упоминалось выше, вообще могут не порождать длинных диагоналей, а если таковые и обнаруживаются, то их появление носит случайный характер. Сказанное позволяет ввести меру отношения количества точек, формирующих диагональные линии длиной $l < l_{\min}$, к количеству точек, формирующих диагональные линии $l \geq l_{\min}$,

$$CLEAN = \frac{\sum_{l=1}^{l_{\min}-1} l P^\varepsilon(l)}{\sum_{l=l_{\min}}^N l P^\varepsilon(l)}, \quad (10)$$

которое называется мерой чистоты (cleanness, CLEAN) и показывает влияние стохастической составляющей процесса. Очевидно, что преобладание последней приведет к росту значения CLEAN.

Использование окна для вычисления мер

При изучении реальных систем зачастую требуется обнаружить фазовые переходы между режимами системы. Как правило, их можно определять непосредственно по рекуррентной диаграмме. Однако, если стоит задача сравнения нескольких измерений, классификации и т.п., становится необходимым иметь некоторые количественные по-

казатели. В качестве примера можно привести изучение различий в динамике солнечной и геомагнитной активностей. В работах [9, 10] рассматривался фазовый переход в районе 1934–37-го гг. от периода умеренной солнечной активности к более высокой. Вычисление некоторых количественных мер диаграмм, построенных для диапазонов до и после упомянутого рубежа, показало существенное различие полученных значений.

Последовательное вычисление количественных мер фрагмента диаграммы, ограниченного окном некоторого размера $S < N$, перемещаемого вдоль главной диагонали диаграммы $\mathbf{R}_{i=j} = 1$, позволит получить динамику изменения значений вычисленных мер. Экстремумы на графиках изменения мер будут свидетельствовать о точках фазовых переходов. Вычисления мер в окне

$$\mathbf{W}_{i,j}^{w,S} = \mathbf{R}_{i+w,j+w}, \quad i, j \in [1 \dots S], \quad w \in [1 \dots n = N - S]$$

ничем не отличаются от вычисления мер для всей рекуррентной диаграммы. Частотные распределения длин диагональных и горизонтальных линий принимают вид $P_w^{\varepsilon,S}(l) = \bigcup_{i=1}^S P_i(l)$ и $P_w^{\varepsilon,S}(v) = \bigcup_{i=1}^S P_i(v)$ соответственно.

Выражения для мер, вычисляемых в окне: $RR_w = \frac{1}{S^2} \sum_{i,j=1}^S \mathbf{W}_{i,j}^{w,S}$ – для рекуррентности

$$RR; DET_w = \frac{\sum_{l=l_{\min}}^S l P_w^{\varepsilon,S}(l)}{\sum_{i,j}^S \mathbf{W}_{i,j}^{w,S}} \text{ – для детерминизма DET; } L_w = \frac{\sum_{l=l_{\min}}^S l P_w^{\varepsilon,S}(l)}{\sum_{l=l_{\min}}^S P_w^{\varepsilon,S}(l)} \text{ – для сред-}$$

ней длины диагональных линий L ; $L_{\max_w} = \max(\{l_i; i = 1 \dots S_l - 1\})$, $DIV_w = \frac{1}{L_{\max_w}}$ – для

дивергенции DIV; $ENTR_w = - \sum_{l=l_{\min}}^S p_w^S(l) \ln p_w^S(l)$, где $p_w^S(l) = \frac{P_w^{\varepsilon,S}(l)}{\sum_{l=l_{\min}}^S P_w^{\varepsilon,S}(l)}$ – для энтро-

пии ENTR; $RATIO_w = S^2 \frac{\sum_{l=l_{\min}}^S l P_w^{\varepsilon,S}(l)}{\left(\sum_{l=1}^S l P_w^{\varepsilon,S}(l)\right)^2}$ – для соотношения RATIO;

$$LAM_w = \frac{\sum_{v=v_{\min}}^S v P_w^{\varepsilon,S}(v)}{\sum_{i,j}^S \mathbf{W}_{i,j}^{w,S}} \text{ – для замирования LAM; } TT_w = \frac{\sum_{v=v_{\min}}^S v P_w^{\varepsilon,S}(v)}{\sum_{v=v_{\min}}^S P_w^{\varepsilon,S}(v)} \text{ – для показателя}$$

задержки TT; $CLEAN_w = \frac{\sum_{l=l_{\min}}^{l_{\min}-1} l P_w^{\varepsilon,S}(l)}{\sum_{l=l_{\min}}^S l P_w^{\varepsilon,S}(l)}$ – для показателя чистоты CLEAN. Таким образом,

в результате вычислений набора мер в смещаемом вдоль главной диагонали окне мы получаем набор векторов мер $\mathbf{Q}^n$, где $n = N - S$ – количество выполненных вычислений.

Выбор размера окна априорен. Большой размер дает более гладкие графики мер за счет невысокого влияния малозаметных событий во всем объеме данных. Меньший размер дает более четкие графики, которые, однако, могут быть отягощены излишним влиянием шумов и флуктуаций. Экспериментально удалось установить, что хорошие результаты дает применение окна размером $S \approx N/5$, при этом он не должен быть кратен основному периоду сигнала.

Применение к модельным данным

В качестве примера анализа модельных данных рассмотрим рекуррентную диаграмму для системы Лоренца (рис. 1).

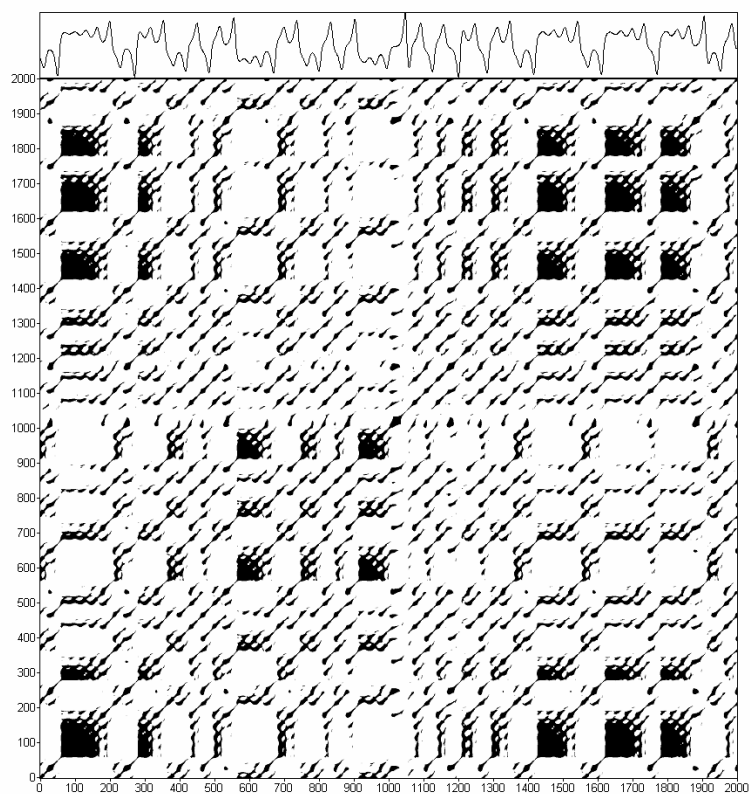


Рис. 1. Рекуррентная диаграмма трех компонент системы Лоренца. Длина ряда $N = 2000$ точек, $\varepsilon = 0,2$

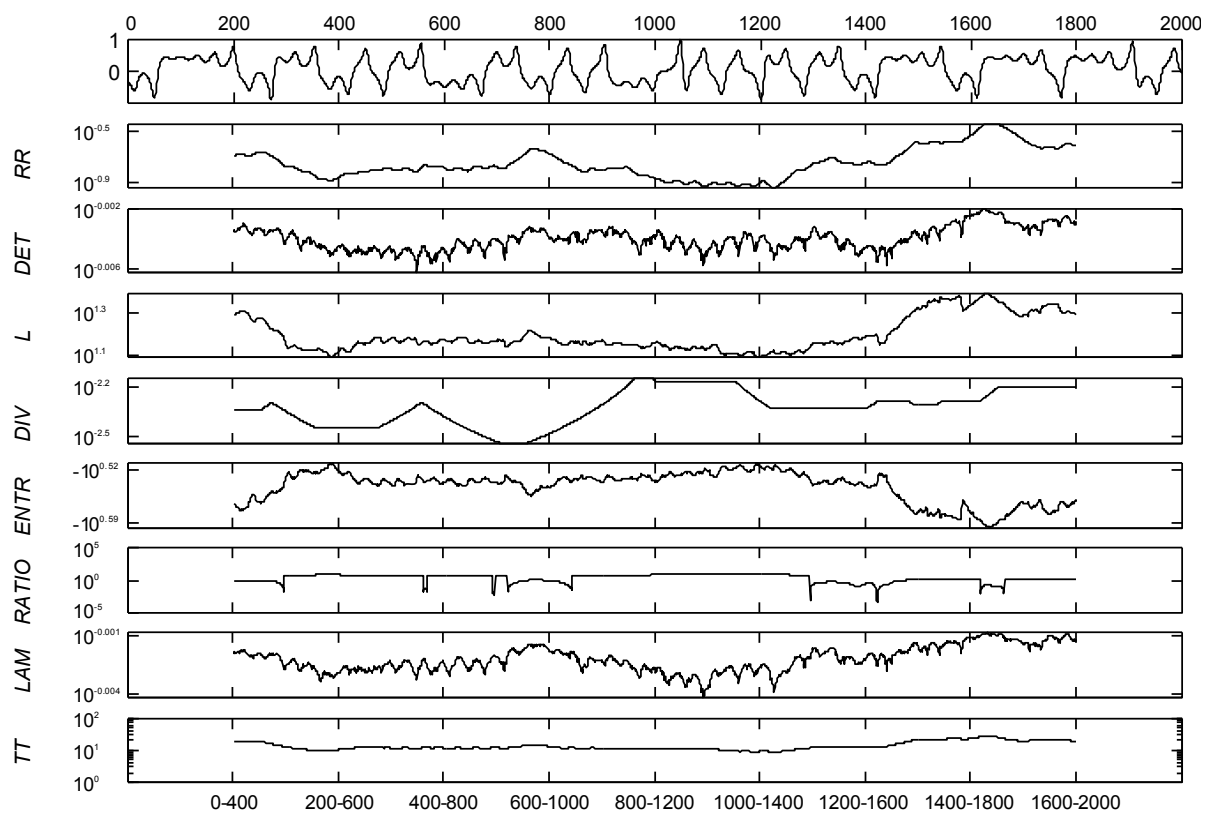


Рис. 2. Графики эволюции мер относительно X-компоненты системы Лоренца

По диаграмме проведем расчет мер в окне размером $S = 400$, после чего построим графики эволюции мер, расположенные относительно X -компоненты системы Лоренца (рис. 2). Мере *CLEAN* не будем включать в данный расчет по причине гладкости траектории модельного ряда. По оси Y графики мер изобразим в логарифмическом масштабе. Подписи оси X для графиков мер изобразим в виде диапазонов. Левое число диапазона обозначает отступ от левого нижнего угла диаграммы, правое – верхнюю правую точку, охваченную окном.

Очевидна реакция мер на изменения структуры диаграммы. Действительно, например в районе линий с координатами 195 и 890 на диаграмме можно видеть достаточно резкое изменение ее структуры, что свидетельствует о смене режима. Меры *RR*, *L*, *ENTR* обнаружили этот факт локальными экстремумами. Можно обнаружить и другие события, которые будут сопоставимы со структурой рассматриваемой рекуррентной диаграммы.

Применение к рядам чисел Вольфа, *aa*-индекса

Рассмотрим рекуррентные диаграммы и графики рассчитанных по ним мер для чисел Вольфа (солнечная активность) – рис. 3, 4, *aa*-индекса (геомагнитная активность) – рис. 5, 6, индекса осадков в Индии – рис. 7, 8. Диапазон исследуемых данных – с января 1871 г. по декабрь 1994 г. Все диаграммы были построены с параметрами $m = 3$, $\tau = 1$, $\varepsilon = 0,1$. Для вычисления мер использовалось окно шириной $S = 312$ точек, что соответствует 26 годам. Для упрощения визуального анализа временные ряды мер предварительно подвергались аппроксимации сплайнами с высоким коэффициентом точности.

На графиках для диаграммы чисел Вольфа нас интересует фазовый переход в районе 1934–37 гг. Действительно, графики мер *RR*, *DET*, *ENTR*, *RATIO* и *LAM* обнаруживают экстремум в конце 1935 г. (на графиках отмечен пунктирной линией). Более того, поведение этих мер до и после рубежа различно, что отчетливо видно на графиках. Можно обнаружить и другие экстремумы – например, в конце 1899 г. (на графиках линия малым пунктиром). Это событие также видно и на диаграмме, пусть не настолько отчетливо, как предыдущее.

Графики для диаграммы *aa*-индекса более интересны, так как, несмотря на достаточно высокую зашумленность и стохастичность сигнала, меры показывают высокую чувствительность к изменениям в структуре диаграммы, а значит и в поведении исследуемого процесса. Фактически виден диапазон с середины 1934 г. по середину 1938 г., в течение которого происходила перестройка поведения возмущенности геомагнитного поля Земли. Меры *DET*, *L*, *ENTR*, *LAM* и *TT* демонстрируют общий экстремум в конце 1936 г. Таким образом, если приведенная выше оценка точки фазового перехода для чисел Вольфа справедлива, можно говорить, что реакция геомагнитного поля Земли на существенную смену режима произошла с задержкой в один год.

Следует также отметить, что замеченный на графиках для диаграммы чисел Вольфа экстремум в конце 1899 г. также проявился и на графиках для диаграммы *aa*-индекса (отмечен линией с малым пунктиром). Более того, демонстрируют его уже упомянутые меры *DET*, *L*, *ENTR*, *LAM* и *TT*. Следует особо отметить, что на рекуррентной диаграмме *aa*-индекса это событие фактически неразлично.

В случае графиков для диаграммы среднего индекса осадков в Индии мы сталкиваемся с наличием достаточно большого количества экстремумов, практически не проверяемых по диаграмме. Сам временной ряд осадков представляет собой модулированный пилообразный сигнал с более-менее постоянным и очевидным периодом в 1 год. Однако как продолжительность периода осадков, так и их интенсивность меняются и подвержены более высокой периодичности, выявление которой выходит за рамки данной работы. Для нас в данном случае важно отметить отсутствие совпадений экстремумов с рядами чисел Вольфа и *aa*-индекса.

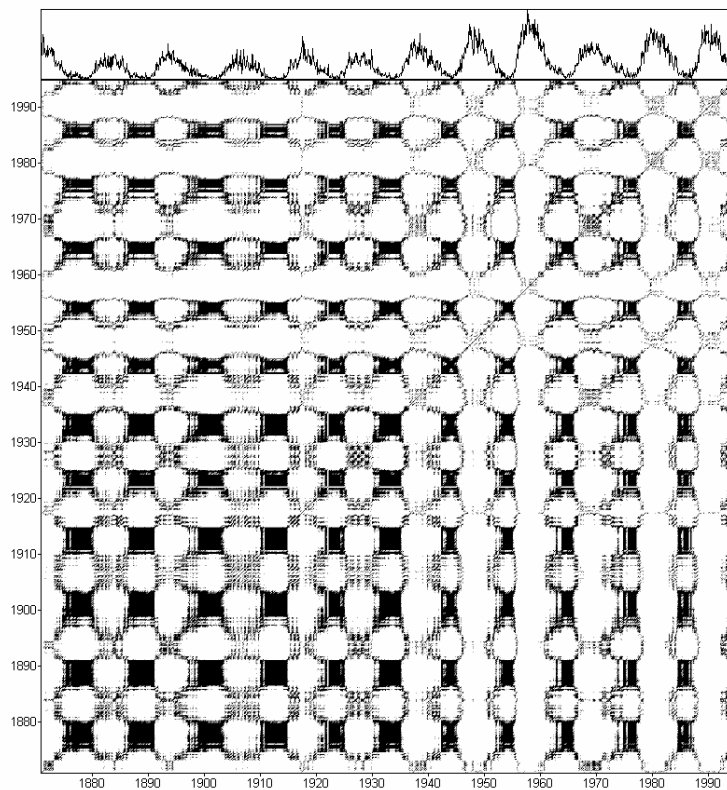


Рис. 3. Рекуррентная диаграмма месячных значений чисел Вольфа с 1871/01 по 1994/01 ($m = 3$, $\tau = 1$, $\varepsilon = 0,1$)

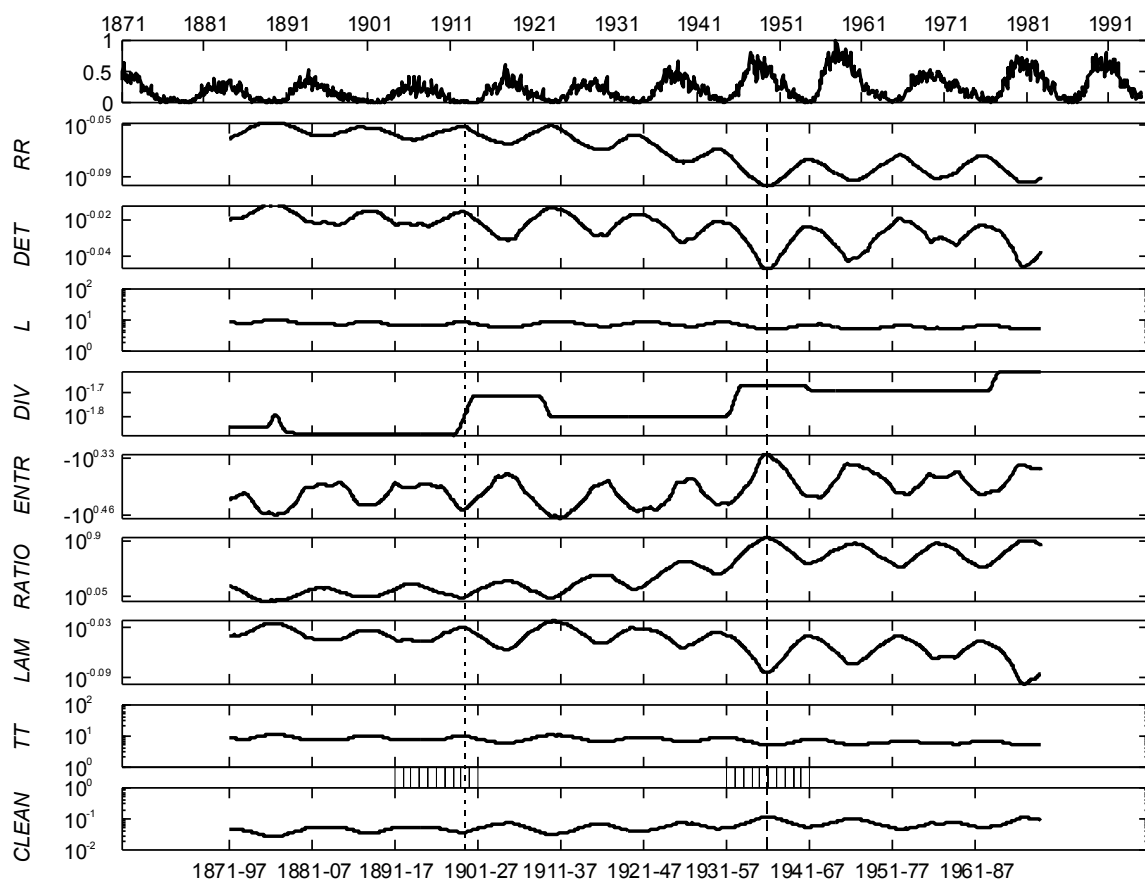


Рис. 4. Графики мер для диаграммы чисел Вольфа

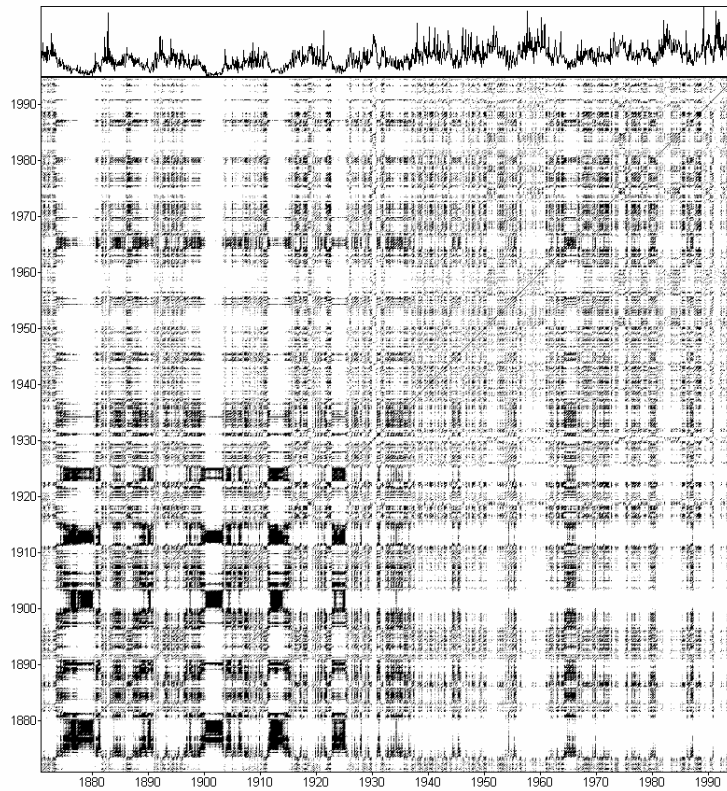


Рис. 5. Рекуррентная диаграмма месячных значений аа-индекса с 1871/01 по 1994/01 ($m = 3$, $\tau = 1$, $\varepsilon = 0,1$)

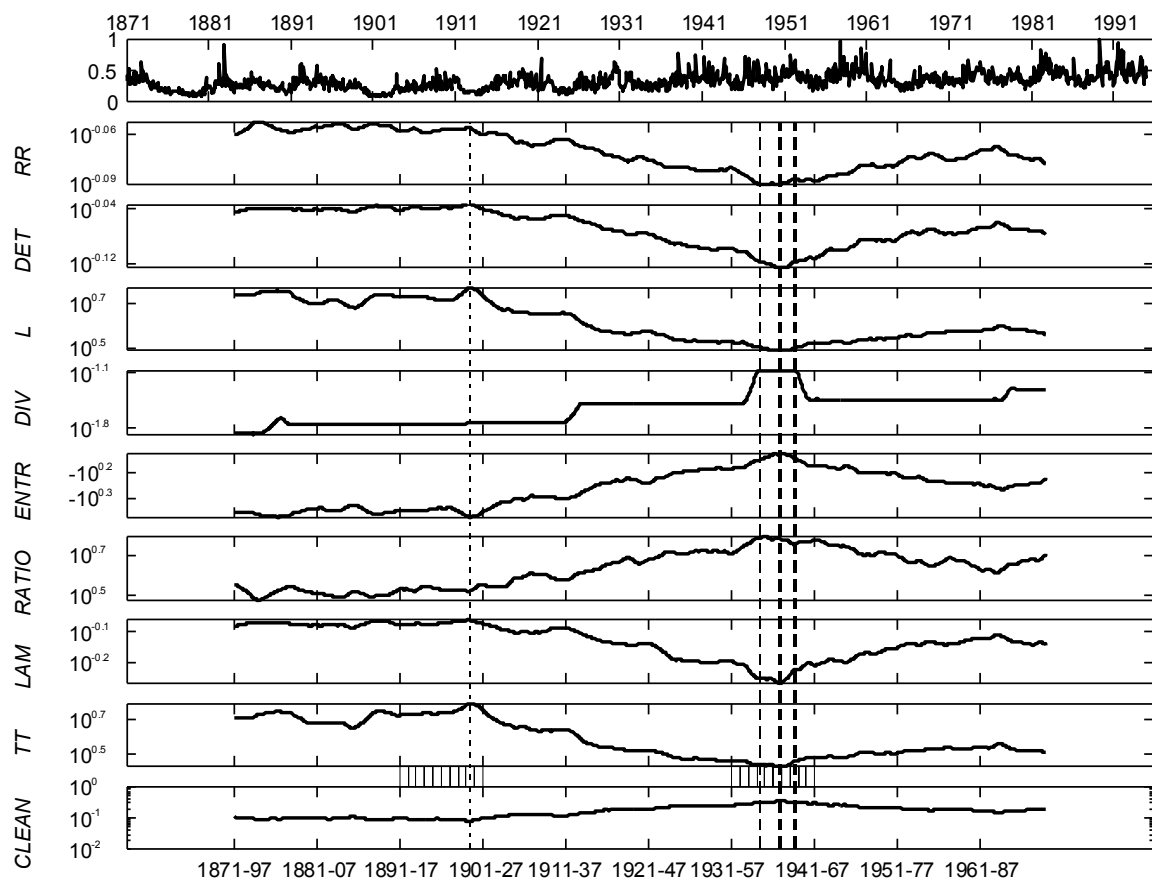


Рис. 6. Графики мер для диаграммы аа-индекса

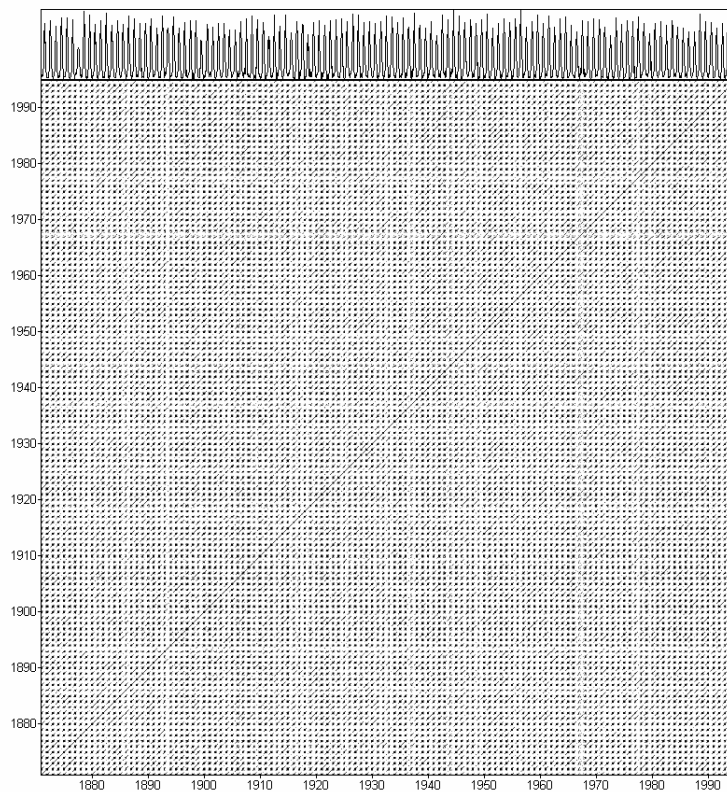


Рис. 7. Рекуррентная диаграмма среднемесячных значений уровня осадков в Индии с 1871/01 по 1994/01 ($m = 3$, $\tau = 1$, $\varepsilon = 0,1$)

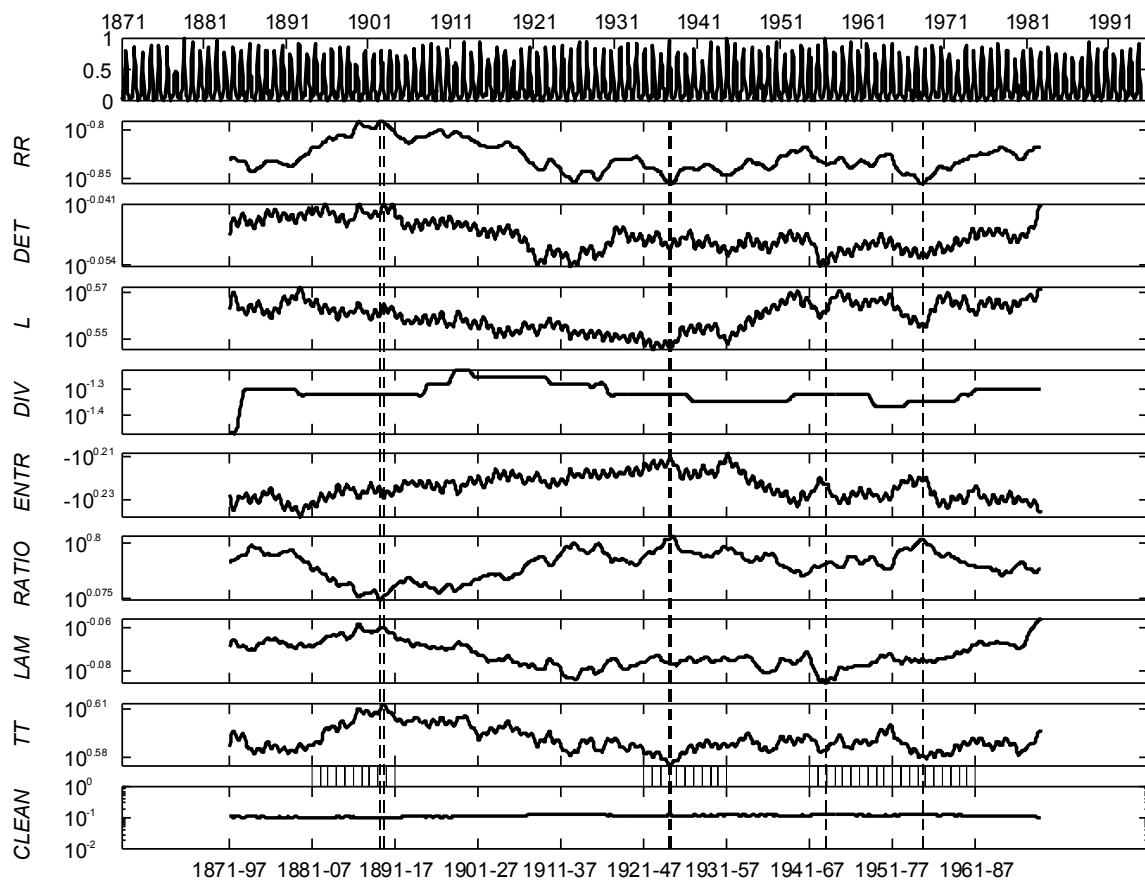


Рис. 8. Графики мер для диаграммы уровня осадков

Заключение

Рассмотренная в работе методика позволяет облегчить исследователям поиск границ фазовых переходов при изучении сложных природных процессов. Было отмечено повышение чувствительности количественных мер в случае применения к сложным сигналам с высокими шумовой и стохастической составляющими. Методика позволяет обнаруживать сложно различимые на диаграммах границы фазовых переходов.

Применение методики к рядам чисел Вольфа и аа-индекса позволило уточнить границы фазового перехода во второй половине тридцатых годов 20-го века, обнаружить границы других, менее существенных переходов.

Очевидно, что все рассмотренные меры реагируют на различные типы фазовых переходов по-разному. Рассмотрение этого вопроса осталось за рамками данной работы по причине его обширности. По этой же причине не был приведен более подробный анализ поведения мер на рассмотренных природных временных рядах. Была введена мера CLEAN как отношение между количеством рекуррентных точек диаграммы, не участвующих в формировании диагональных линий, и количеством рекуррентных точек диаграммы, участвующих в этом. Мера определена как мера чистоты траектории.

Литература

1. Eckmann J.-P., Kamphorst S. O., Ruelle D. Recurrence Plots of Dynamical Systems // *Europhysics Letters*. – 1987. – V. 5. – P. 973–977.
2. Киселев В.Б. Некоторые методы нелинейного анализа // *Научно-технический вестник СПбГУ ИТМО*. – 2005. – Вып. 20. – С. 172–180.
3. Киселев В.Б. Рекуррентный анализ – теория и практика // *Научно-технический вестник СПбГУ ИТМО*. – 2006. – В. 29. – С. 118–127.
4. Takens F. Detecting Strange Attractors in Turbulence. – Berlin, Springer, 1981. – P. 366–381.
5. Киселев В.В., Волобуев Д.М. Реконструкция аттракторов в трехмерном фазовом пространстве и моделирование геомагнитных пульсаций // *Вопросы геофизики*. – Изд. СПбГУ. – 1998. – В. 35. – С. 338–348.
6. Zbilut J. P., Webber Jr. C. L. Embeddings and delays as derived from quantification of recurrence plots // *Physics Letters A*. – 1992. – V. 171. – № 3–4. – P. 199–203.
7. Marwan N., Meinke A. Extended Recurrence Plot Analysis and its Application to ERP Data // *International Journal of Bifurcation and Chaos «Cognition and Complex Brain Dynamics»*. – 2004. – V. 14. – № 2.
8. Webber Jr.C.L., Zbilut J.P. Dynamical assessment of physiological systems and states using recurrence plot strategies // *Journal of Applied Physiology*. – 1994. – V. 76. – № 2. – P. 965–973.
9. Киселев В.В., Киселев В.Б. Различия в динамике солнечной и геомагнитной активности // *Вопросы геофизики*. – Изд. СПбГУ. – 2006. – Вып. 39. – С. 130–139.
10. Киселев В.Б. Определение стабильности траектории процесса в фазовом пространстве при помощи рекуррентного анализа // *Научно-технический вестник СПбГУ ИТМО*. – 2007. – Вып. 40. – С. 121–130.

Киселев Владислав Борисович

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, contact@impsoft.spb.ru

Крылов Борис Алексеевич

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, krylov@mail.ifmo.ru

УДК 004.414.23

**ИССЛЕДОВАНИЕ СЕТЕЙ ПЕРЕДАЧИ ДАННЫХ С УЧЕТОМ
ОГРАНИЧЕННОЙ ДЛИНЫ ПАКЕТОВ****Нгуен Дык Тай, Т.И. Алиев**

С использованием системы имитационного моделирования GPSS World проведено исследование влияния ограниченной длины пакетов на характеристики функционирования сетей передачи данных. Показано, что погрешность результатов расчета характеристик в предположении о неограниченной длине пакетов может достигать десятков процентов.

Ключевые слова: имитационное моделирование, сеть передачи данных, длина пакетов, распределение, нормирование, Ethernet.

Введение

При исследовании сетей передачи данных (СПД) в качестве математических моделей используются сети массового обслуживания (СМО), в которых узлы, представляющие собой системы массового обслуживания (СМО), отображают задержки при передаче пакетов по каналам связи (КС) [1]. Точные аналитические результаты получены для экспоненциальной модели СПД в предположении о том, что длины передаваемых пакетов распределены по экспоненциальному закону в интервале от нуля до бесконечности. В реальных СПД длина пакетов обычно ограничена снизу и сверху, например, в стеке протоколов TCP/IP длина передаваемых в СПД пакетов находится в интервале от 20 до 64К байт, а для сети Ethernet – от 64 до 1518 байт.

В работе с использованием системы имитационного моделирования GPSS исследуется влияние ограниченной длины пакетов на характеристики функционирования СПД. Выполнен сравнительный анализ полученных результатов с результатами аналитического расчета в предположении о неограниченной длине пакетов. Показано, что различие в результатах может достигать десятков процентов, что делает необходимым учет ограниченной длины пакетов в процессе системотехнического проектирования СПД.

1. Аппроксимация и нормирование распределения в GPSS World

Поскольку время передачи пакета t по каналам связи связано с его длиной l зависимостью $t = l/C$, где C – пропускная способность канала связи, то ограничение длины пакетов можно рассматривать как ограничение времени передачи пакетов. В системе имитационного моделирования GPSS используются законы распределения случайных величин, принимающих значения в интервале $[0; +\infty)$. В то же время длина пакетов и, следовательно, время их передачи по каналу связи ограничены слева и справа. Таким образом, возникает задача преобразования законов распределений, используемых в GPSS, в законы распределения, ограниченные слева и справа. Эта задача может быть решена двумя способами: усечением распределения слева и справа (рис. 1, а); смещением распределения слева и усечением справа (рис. 1, б). Полученные распределения нормируются так, чтобы сохранились средние значения генерируемых случайных величин.

Рассмотрим указанные способы преобразования законов распределений времени передачи пакетов применительно к экспоненциальному распределению. Для нормирования усеченного экспоненциального распределения может использоваться либо имитационный метод, либо комбинация аналитического и имитационного метода.

1.1. Имитационный метод

На рис. 1, а, показана плотность экспоненциального распределения $f(t) = \lambda e^{-\lambda t}$, где $\lambda > 0$ – параметр, представляющий собой величину, обратную средней длительности передачи пакета. В GPSS будем использовать метод розыгрыша случайных величин в интервале $[t_{\min}, t_{\max}]$, обеспечивая $\int_{t_{\min}}^{t_{\max}} f(t) dt = 1$. На рис. 2, а, представлен GPSS-фрагмент, генерирующий распределение для случая $t_{\min} = 0$ и $t_{\max} = 4$. Параметр *time* сохраняет значения случайной величины (рис. 2, б).

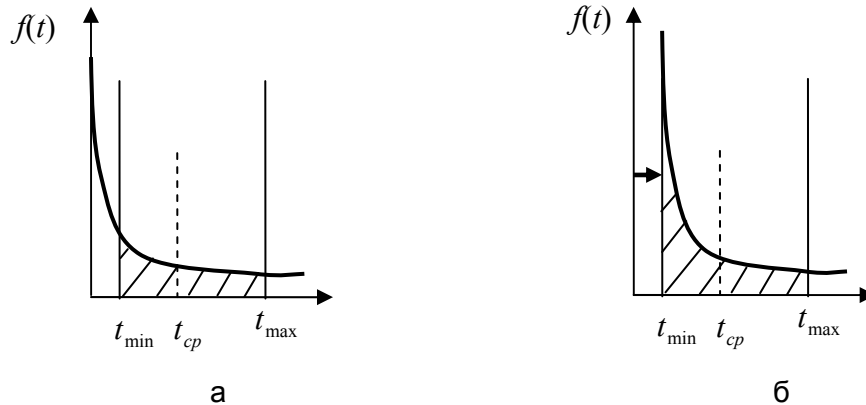


Рис. 1. Преобразование законов распределения с усечением (а) и со смещением (б)

```
s FVARIABLE (Exponential(4,0,2)) t_max = 4.
time FVARIABLE X$SUM_
WAITTIME TABLE v$time,0,0.5,12
INITIAL X$SUM_,0
GENERATE v$serve_
SAVEVALUE SUM_,v$s
TEST GE X$SUM_,0,RLS
TEST L X$SUM_,4,RLS
TABULATE WAITTIME
RLS TERMINATE 1
START 1000000
```

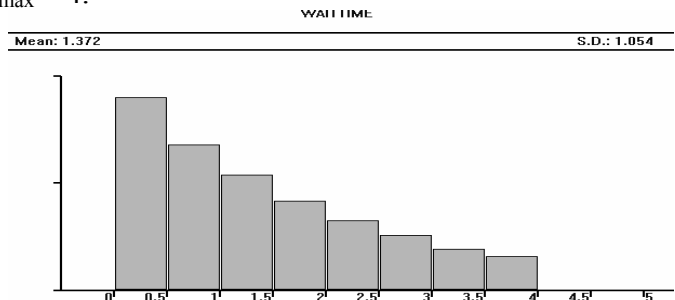


Рис. 2. Имитационный метод нормирования усеченного экспоненциального распределения: а – GPSS-фрагмент; б – гистограмма распределения

$\frac{t_{\max}}{t_{cp}}$	МО	ν	A1
2	1,372	0,768	1,4577
3	1,683	0,844	1,1884
4	1,842	0,901	1,0858
5	1,924	0,943	1,0395
6	1,961	0,969	1,0199
7	1,984	0,989	1,0081
8	1,991	0,997	1,0045
9	1,994	1	1,0030

Таблица 1. Характеристики экспоненциального распределения «с усечением»

Характеристики распределения «с усечением» приведены в табл. 1 для случая $t_{\min} = 0$, где МО – математическое ожидание, ν – коэффициент вариации, а A1 – отно-

шение t_{cp} / MO . Очевидно, когда t_{max} / t_{cp} уменьшается, математическое ожидание и коэффициент вариации тоже уменьшаются. Когда t_{max} / t_{cp} увеличивается до 9, распределение «с усечением» практически становится экспоненциальным.

Для нормирования экспоненциального распределения с сохранением математического ожидания параметр $time$ необходимо пересчитать путем умножения на коэффициент $A1$. Программа на GPSS для исследования системы $M/M^*/1$ (M^* – обозначение экспоненциального распределения «с усечением») имеет следующий вид:

input_time	FVARIABLE (Exponential(5,0,4))	DEPART	QTurn
s	FVARIABLE (Exponential(4,0,2))	REP	SAVEVALUE SUM_,v\$s
time	FVARIABLE 1.4577#X\$SUM_	TEST G	X\$SUM_,0,REP
WAITTIME	TABLE v\$time,0,0.5,12	TEST L	X\$SUM_,4,REP
INITIAL	X\$SUM_,0	ADVANCE	v\$time
	GENERATE v\$input_time	TABULATE	WAITTIME
	QUEUE QTurn	RELEASE	Turn
	SEIZE Turn	TERMINATE	1

Результаты исследования характеристик системы $M/M^*/1$ для случая $t_{min} = 0$ показаны на рис. 3, а, а для случая $t_{min} > 0$ – на рис. 3, б.

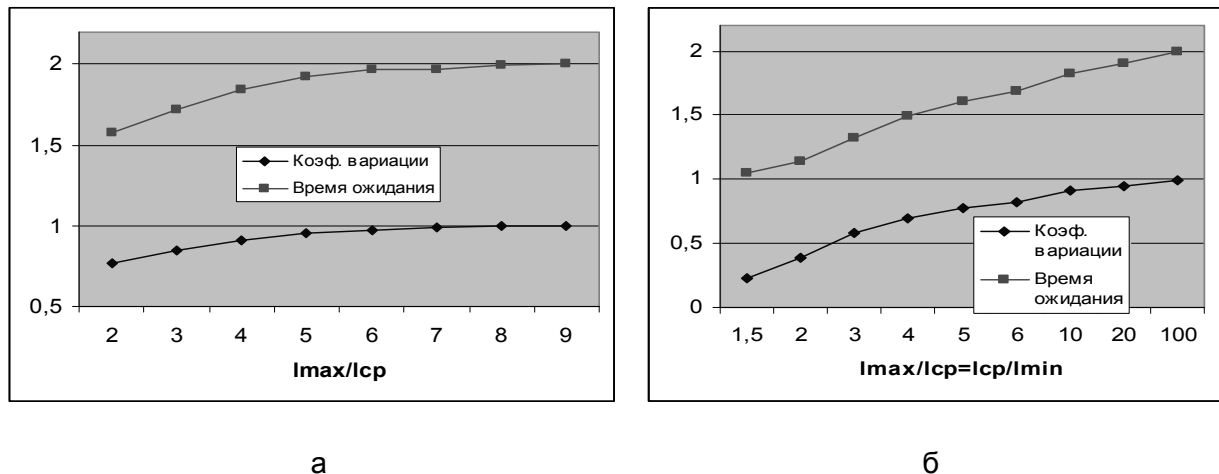


Рис. 3. Характеристики системы $M/M^*/1$ при $t_{min} = 0$ (а) и $t_{min} > 0$ (б)

Анализ представленных результатов показывает, что отличие среднего времени ожидания пакетов в узле перед каналом связи в случае ограниченной длины передаваемых пакетов может составлять десятки процентов от значений, полученных в предположении о неограниченной длине пакетов. Причем это отличие более значительно, если $t_{min} > 0$.

1.2. Комбинация аналитического и имитационного метода

Пусть время передачи пакета в канале распределено по экспоненциальному закону: $F(t) = 1 - \lambda e^{-\lambda t}$, $f(t) = \lambda e^{-\lambda t}$. Как видно из рис. 4, а, начало и хвост плотности распределения $f(t)$ можно «отрезать», поскольку время передачи пакета в канале всегда больше нуля и не превышает t_{max} (длина пакетов ограничена слева и справа).

Для нормирования плотности распределения $f(t)$ используется функция $f_2(t)$ в виде $f_2(t) = A f(t)$, где A – нормировочный коэффициент, определяемый из условия

$$A \int_{t_{min}}^{t_{max}} f(t) dt = 1, \quad (1)$$

$$\lambda = \frac{1}{t_{cp}} = \frac{1}{l_{cp}/C} = \frac{C}{l_{cp}},$$

где l_{cp} – средняя длина пакета.

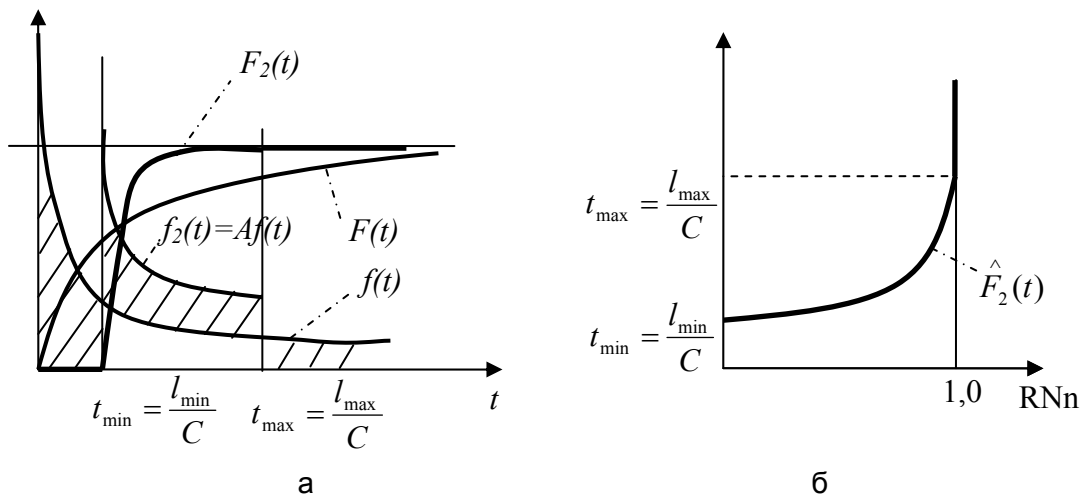


Рис. 4. Комбинационный метод нормирования усеченного экспоненциального распределения: а – функция и плотность распределения; б – графическая интерпретация функции FN\$XPDIS2

Из уравнения (1) получим:

$$A = 1 / \int_{t_{min}}^{t_{max}} f(t) dt = 1 / \int_{t_{min}}^{t_{max}} \lambda e^{-\lambda t} dt = 1 / (e^{-\frac{t_{min}}{t_{cp}}} - e^{-\frac{t_{max}}{t_{cp}}}) = 1 / (e^{-\frac{l_{min}}{l_{cp}}} - e^{-\frac{l_{max}}{l_{cp}}}).$$

Очевидно, что $A > 1$.

Функция распределения $F_2(t)$ имеет вид $A(B - e^{-\lambda t})$ и удовлетворяет условию $F_2(t_{max}) = 1$. В результате $F_2(t) = A(e^{-\lambda t_{min}} - e^{-\lambda t})$, $t_{min} \leq t \leq t_{max}$.

Для исследования влияния полученного распределения («с усечением») времени передачи пакетов на характеристики канала воспользуемся имитационным моделированием в среде GPSS World. В GPSS World экспоненциальное распределение формируется по методу обратной функции [2]. В GPSS табличное представление экспоненциального распределения для математического ожидания, равного 1, содержит 24 пары значений, используемых для преобразования значений $[-\ln(1-RNn)]$ (где RNn – имя генератора равномерно распределенных случайных величин в интервале от 0 до 1):

```
XPDIS      FUNCTION      RN1, C24
0, 0/.100, .104/.200, .222/.300, .355/.400, .509/.500, .690/.600, .915/.700, 1.200/.
750, 1.380/.800, 1.600/.840, 1.830/.880, 2.120/.900, 2.300/.920, 2.520/.940, 2.810/.
.950, 2.990/.960, 3.200/.970, 3.500/.980, 3.900/.990, 4.600/.995, 5.300/.998, 6.200
/.999, 7/1, 8
```

Для моделирования функции $F_2(t)$ необходимо сформировать новый ряд пар чисел, поскольку обратная функция $\hat{F}_2(t)$ имеет вид (рис. 4, б)

$$t = \hat{F}_2(t) = -t_{cp} \ln(e^{-\frac{t_{min}}{t_{cp}}} - \frac{F_2(t)}{A}).$$

Для случая $t_{cp} = 2$, $A = 1,1565$ (соответственно $t_{cp} / t_{min} = 2$, $t_{max} / t_{cp} = 2$) имеем следующее множество пар чисел:

```
XPDIS2      FUNCTION      RN1, C24
0, 1/0.1, 1.1617/0.2, 1.3377/0.3, 1.5307/0.4, 1.7443/0.5, 1.9835/0.6, 2.2552/0.7, 2
.5697/0.75, 2.7477/0.8, 2.9431/0.84, 3.1144/0.88, 3.3018/0.9, 3.4025/0.92, 3.5086
```

/0.94, 3.6206/0.95, 3.679/0.96, 3.7392/0.97, 3.8013/0.98, 3.8654/0.99, 3.9316/0.995, 3.9655/0.998, 3.9861/0.999, 3.993/0.9999, 3.9993

В GPSS World блок ADVANCE имеет вид ADVANCE A1, FN\$XPDIS2. При этом сохраняется математическое ожидание распределения. Отметим, что этот метод реализуется аналитически только для экспоненциального распределения.

Результаты характеристик системы М/М*/1 с использованием этого метода почти совпадают с результатами первого метода. Результаты обоих методов оказываются почти одинаковыми, что подтверждает корректность аналитического способа построения обратной функции распределения.

2. Результаты исследования для гиперэкспоненциального распределения

Для исследования гиперэкспоненциального распределения длины пакетов воспользуемся первым методом. Аппроксимацию гиперэкспоненциального распределения выполним по двум моментам. Случайная величина формируется из двух экспоненциальных распределений, причем с вероятностью q значение случайной величины выбирается из первого распределения с математическим ожиданием t_1 и с вероятностью $(1 - q)$ – из второго распределения с математическим ожиданием t_2 .

Результаты исследования системы М/Н₂/1 для гиперэкспоненциального распределения с коэффициентом вариации 2 ($q = 0,399$, $t_1 = 5,00626$, $t_2 = 0,00416$) для случая $l_{\min} = 0$ представлены на рис. 5. График показывает зависимость времени ожидания пакетов и коэффициента вариации времени передачи пакетов от отношения $l_{\max}/l_{cp}$ ($l_{\min} = 0$).

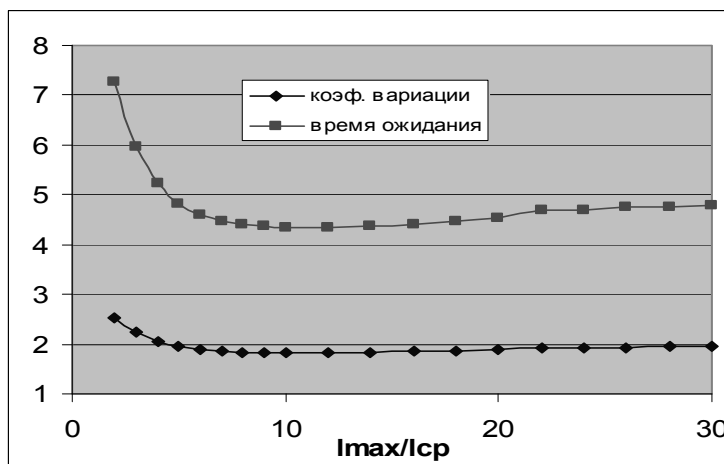


Рис. 5. Характеристики системы М/Н₂/1

Ограничение распределения по способу «со смещением» выполняется проще, чем с усечением. Например, для экспоненциального распределения можно использовать встроенную функцию Exponential (1, смещение, математическое ожидание). Поскольку гиперэкспоненциальное распределение строится на основе экспоненциального, то для этого распределения целесообразно использовать способ «со смещением». На рис. 6 показана зависимость времени ожидания пакетов от отношения $l_{\max}/l_{cp}$ ($= l_{cp}/l_{\min}$) для экспоненциального и гиперэкспоненциального распределения с коэффициентом вариации 2 для двух способов «с усечением» и «со смещением». Из графика видно, что для экспоненциального распределения оба способа дают почти одинаковые значения времени ожидания. Однако для гиперэкспоненциального распределения способ «со смещением» дает значительно большие значения времени ожидания, чем для случая «с усечением».

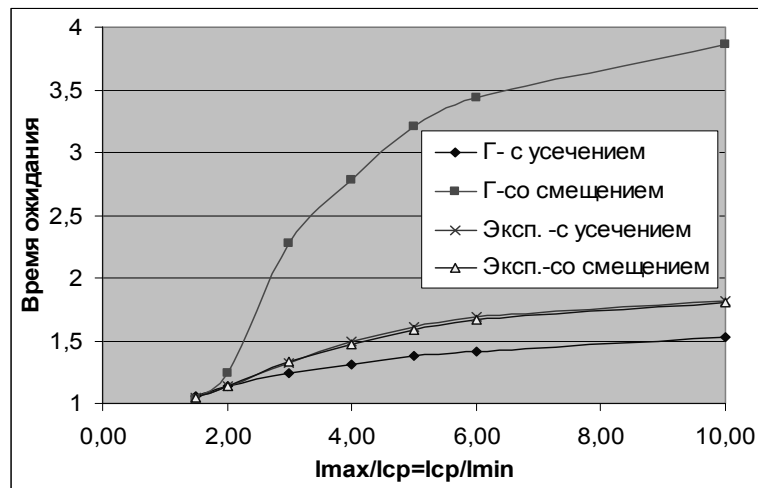


Рис. 6. Зависимость времени ожидания пакетов от отношения $l_{\max}/l_{cp}$

3. Исследование влияния длины кадров на характеристики локальной сети Ethernet

Для локальной сети Ethernet при средней длине пакета (кадра) 512 байт минимальная длина равна 64 байт, максимальная длина – 1518 байт. Отсюда получаем $l_{\max}/l_{cp} \approx 3$, $l_{cp}/l_{\min} = 8$. Рассмотрим ограничение длины пакетов для этой сети при использовании подхода «со смещением» для трех распределений: экспоненциального, равномерного и гиперэкспоненциального (рис. 7). Зависимость времени ожидания пакетов от загрузки канала показана на рис. 8. Как видно из рис. 8, для обоих распределений – экспоненциального и гиперэкспоненциального – их традиционные распределения (без ограничения слева и справа) всегда дают худшие результаты, чем распределения с ограничением (в этом случае – со смещением). При этом чем больше загрузка канала, тем больше отличие времени ожидания. Для гиперэкспоненциального распределения, когда загрузка составляет 0,9, время ожидания для традиционного случая в 2 раза больше, чем в случае распределения с ограничением слева и справа.

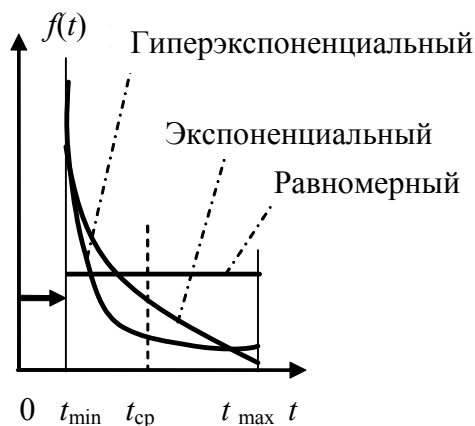


Рис. 7. Преобразование законов распределений «со смещением»

Для стеков протоколов TCP/IP сети при средней длине IP-пакета 1500 байт минимальная длина IP-пакета равна 20 байт, максимальная длина – 64 Кбайт. Тогда $l_{\max}/l_{cp}=75$, $l_{cp}/l_{\min} \approx 43$. Большое отношение между максимальной (минимальной) и средней длиной пакетов позволяет использовать традиционные распределения без ограничения слева и справа, поскольку погрешность получаемых при этом результатов составляет менее 1%.

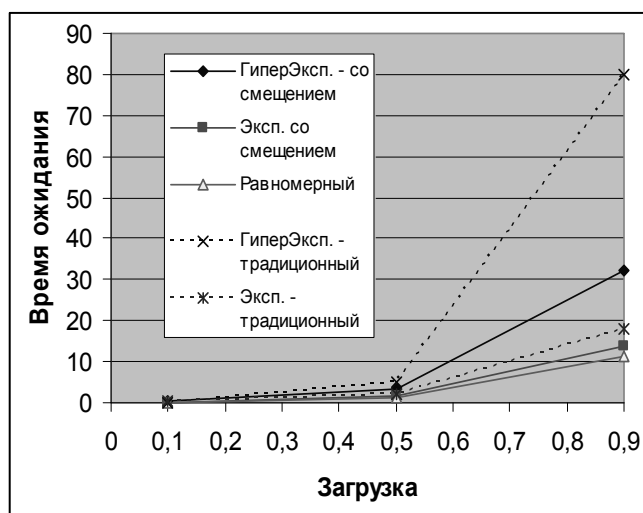


Рис. 8. Зависимость времени ожидания пакетов от загрузки канала

4. Исследование влияния экспоненциального времени передачи пакетов с ограниченной длиной пакетов на характеристики СПД

Воспользуемся программным комплексом проектирования СПД [3] для исследования характеристик СПД (рис. 9) со следующими параметрами: длина одного пакета равна 1500 байт; длина обрамления – 46 байт; длина текстового сообщения – 15000 байт; вероятность передачи пакета по основному пути – 0,9; тип канала – дуплексный; внешние интенсивности текстовых сообщений одинаковы и равны 3 с^{-1} . Результаты характеристик СПД для случая $l_{\min} = 0$ и $l_{\max} / l_{\text{cp}} = 2$ представлены в табл. 2, а для случая $l_{\min} > 0$ – в табл. 3. Здесь T – среднее время доставки пакетов в СПД, $\%T$ – относительное отклонение среднего времени доставки пакетов в СеМО с узлами типа М/М*/1 по сравнению экспоненциальной СеМО.

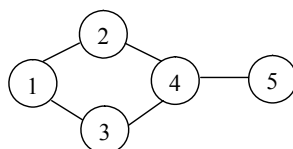


Рис. 9. Пример сети передачи данных

$l_{\max} / l_{\text{cp}}$	$l_{\max}$	T	$\%T$
2	3000	0,0369	63,7
3	4500	0,0545	45,5
4	6000	0,0719	28,06
6	9000	0,0857	14,23
7	10500	0,0924	7,51
8	12000	0,0963	3,69

Таблица 2. Характеристики СПД для случая $l_{\min} = 0$ и $l_{\max} / l_{\text{cp}} = 2$

Анализ представленных результатов показывает, что относительное отклонение времени доставки пакетов может достигать 65% для случая $l_{\min} = 0$, $l_{\max} = 3000$ байт, $l_{\text{cp}} = 1500$ байт. Это отклонение составляет меньше 20% только тогда, когда $l_{\max} \geq 9000$ байт. Для случаев $l_{\min} > 0$ максимальное отклонение составляет около 40%, когда $l_{\min} = 750$ байт и $l_{\max} = 3000$ байт.

l_{cp} / l_{min}	l_{max} / l_{cp}	l_{min}	l_{max}	T	$\%T$
6	1,5	250	2250	0,0675	32,5
5	2	300	3000	0,0698	30,2
1,5	6	1000	9000	0,0705	29,4
1,5	20	1000	30000	0,0715	28,4
2	2	750	3000	0,0609	39,1
3	3	500	4500	0,0691	30,9
4	4	375	6000	0,0762	23,7
5	5	300	7500	0,0812	18,8
6	6	250	9000	0,0861	13,9
10	10	150	15000	0,0918	8,1
20	20	75	30000	0,0940	5,9
100	100	15	150000	0,0989	1,1

Таблица 3. Характеристики СПД для случая $l_{min} > 0$

Заключение

В работе рассмотрены разные подходы по оценке влияния ограниченной длины пакетов с использованием нормирования разных распределений на характеристики каналов в отдельности и, следовательно, сети в целом. Показано, что для нормирования усеченного экспоненциального распределения оба метода (имитационный и комбинация аналитического и имитационного методов) дают одинаковые результаты. Для экспоненциального распределения оба способа ограничения пакетов слева и справа (усечением и смещением) дают практически одинаковые временные характеристики передачи данных. Для сети Ethernet относительное отклонение результатов распределения со смещением по сравнению с традиционными составляет десятки процентов, и это отклонение увеличивается с увеличением загрузки канала и может превышать 100% для гиперэкспоненциального распределения. Традиционные распределения могут использоваться при моделировании сети TCP/IP из-за большого отношения между максимальной (минимальной) и средней длиной пакетов. Различие в результатах для сети передачи данных в общем случае может составлять десятки процентов, что делает необходимым учет ограниченной длины пакетов в процессе системотехнического проектирования СПД.

Литература

1. Нгуен Дык Тай. Оценка пропускных способностей каналов связи распределенных сетей передачи данных // Научно-технический вестник СПбГУ ИТМО. – 2007. – Вып. 39. – С. 258–267.
2. Боев В.Д. Моделирование систем. Инструментальные средства GPSS World: Учебное пособие. – СПб: БХВ-Петербург, 2004. – 368 с.
3. Алиев Т.И., Нгуен Дык Тай. Программный комплекс аналитического и имитационного моделирования сетей передачи данных // Всерос. н.-пр. конф. «Имитационное моделирование. Теория и практика» (ИММОД-2007). – СПб, 2007. – С. 11–16.

Нгуен Дык Тай

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, duc_tai@mail.ru

Алиев Тауфик Измаилович

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, aliev@d1.ifmo.ru

ОЦЕНКА ЕМКОСТИ БУФЕРНОЙ ПАМЯТИ В ПРОМЕЖУТОЧНЫХ УЗЛАХ ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ

В.В. Соснин, Т.И. Алиев

Показана невозможность применения аналитических методов для оценки емкости буферной памяти в промежуточных узлах компьютерных сетей. Предлагается целый ряд полученных на имитационных моделях результатов, дающих на порядки более точные оценки. Эмпирически получено неравенство, применимое для оценки емкости буферной памяти в области высоких и средних загрузок.

Ключевые слова: телекоммуникационный узел, буферная память, верхняя оценка, имитационное моделирование.

Введение

При передаче протокольных блоков данных (ПБД) по телекоммуникационным сетям существует необходимость хранить эти блоки в промежуточных узлах сети в течение времени, необходимого для принятия определенного решения (например, для осуществления маршрутизации, если промежуточным узлом является маршрутизатор). Промежуточным узлом телекоммуникационной сети (ПУТС) может также быть коммутатор или компьютер, выполняющий роль моста/маршрутизатора. В моменты высоких загрузок в ПУТС будет скапливаться большое число ждущих обработки ПБД. Если их суммарный размер превысит емкость буферной памяти (БП), установленной в ПУТС, то некоторые из ПБД будут потеряны. Для снижения процента потерь можно либо увеличить скорость обработки ПБД в ПУТС, либо поставить больше БП, чтобы в моменты пиковых загрузок приходящие ПБД не терялись, а ожидали своей очереди. Эти два параметра (скорость обработки и емкость БП) в совокупности определяют мощность ПУТС.

При проектировании ПУТС существует задача определения оптимальной мощности ПУТС путем выбора соотношения емкости БП и скорости обработки. При этом увеличение объема БП является более дешевой альтернативой. Для скорости обработки есть верхнее ограничение, определяемое суммарной скоростью передачи данных по всем каналам связи, объединенным в рассматриваемом ПУТС. Однако указать верхнюю оценку для емкости БП можно лишь вероятностно, так как в моменты пульсаций скорости трафика в БП будет оказываться непредсказуемое число ПБД. В предлагаемой работе исследовалась возможность оценки емкости БП, которой было бы достаточно для обеспечения некоторого заданного процента потерь. Существующие аналитические методы, которые позволяют это сделать, дают слишком грубые оценки, поэтому исследование проводилось на имитационных моделях.

Постановка задачи

Попробуем аналитическими методами оценить такую емкость БП перед обслуживающим прибором, при которой процент потерь заявок вследствие переполнения будет меньше, чем $P^* = 10^{-8}$ (выбор именно этого значения обусловлен тем, что при соответствующих имитационных исследованиях получение оценок с P^* более низких степеней слишком вычислительно трудоемко, как указано ниже по тексту). На практике допустимый процент потерь ПБД на порядки больше указанного, но будем использовать именно это значение P^* , чтобы получить верхние оценки, которые останутся верными (хотя и завышенными) и в более благоприятных условиях.

При моделировании проектируемых телекоммуникационных сетей зарекомендовал себя способ, при котором каждый из ПУТС рассматривается как система массового обслуживания (СМО). Под заявкой в такой СМО будем понимать ПБД, под обслуживающим прибором – управляющий блок ПУТС, осуществляющий все необходимые

действия перед отправкой ПБД к следующему ПУТС. Очевидно, что время между поступлением заявок в такую СМО является случайной величиной. Время обслуживания в приборе также является случайной величиной, так как оно может определяться либо размером ПБД (если в процедуру обработки включать и время отправки ПБД по каналу связи), либо различием для разных ПБД уровня качества обслуживания, определенного политикой QoS в данном ПУТС.

В качестве примера рассмотрим простейший случай СМО М/М/1 (символика Кендалла [1, с. 107]): закон распределения времени между поступающими ПДБ является экспоненциальным, и время обработки ПБД в приборе также распределено по экспоненциальному закону. Пусть обслуживающий прибор загружен на 90%. Для оценки требуемой емкости БП попробуем рассчитать такое количество заявок в СМО, вероятность превышения которого меньше либо равна P^* . Отметим, что рационально давать оценку именно числу заявок в СМО, а не числу заявок, ожидающих в очереди, так как на практике обрабатываемый в данный момент в ПУТС ПБД не извлекается из БП. Применяя формулы [2, с. 208], найдем математическое ожидание числа заявок Q , находящихся в системе: $M[Q] = \frac{\rho}{1-\rho} = 9$, где ρ – загрузка. Закон распределения числа зая-

вок в СМО М/М/1 является геометрическим. Геометрический закон распределения случайной величины X задается формулой $\Pr(X = k) = \gamma(1-\gamma)^k$, где γ – параметр закона, принадлежащий интервалу $(0;1)$, $\Pr(X = k)$ – вероятность того, что величина X примет значение k . Зная $M[Q]$, рассчитаем параметр $\gamma = \frac{1}{1+M[Q]} = 0,1$. Теперь из неравенства

$\sum_{k=a}^{\infty} \gamma(1-\gamma)^k \leq P^*$ найдем a , которое и будет являться оценкой минимального числа заявок, вероятность превышения которого меньше P^* . Сумма представляет собой геометрическую прогрессию, поэтому $\frac{\gamma(1-\gamma)^a}{1-(1-\gamma)} \leq P^* \Rightarrow (1-\gamma)^a \leq P^*$, следовательно

$a \geq \log_{1-\gamma} P^* \Rightarrow a \geq \frac{\ln P^*}{\ln(1-\gamma)} = \frac{\ln 10^{-8}}{\ln 0,9} \approx 175$. Получилось, что емкость БП должна примерно в 20 превышать среднюю наполненность БП, т.е. $a \approx 20 \cdot M[Q]$.

Как видим, при наличии исчерпывающей информации о параметрах СМО можно дать точную оценку емкости БП. Но попробуем теперь осуществить расчеты, рассматривая СМО М/М/1 как СМО М/Г/1 с коэффициентом вариации времени обработки, равным 1. Аналитическими методами для этой СМО М/Г/1 можно получить, что среднеквадратическое отклонение величины Q равно $\sigma[Q] = M[Q] = 9$. Не зная закона распределения Q , можно в соответствии с неравенством Чебышева оценить значение a , которое случайная величина Q не превысит с вероятностью P^* :

$$P(|Q - M[Q]| \geq a) \leq \frac{\sigma^2[Q]}{a^2} = P^* = 10^{-8}, \text{ откуда } a = \frac{\sigma[Q]}{\sqrt{P^*}} = 10^4 \cdot \sigma[Q] = 10^4 \cdot M[Q].$$

Таким образом, запас емкости БП в СМО М/Г/1 должен на четыре порядка превышать среднюю наполненность БП, что на три порядка превышает запас емкости, рассчитанный для СМО М/М/1. Конечно, это лишь верхняя оценка, но руководствоваться на практике такой оценкой невозможно, так как стоимостные затраты при этом будут слишком велики. Заметим, что при рассмотрении СМО Г/Г/1 для получения оценок нельзя применить даже и неравенство Чебышева, так как в этом случае нельзя аналитическими методами оценить коэффициент вариации величины Q .

На практике далеко не всегда можно с достаточной точностью аппроксимировать закон распределения величины Q . Поэтому в общем случае для получения достоверных

оценок емкости БП применение аналитических методов невозможно. В работе ставилась задача на основе большого числа имитационных экспериментов предложить эмпирическую оценку числа a всего по двум первым моментам законов распределения времени обработки и времени между поступлением заявок (такой подход обоснован, так как определяющее влияние на характеристики системы оказывают именно два первых момента законов распределения ее параметров).

Методы исследований

В качестве среды имитационного моделирования в работе использовалась GPSS World. Исследовался класс СМО G/G/1. Значение P^* было выбрано равным 10^{-8} . Задачей было выявить зависимость максимальной длины очереди от первых двух моментов законов распределения параметров этой СМО, а для этого при исследовании производилось варьирование:

- 1) загрузки (менялось соотношение математического ожидания интенсивности прихода заявок и скорости их обработки);
- 2) коэффициента вариации времени между поступлением заявок ($KB_{ТП}$), см. таблицу;
- 3) коэффициента вариации времени обработки заявок ($KB_{ТО}$), см. таблицу.

Номер эксперимента	$KB_{ТП}$	$KB_{ТО}$
1	0,5	0,5
2	0,5	1,0
3	0,5	1,5
4	1,0	0,5
5	1,0	1,0
6	1,0	1,5
7	1,5	0,5
8	1,5	1,0
9	1,5	1,5

Таблица. Описание проведенных экспериментов

Каждый эксперимент проводился независимо для семи разных наборов датчиков случайных чисел (ДСЧ), используемых при моделировании законов распределений. Варьирование $KB_{ТП}$ и $KB_{ТО}$ проводилось с применением разных законов распределений. Проведенное сравнение показало, что при равенстве первых двух моментов получаемые результаты отличаются не более чем на 5–10% вне зависимости от выбора закона распределения (т.е. влияние на результаты высоких моментов распределения очень невелико). Поэтому было решено основную часть экспериментов проводить, используя гамма-распределение как самое удобное в применении в среде GPSS World.

Заметим, что для получения достоверных оценок максимального числа заявок в очереди оказалось необходимым пропускать через созданную модель более чем 10^8 транзактов, что требует на компьютере с процессором тактовой частоты 1666 Mhz и объемом ОЗУ 1024 МБ более 30 минут. При этом результатом одного такого прогона модели является получение одной точки на приводимых ниже графиках (см. рис. 2). С учетом того, что использовались семь наборов ДСЧ и того, что каждый график построен по не менее чем 15-ти точкам, общее чистое время моделирования на указанном компьютере составляет более 470 часов.

Результаты работы

На рис. 2 приведены результаты моделирования СМО G/G/1 при $KB_{ТП} = 1$ и $KB_{ТО} = 1,5$. Использовано обозначение $Max(Q)$ – максимальное полученное в экспери-

менте значение числа заявок в системе. Значение $M[Q]$ в данном случае получено точно с использованием формулы Поллячека–Хинчина для системы M/G/1. На рисунке также показано отношение максимального числа заявок к среднему. Именно это отношение представляет интерес. Было замечено, что отношение максимального числа заявок к среднему имеет асимптотический характер: по мере увеличения загрузки это отношение убывает и не превышает в пределе значения 20. Этот вывод справедлив для любых законов распределения.

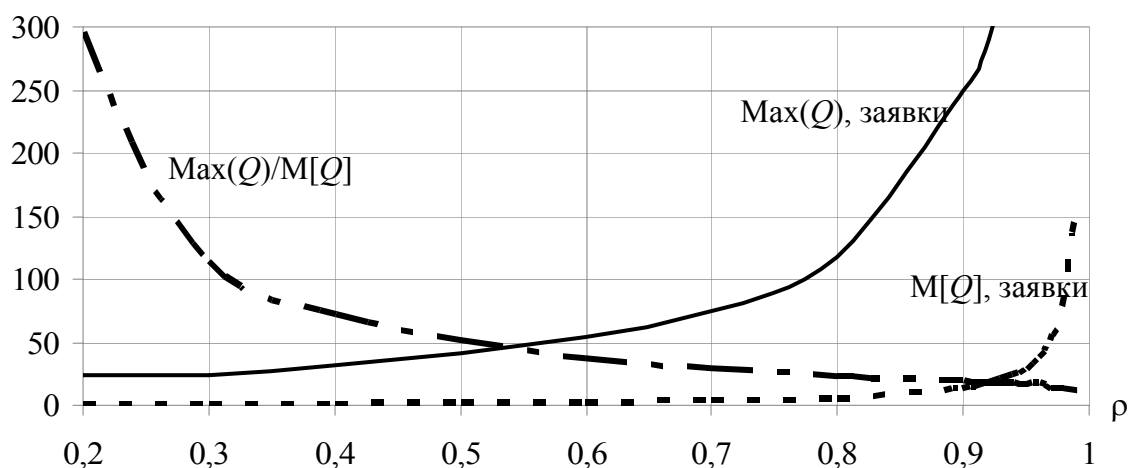


Рис. 2. Зависимость размера очереди в СМО от загрузки

Иллюстрирует сделанное утверждение рис. 3. На нем показана зависимость отношения максимального размера очереди к ее среднему размеру от загрузки и от выбранных значений KB_{TP} и KB_{TO} . При этом по мере движения от точки А к точке Б в моделируемых СМО, соответствующих пересекаемым графикам, изменялись законы распределения времени между приходом заявок и времени обработки заявок так, что сумма коэффициентов вариации обоих законов уменьшалась.

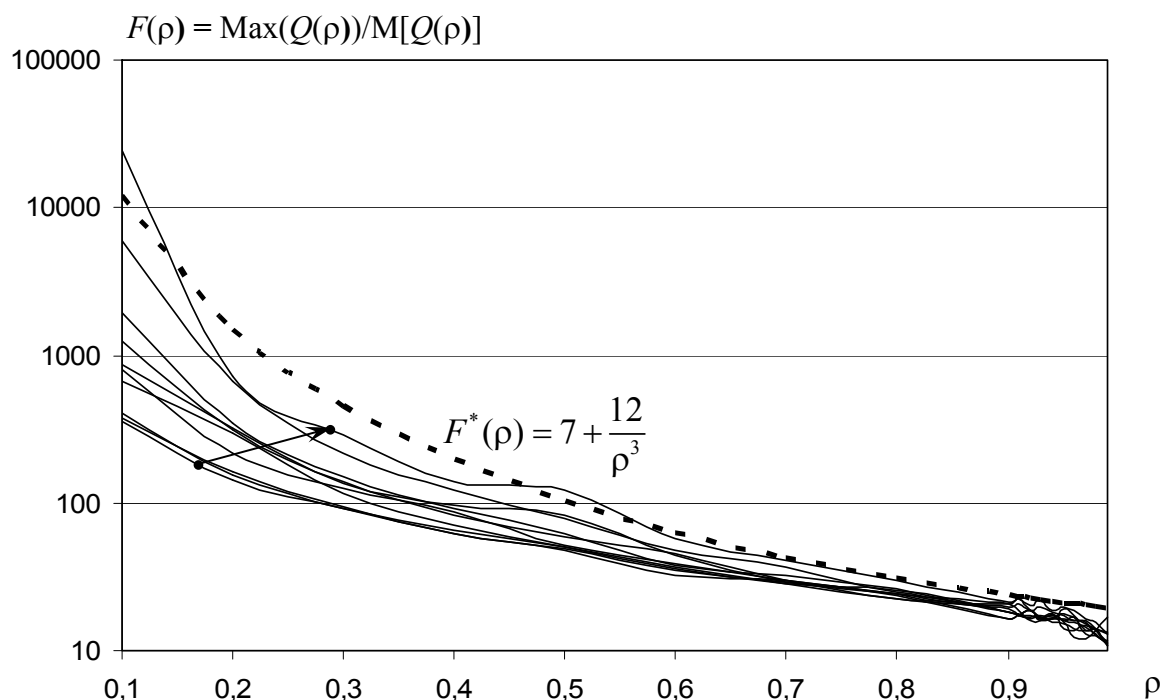


Рис. 3. Эмпирическая оценка максимального размера очереди

Например, график, которому принадлежит точка А, получен при коэффициентах вариации, равных 1,5, а график, которому принадлежит точка Б, получен при коэффициентах вариации, равных 0,5. Как видно, по мере увеличения коэффициентов вариации каждый из последующих графиков функции $F(\rho)$ лежит ниже предыдущего. Следовательно, можно дать верхнюю оценку отношению $\text{Max}(Q)/M[Q]$. На рис. 3 пунктирной линией отмечен график функции $F^*(\rho) = 7 + \frac{12}{\rho^3}$, который дает такую оценку.

График функции $F^*(\rho)$ во всех точках находится выше полученных экспериментально данных и дает оценку приемлемой точности при соблюдении следующих условий:

- 1) $KB_{\text{ТП}}$ и $KB_{\text{ТО}}$ в сумме превышают 1,5;
- 2) нагрузка превышает 0,5.

При этих условиях функция $F^*(\rho)$ позволяет оценить $\text{Max}(Q)$ по известному $M[Q]$ с точностью на 3 порядка выше, чем позволяло неравенство Чебышева. Эту эмпирическую формулу можно считать приемлемой оценкой для всего класса систем, ограниченного перечисленными условиями.

Выводы

Результаты многочисленных имитационных экспериментов позволяют сформулировать следующие выводы.

1. Аналитические методы неприменимы для получения оценки максимального размера БП в ПУТС, так как дают слишком завышенную оценку.
2. Имитационное моделирование позволяет получить оценки на несколько порядков более точные, чем при использовании аналитических методов.
3. Полученная эмпирически аналитическая зависимость позволяет оценить емкость БП, зная средний уровень ее заполненности, при определенных условиях с точностью, на три порядка превышающей точность такой же оценки, полученной с использованием неравенства Чебышева.

Литература

1. Шнепс М. А. Системы распределения информации. Методы расчета. – М.: Связь, 1979. – 344 с.
2. Клейнрок Л. Теория массового обслуживания. – М.: Машиностроение, 1979. – 432 с.

Соснин Владимир Валерьевич

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, vsosnin@mail.ru

Алиев Тауфик Измаилович

— Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, aliev@dl.ifmo.ru

**АЛГОРИТМЫ ОПТИМИЗАЦИИ РАБОТЫ БЕСПРОВОДНОЙ
СЕНСОРНОЙ СЕТИ НА БАЗЕ ПРОТОКОЛА ZIGBEE**

С.В. Трифонов, Т.Е. Истомин, А.В. Чечендаев, М.И. Миненко, Я.А. Холодов

Рассматриваются вопросы функционирования беспроводных сетей на основе протокола ZigBee и проблемы энергосбережения с одновременным уменьшением времени доставки сообщений. Даны теоретические выкладки и описаны алгоритмы распределения ролей и установки расписаний для маршрутизаторов. Также приведены и проанализированы результаты проведенных экспериментов.

Ключевые слова: беспроводные сети, ZigBee, энергосбережение.

Введение

В работе рассматриваются беспроводные сети, для которых характерно сверхнизкое энергопотребление и большое количество миниатюрных устройств, обменивающихся относительно небольшим количеством информации. Существующие протоколы беспроводной связи, такие как Bluetooth и Wi-Fi, не учитывают эту специфику рассматриваемых сетей. Например, энергопотребление протокола Wi-Fi слишком велико, а попытка обеспечить универсальность протокола Bluetooth привела к его усложнению и неприменимости к широкому кругу задач, требующих дешевизны передающих устройств. Именно поэтому был разработан протокол ZigBee, учитывающий недостатки более ранних протоколов.

Целью данной работы является построение и изучение алгоритмов, управляющих работой беспроводной сети для минимизации энергопотребления и времени доставки сообщений.

1. Беспроводные сенсорные сети на основе протокола ZigBee

1.1. Основные характеристики ZigBee

За стандартом ZigBee [1] закреплены 27 каналов в трех частотных диапазонах – 2,4 ГГц (16 каналов), 915 МГц (10 каналов) и 868 МГц (1 канал). Максимальная скорость передачи данных для этих эфирных диапазонов составляет 250 кбит/с, 40 кбит/с и 20 кбит/с соответственно. Доступ к каналу осуществляется по контролю несущей, т.е. устройство сначала проверяет, не занят ли эфир, и только после этого начинает передачу.

Изначально стандарт ZigBee разрабатывался с целью максимально снизить энергопотребление устройств, задействованных в беспроводной сети. При этом большую часть времени аппаратура находится в спящем режиме, лишь изредка прослушивая эфир, а длительность жизни устройств повышается за счет увеличения задержек доставки сообщений, что не всегда приемлемо. Другая принципиальная проблема заключается в расширяемости сетей ZigBee, которая приводит, как будет указано ниже, к лишней трате энергии батарей.

1.2. Принцип работы протокола канального уровня

Протоколы физического и канального уровней ZigBee определены спецификацией IEEE 802.15.4 [2]. Существует два режима работы сети – маячковый и без маяков. В режиме без маяков сеть работает асинхронно, и маршрутизаторы должны постоянно прослушивать эфир, что не позволяет построить на основе этого режима сеть со сверхнизким энергопотреблением.

В маячковом режиме координатор и каждый из маршрутизаторов с определенным периодом посылают в эфир кадры, называемые маяками, позволяя производить синхронизацию часов на родительских и подчиненных узлах. Интервал между маяками

(*BI*) включает в себя активный период, называемый суперфреймом, и, возможно, неактивный период (рис. 1). Суперфрейм разделен на 16 одинаковых временных слотов, в течение которых возможна передача кадров данных. В течение неактивного периода все узлы могут входить в спящий режим, сохраняя тем самым энергию.

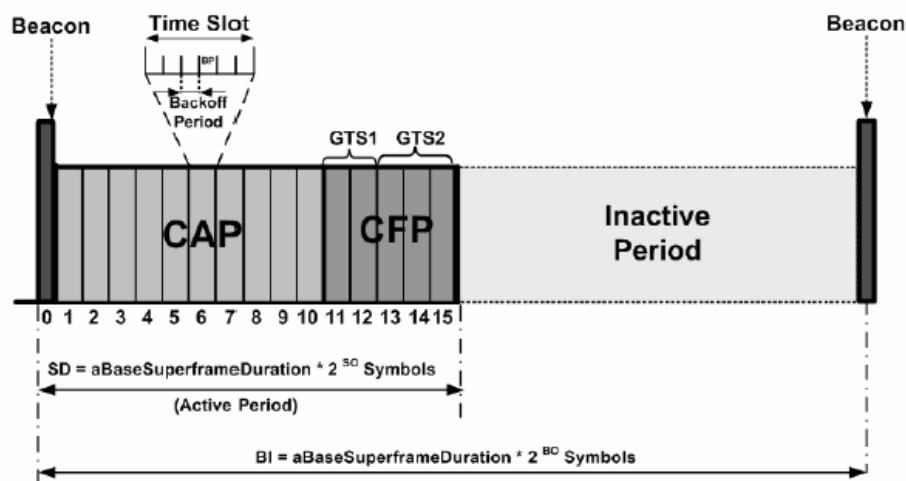


Рис. 1. Принцип работы сети в маячковом режиме

Интервал между маяками и длительность суперфрейма определяются параметрами Beacon Order (*BO*) и Superframe Order (*SO*) соответственно:

$$BI = aBaseSuperframeDuration \cdot 2^{BO}, \quad 0 \leq BO \leq 14,$$

$$SD = aBaseSuperframeDuration \cdot 2^{SO}, \quad 0 \leq SO \leq BO,$$

где *aBaseSuperframeDuration* обозначает минимальную длительность суперфрейма. Эта величина фиксирована и равна 960 символов (символ – 4 бита), что соответствует 15,36 мс, предполагая 250 кбит/с в частотном диапазоне 2,4 ГГц.

В течение периода конкурентного доступа (CAP) узлы соперничают за получение доступа к физической среде, используя механизм слотного CSMA-CA (slotted carrier sense multiple access with collision avoidance). Этот механизм довольно сложен, и его всестороннее исследование средствами имитационного моделирования представлено в [3]. В протоколе IEEE 802.15.4 также предусмотрен период неконкурентного доступа к среде (CFP), в течение которого узлам могут выделяться слоты гарантированного доступа (GTS).

<i>BO</i>	Интервал между маяками, с	Средняя сила тока, мА	Время жизни от CR2450, мес.	Время жизни от 2*AA, лет
2	0,061	10,58	0,078	0,027
3	0,122	5,31	0,156	0,054
4	0,245	2,67	0,311	0,108
5	0,491	1,35	0,613	0,213
6	0,983	0,69	1,192	0,414
7	1,966	0,36	2,255	0,783
8	3,932	0,20	4,070	1,413
9	7,864	0,12	6,810	2,364

Таблица 1. Рассчитанные параметры для различных значений Beacon Order

Зная характерное энергопотребление устройств [4], временные и другие параметры в различных режимах, можно теоретически рассчитать время жизни при питании от

различных батарей. Приведенные в табл. 1 данные рассчитаны для маршрутизаторов с учетом присутствия второго собственного маяка и следующего за ним суперфрейма ($SO = 0$). Уточним, что первый маяк посылается родителем данного маршрутизатора. В реальных ZigBee сетях возможность самоорганизации приводит к тому, что большинство устройств аппаратно является маршрутизаторами, и, следовательно, к передаче ими собственных маяков. Таким образом, даже конечные устройства, являясь потенциальными маршрутизаторами, передают собственные маяки.

2. Оптимизационные алгоритмы управления работой беспроводной сенсорной сети на основе протокола ZigBee

2.1. Постановка задачи оптимизации

Под оптимальностью мы будем понимать некоторую комбинацию различных задач, не разрешимых в общем случае одновременно, но для каждой из которых можно эмпирически построить решение, в том или ином смысле приближенное к оптимальному. Итак, основные задачи – минимизация энергопотребления и минимизация среднего или максимального времени доставки сообщения. Сформулируем основные предположения о структуре и назначении сети, в которых мы пытаемся разрешить эти задачи:

- подключения к сети новых устройств и изменения ее конфигурации происходят относительно редко (характерное время жизни неизменной конфигурации много больше времени доставки пакета);
- на сеть не возлагается задача передачи сигнала между произвольными узлами – достаточно иметь возможность передавать сигнал от узла к координатору и от координатора до узла;
- объем информации, генерируемый сетью в единицу времени, не превышает ее пропускной способности.

2.2. Задача минимизации энергопотребления

Ввиду предположений, введенных в п. 2.1, возникает возможность разделения устройств по ролям, энергопотребление в которых сильно различается. Эти роли таковы: конечное устройство, маршрутизатор и координатор. Идея заключается в следующем. Допустим, что после включения сети координатор узнает о присутствии в сети всех устройств, т.е. в дальнейшем предполагается, что новых устройств появиться не может. Тогда часть устройств оказывается маршрутизаторами, а часть – конечными устройствами дерева топологии сети, т.е. устройствами, не имеющими дочерних узлов. Это приводит к отсутствию необходимости в передаче конечными устройствами кадра маяка и участия в следующем за маяком суперфрейме, что снижает энергопотребление конечных устройств уже почти вдвое.

Дальнейшая оптимизация связана с возможностью конечного устройства засыпать сразу же после родительского маяка, не слушая суперфрейм, при условии, что устройству не адресовано данных (устройства могут определять это по содержанию кадра маяка) и у него нет данных для отправки. Эта оптимизация позволяет снизить энергопотребление еще в 3–4 раза.

Чем более быстрой является сеть, тем более эффективны обе оптимизации, поскольку в быстрых сетях устройства проводят во сне меньшую долю времени, а оптимизации не влияют на силу тока в режиме сна. В быстрых сетях батареи маршрутизаторов будут разряжаться в 6–8 раз быстрее, чем батареи конечных устройств, и время жизни всей сети будет определяться временем жизни маршрутизаторов. Чтобы этого избежать, предлагается следующий подход.

Рассмотрим сеть, состоящую из N узлов. Обозначим через $V = \{a_1, a_2, \dots, a_N\}$ множество всех узлов. Пусть два узла, которые находятся в области прямой видимости друг друга,

называются смежными. Таким образом, мы получим граф $G = \{V, E\}$, вершины которого – узлы $a_i \in V$, а ребра – пары смежных узлов $(a_i, a_j) \in E$. Для простоты считаем граф связанным и неориентированным. Узел a_0 – координатор сети. Построим все связанные подграфы T_k графа G , являющиеся деревом с корневым элементом a_0 и содержащие все вершины множества V каждый. В общем случае это можно сделать не одним способом, поэтому использован индекс $k \in \overline{1, K}$. Пусть R_k – набор всех маршрутизаторов графа T_k .

Если динамически менять роли узлов, то можно приблизить время жизни сети ко времени жизни конечных устройств. Это возможно за счет того, что большую часть времени каждый из узлов будет находиться в роли конечного устройства, лишь изредка участвуя в маршрутизации. При этом наборы одновременно работающих маршрутизаторов циклически сменяют друг друга, например, по прошествии определенного периода или уменьшении заряда батарей на определенный процент. Так или иначе, решения о перестроении топологии принимаются на координаторе сети.

Решим задачу максимизации времени жизни сети. Рассмотрим произвольное подмножество $\{R_{k_m}\}_{m=1}^M$ множества $\{R_k\}_{k=1}^K$. Имеем M_i – число наборов из $\{R_{k_m}\}_{m=1}^M$, содержащих a_i . Тогда средняя сила тока в узле a_i выразится формулой

$$I_i = I_R \left(\frac{M_i}{M} \right) + I_E \left(\frac{M - M_i}{M} \right) = I_E + (I_R - I_E) \frac{M_i}{M},$$

где I_R и I_E – средние за время BI силы тока в узлах, находящихся в роли маршрутизатора и конечного устройства соответственно. Время жизни сети

$$T = \min_i \frac{Q_i}{I_i} \rightarrow \max,$$

где Q_i – заряд батареи устройства a_i . Далее для простоты предполагаем, что в начальный момент времени $Q_i = Q$. Тогда

$$\max_i I_i = I_E + (I_R - I_E) \frac{\max_i M_i}{M} \rightarrow \min.$$

Если наборы $\{R_{k_m}\}_{m=1}^M$ независимы, то $\max_i M_i = 1$, и остается лишь условие

$$M \rightarrow \max.$$

Это и есть искомое условие. Итак, для максимизации времени жизни сети необходимо найти максимальное число независимых наборов маршрутизаторов.

2.3. Алгоритм распределения ролей

Требуется решить задачу о поиске максимального количества M независимых наборов маршрутизаторов $\{R_{k_m}\}_{m=1}^M$ графа $G = \{V, E\}$ и соответствующих им деревьев $\{T_m\}_{m=1}^M$, покрывающих всю сеть. Точное решение этой задачи для больших графов вызывает определенные трудности – для ее решения требуется много времени, что в реальных условиях неприемлемо. Поэтому будем использовать следующий алгоритм на графе, который позволяет найти не максимальное, а достаточно большое количество наборов маршрутизаторов.

1. Пусть вначале вершина a_0 (координатор) покрашена в красный цвет, а все остальные вершины не покрашены.

2. Покрасим в черный цвет всех неокрашенных соседей красных вершин.

3. Если при выполнении п. 2 не было покрашено ни одной вершины, то набор не удалось построить – прекращаем работу алгоритма. Иначе, продолжаем.

4. Рассмотрим множество черных вершин. Выберем из него вершину, имеющую наибольшее число неокрашенных соседей, и перекрасим ее в красный цвет.

5. Если все вершины окрашены, переходим к п. 6. Иначе, повторяем действия, начиная с п. 2.

6. Вершины, окрашенные в красный цвет (не считая a_0), являются одним из искоемых наборов. Красим их в зеленый цвет, a_0 – в красный, остальные делаем неокрашенными. Повторяем действия, начиная с п. 2.

Этот алгоритм является «жадным», т.е. на каждом шаге он старается покрасить как можно больше вершин графа, что обеспечивает одновременно минимальность получаемых наборов маршрутизаторов и приемлемую скорость работы алгоритма. В результате работы алгоритма мы получаем число M и наборы $R_m, m \in \overline{1, M}$. Для того чтобы получить дерево T_m , соответствующее набору R_m , пользуемся следующими правилами.

1. Вначале соединяем координатор со всеми его соседями.
2. К присоединенным соседям-маршрутизаторам подсоединяем их соседей.
3. Повторяем пункт 2, пока все узлы не будут соединены.

2.4. Модификация протокола сетевого уровня

Для реализации вышеописанной идеи требуется централизованное управление распределением ролей. Эту задачу решает координатор. При этом ему необходимы полные данные о графе G . Для получения этих данных после построения в стандартном режиме ZigBee произвольной топологии сети, включающей все устройства, нужно, чтобы каждое устройство в течение одного интервала между маяками прослушивало эфир. Поскольку в стандартном режиме ZigBee (в дальнейшем будем называть этот режим *спонтанным*) все устройства посылают маяки, таким образом будет установлена полная матрица смежности графа G .

Зная набор адресов A узлов, которые должны быть подчинены маршрутизатору, и набор адресов B своих текущих непосредственных подчиненных, маршрутизатор разрывает связи с узлами из B/A при помощи сообщения о переходе сети в режим построения фиксированной топологии – *заданный* режим (в противоположность спонтанному режиму). Если узел получает такое сообщение, то он передает его своим подчиненным, отсоединяется и начинает сканировать эфир в поисках маяка, в котором указан его адрес. Для подсоединения к себе узлов из A/B маршрутизатор добавляет в каждый отправляемый им кадр маяка адреса этих узлов.

Узлам из $A \cap B$ маршрутизатор сообщает о переходе сети в новый режим другим способом – рассылкой сообщений специального типа, содержащих структуру поддерева, за которую и будет отвечать принимающий узел. Такое сообщение посылается также и присоединившимся узлам из множества A/B . Каждый маршрутизатор получает ровно одно сообщение этого типа за все время построения топологии. Это обстоятельство особенно важно, чтобы не загружать сеть лишней рассылкой данных непосредственно из координатора.

При получении сообщения, содержащего структуру поддерева, узел начинает действовать по принципу, описанному выше, пока не будет построено все требуемое дерево топологии.

Понятно, что данный способ перехода из спонтанного режима в заданный не является единственным. Возможны и другие методы, каждый из которых имеет свои преимущества и недостатки. Выбор оптимального метода зависит от требований к сети и рассматриваемому переходному процессу.

2.5. Задача минимизации времени доставки сообщений о событиях координатору

Имея в виду предположения, сделанные в п. 2.1, рассчитаем время доставки Δ_a сообщений координатору от произвольного узла сети $a \in V$. При постоянной топологии сети сообщения координатору от a всегда передаются по одной и той же цепочке узлов от детей к родителям. Обозначим ее через $\{b_l\}_{l=0}^d$, причем $b_0 = a_0$ (координатор) и $b_d = a$.

Поскольку передача от дочернего узла к родителю может происходить только во время родительского суперфрейма (суперфрейма, непосредственно следующего за родительским маяком), то нельзя точно определить время между возникновением события и передачей сообщения о нем от b_d к b_{d-1} . Без дополнительных предположений о возникновении событий можно сказать только то, что это время есть некоторая случайная величина τ_0 , распределенная в интервале $[0, BI)$.

Дальнейшая часть времени доставки определяется задержками на маршрутизаторах. Задержка на маршрутизаторе равна промежутку времени между собственным суперфреймом и родительским суперфреймом на этом маршрутизаторе. В нашем случае множество маршрутизаторов $\{b_l\}_{l=1}^{d-1}$ состоит из $d-2$ элементов. Задержка на каждом из них – это число τ_{b_l} из интервала $[SD, BI - SD)$. Итак,

$$\Delta_a = \tau_0 + \sum_{l=1}^{d-1} \tau_{b_l}.$$

Далее будем считать случайную величину τ_0 равномерно распределенной в своем интервале. Тогда математическое ожидание $E \tau_0 = BI / 2$.

Можно изменять задержки на маршрутизаторах, устанавливая смещения собственного маяка маршрутизатора относительно маяка его родителя. Если собственный суперфрейм находится непосредственно перед суперфреймом родителя, то достигается минимальная задержка, однако этот вариант не всегда возможен из-за наложения суперфреймов. Расчет показывает, что среднее по всем узлам время доставки сообщений τ_T для произвольного дерева топологии T равно

$$\tau_T = \frac{BI}{2} + \frac{1}{N-1} \sum_{a \in V/a_0} w(T, a) \tau_a,$$

где $w(T, a)$ – размер поддеревы, корнем которого является узел a , не считая самого узла. Итак, задача минимизации среднего времени доставки сообщений о событиях в сети координатору для топологии T (т.е. задача $\tau_T \rightarrow \min$) сводится к минимизации величины

$$\sum_{a \in V/a_0} w(T, a) \tau_a \rightarrow \min.$$

Здесь предполагается, что BI является константой.

2.6. Алгоритм распределения слотов суперфреймов

Разделим интервал между маяками BI на $S = 2^{BO-SO}$ слотов. Каждый из них, таким образом, будет равен длительности суперфрейма SD . Поставим в соответствие каждому маршрутизатору $a \in R$ число $s = Sf(a) \in [0, S-1]$. Пусть число s называется номером слота суперфрейма узла a и определяет промежуток времени Δt между началом собственного маяка устройства a и началом маяка координатора сети следующим образом:

$$\Delta t = SD \cdot s.$$

Очевидно, что должно выполняться равенство $Sf(a_0) = 0$. Имея функцию $Sf : R \rightarrow \overline{0, S-1}$, мы можем установить величины $\tau_a = (Sf(parent(a)) - Sf(a)) \bmod S$ для любого $a \in R$. Таким образом, можно свести задачу предыдущего раздела к отысканию функции Sf , удовлетворяющей условиям:

- 1) $Sf(p) \neq Sf(c)$, если p является родителем c ;
- 2) $Sf(c) \neq Sf(n)$, если c является соседом n ;
- 3) $Sf(c) \neq Sf(l)$, если l является родителем соседа c ;
- 4) $Sf(c) \neq Sf(g)$, если g является соседом дочернего узла c ;

и минимизирующей значение величины τ_T .

Для приближенного решения этой задачи предлагается следующий простой алгоритм. Будем последовательно назначать каждому маршрутизатору значение функции Sf так, чтобы на каждом шаге выполнялись требуемые ограничения. Из допускаемых ограничениями значений функции будем выбирать значение, минимизирующее величину задержки τ_a на этом маршрутизаторе. Обход вершин начнем от координатора и продолжим в порядке уменьшения значений $w(T, a)$. Этот алгоритм дает только приближенное решение задачи, но обладает хорошей вычислительной сложностью.

3. Экспериментальные исследования

Был проведен ряд экспериментов по изучению работы алгоритмов, управляющих работой беспроводной сети ZigBee-устройств с целью достижения минимального энергопотребления и минимизации среднего или максимального времени доставки сообщения.

3.1. Измерение энергопотребления устройств

В основу экспериментальных измерений положен тот факт, что, имея экспериментально установленные осциллограммы суперфреймов устройств, можно измерить энергопотребление устройства за один собственный суперфрейм и один родительский суперфрейм, а также исследовать, как эти энергопотребления зависят от событий, происходящих во время соответствующих суперфреймов. Полученные данные позволили подсчитать среднее энергопотребление устройств при любых интервалах между маяками и трафике. Было сделано по три измерения для каждого из предполагаемых видов суперфреймов. Общая длина передаваемого сообщения во всех случаях – 50 байт. Интегрирование и усреднение полученных данных дает параметры суперфреймов, показанные в табл. 2–5.

Суперфрейм	Трафик	T , мс	ΣT , %	Q , А·с·10 ⁻⁴	σ_Q , %
Собственный	Нет	23,54	5,19	3,62	5,4
	К координатору	23,57	5,07	3,68	5,1
	От координатора	23,48	5,08	3,52	5,0
Родительский	Нет	27,87	5,11	4,90	5,3
	От координатора	27,79	5,08	4,88	5,9
	К координатору	27,89	5,03	4,80	5,2

Таблица 2. Измеренные параметры суперфреймов
(T – длительность суперфрейма, Q – энергопотребление за суперфрейм)

ВО	Средняя сила тока, мА	Время жизни от CR2450, мес.	Время жизни от 2*АА, лет
2	7,83	0,106	0,037
3	3,94	0,212	0,073
4	1,99	0,419	0,145
5	1,01	0,821	0,285
6	0,52	1,580	0,549
7	0,28	2,937	1,020
8	0,16	5,148	1,788
9	0,10	8,256	2,867

Таблица 3. Время жизни конечного устройства в сильно загруженной стабильной сети

ВО	Средняя сила тока, мА	Время жизни от CR2450, мес.	Время жизни от 2*АА, лет
2	13,81	0,060	0,021
3	6,92	0,120	0,042
4	3,48	0,239	0,083
5	1,76	0,473	0,164
6	0,90	0,925	0,321
7	0,47	1,771	0,615
8	0,25	3,265	1,134
9	0,14	5,645	1,960

Таблица 4. Время жизни маршрутизатора в сильно загруженной стабильной сети

ВО	2	3	4	5	6	7	8	9
г (эксперимент)	1,736	1,732	1,725	1,711	1,685	1,637	1,560	1,450
г (теория)	1,992	1,985	1,971	1,943	1,892	1,805	1,673	1,507

Таблица 5. Сравнение теоретических расчетов и экспериментальных данных.
(г – отношение средних сил токов маршрутизатора и конечного устройства)

На основе полученных данных можно сделать вывод о том, что влияние любого трафика на время жизни устройств очень мало и оценивается сверху величиной в 2 % от времен жизни соответствующего устройства. Отношения средних сил токов маршрутизатора и конечного устройства оказались на 13 % меньше, чем теоретические значения. Эти отношения характеризуют эффективность алгоритма распределения ролей.

3.2. Измерение времени доставки сообщений

Схема эксперимента понятна из рис. 2. Запускалась сеть из 8 расположенных в одной комнате устройств. Строилась топология «цепочка» из всех устройств. Устройство, которое периодически генерировало сообщения, ставилось последним в цепочке и связывалось проводом с компьютером. При генерации сообщения узел также посылал сигнал по проводу. Последовательно проводилось два испытания на одной и той же топологии. При этом использовались разные варианты установки расписания. Серым выделены моменты времени, в которые происходили замеры по часам на соответствующем устройстве. Искомое время доставки сообщений – $t_7 - t_2$. По результатам эксперимента построены гистограммы распределения сообщений по временам их доставки (рис. 3, а, б). Интервал усреднения гистограмм – 20 мс.

Наличие двух границ на гистограммах указывает на равномерное распределение времени ожидания $t_3 - t_2$ на конечном устройстве и практически постоянное время доставки за одну волну $t_4 - t_3$. Найденные значения этих границ и средних времен доставки отмечены на осях гистограмм. Они также указаны в табл. 5 вместе с другими величинами.

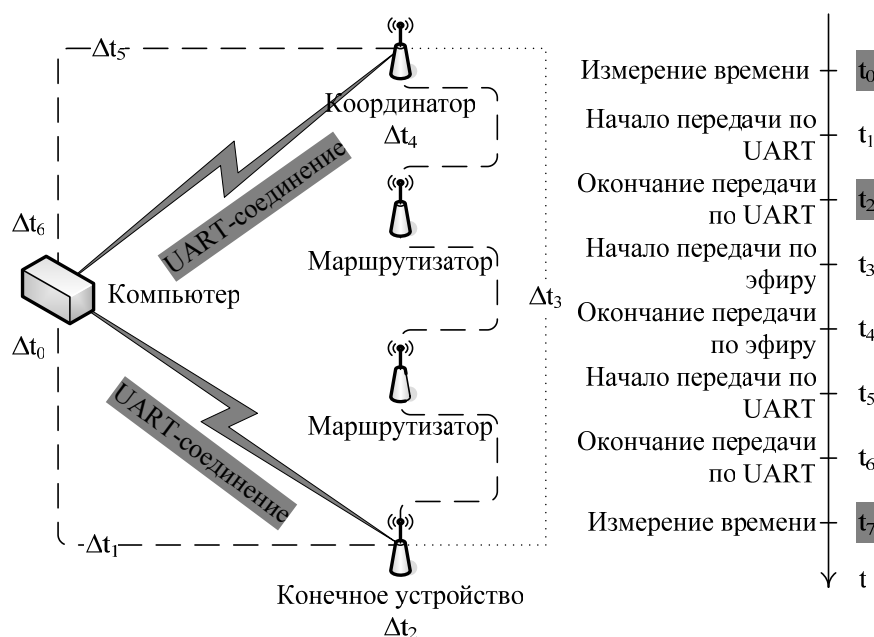


Рис. 2. Схема эксперимента и жизненный цикл сообщения

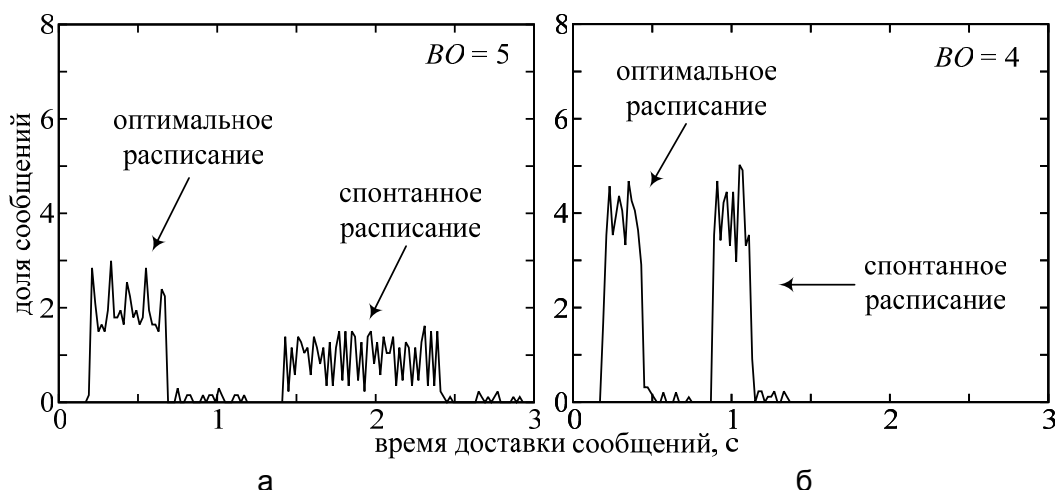


Рис. 3. Гистограммы распределения сообщений по временам их доставки

Проведенные испытания позволяют сделать следующие выводы. При включенном алгоритме установки расписаний происходит уменьшение среднего времени доставки в 3,1 и 4,2 раз в испытаниях с $BO = 4$ и $BO = 5$ соответственно. Минимальное время доставки за одну волну в оптимальном режиме не зависит от BO и составляет 200 мс. Полученные экспериментальные данные для оптимального расписания хорошо согласуются с теоретической оценкой.

BO	Расписание	Среднее время доставки, с	Минимальное время доставки за одну волну, с	Максимальное время доставки за одну волну, с
4	Случайное	1,009	0,890	1,110
	Оптимальное	0,321	0,190	0,430
5	Случайное	1,933	1,430	2,390
	Оптимальное	0,459	0,210	0,670

Таблица 5. Анализ гистограмм

Заключение

Целью работы являлось построение и изучение алгоритмов, управляющих работой беспроводной сети ZigBee-устройств для минимизации энергопотребления и времени доставки сообщений. Были предложены соответствующие алгоритмы, основывающиеся на особенностях протокола канального уровня IEEE 802.15.4 и идеях о динамическом изменении сетевых ролей устройств совместно с установкой расписаний для маршрутизаторов. Был проведен ряд экспериментальных измерений исследуемых характеристик сети, что позволило количественно оценить эффективность предложенных алгоритмов.

Литература

1. ZigBee Specification. ZigBee Alliance, 2006.
2. IEEE TG 15.4. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE standard for Information Technology, IEEE-SA Standards Board, 2003.
3. Koubâa A., Alves M., Tovar E. A Comprehensive Simulation Study of Slotted CSMA/CA for IEEE 802.15.4 Wireless Sensor Networks // Proceedings of the 6th IEEE International Workshop on Factory Communication Systems (WFCS 2006), Torino (Italy), 2006.
4. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver CC2422 Data Sheet, 2005.

Трифонов Сергей Владимирович

— Московский физико-технический институт, студент, serxa@mail.ru

Холодов Ярослав Александрович

— Московский физико-технический институт, кандидат физ.-мат. наук, доцент, kholodov@cres.mipt.ru

Миненко Максим Иванович

— Московский физико-технический институт, соискатель, mminenko2006@yandex.ru

Истомин Тимофей Евгеньевич

— Институт точной механики и вычислительной техники им. С.А. Лебедева РАН, инженер-программист, tim.ist@gmail.com

Чечендаев Алексей Владимирович

— Институт точной механики и вычислительной техники им. С.А. Лебедева РАН, руководитель проекта, a_chechendaev@ipmce.ru

УДК 004.9

ПРОГРАММИРОВАНИЕ ЭЛЕКТРОННОГО КУРСА НА ОСНОВЕ
АВТОМАТНОГО ПОДХОДА

М.В. Плешкова, А.В. Лямин

Описан электронный курс в системе дистанционного обучения AcademicNT, основанной на компетентностном модульно-рейтинговом подходе. Разработана графическая нотация для описания моделей электронного курса на основе конечно-автоматного подхода. Определены компоненты состояний. Построены типовые диаграммы переходов и проведен анализ зависимости количества состояний от количества модулей электронного курса и способа доступа к ним. Предложена методика моделирования логики курса в среде Stateflow системы Matlab.

Ключевые слова: электронный курс, теория конечных автоматов, диаграмма переходов.

Введение

Учебно-методический комплекс (УМК) в системе дистанционного обучения AcademicNT – совокупность учебно-методических материалов, необходимых и достаточных для организации учебного процесса по дисциплине и способствующих эффективному освоению студентами учебного материала. На рис. 1 изображена структура УМК.

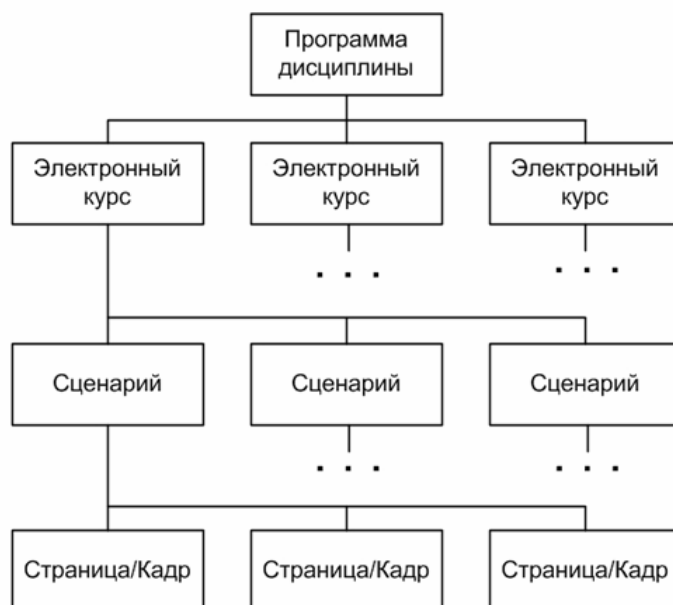


Рис. 1. Структура УМК

УМК основывается на рабочей программе соответствующей дисциплины. Программа дисциплины может содержать несколько электронных курсов. В электронном курсе определяется последовательность сценариев работы студента с системой. Сценарии могут быть представлены электронными тестами, виртуальными лабораториями, электронными практикумами, конспектами или информационными ресурсами. Каждый сценарий содержит набор страниц или кадров, которые описывают интерактивное взаимодействие с системой.

Основными компонентами электронного курса являются метаданные, оглавление, дерево переменных курса, список возможных состояний. В метаданных определяются название курса, автор, учебное заведение, в котором используется данный курс. Оглавление курса служит для ориентации студента по курсу и представляет собой иерархическую модульную структуру, каждый модуль которой может являться контейнером для других модулей или ссылкой на конкретный сценарий. Переменные курса используются для вычисления рейтинга студента по курсу и при проверке условий переходов в состояния компетентности. Совокупность всех объявленных переменных образует синтаксическое дерево, значение переменной может быть задано в виде числа или функции.

Состояния компетентностей определяются как требования к знаниям и практическим умениям, которые должны быть сформированы у студента при освоении соответствующих компетенций [1]. Состояние в курсе позволяет определить список доступных студенту сценариев и правила изменения переменных курса в зависимости от процента набранных баллов, затраченного времени и количества использованных попыток.

Применение теории конечных автоматов

В состояниях курса можно выделить следующие компоненты: события, условия и действия. К событиям относятся инициализация модуля и завершение сценария. В условиях определяется: количество запусков сценария с момента начала обучения; время, прошедшее с момента начала обучения; проценты набранных баллов по результатам как последней, так и всех предыдущих попыток; значения переменных курса. Под действиями понимаются следующее: изменение значения переменной курса, переход в новое состояние, запуск сценария.

На рис. 2 представлен процесс инициализации модуля. После инициализации модуля первым проверяется условие доступа к элементу. Если доступ разрешен, то предпринимается попытка перехода в новое состояние и происходит запуск сценария. По завершению сценария могут быть выполнены изменения переменных курса, переход в новое состояние и повторная инициализация модуля.

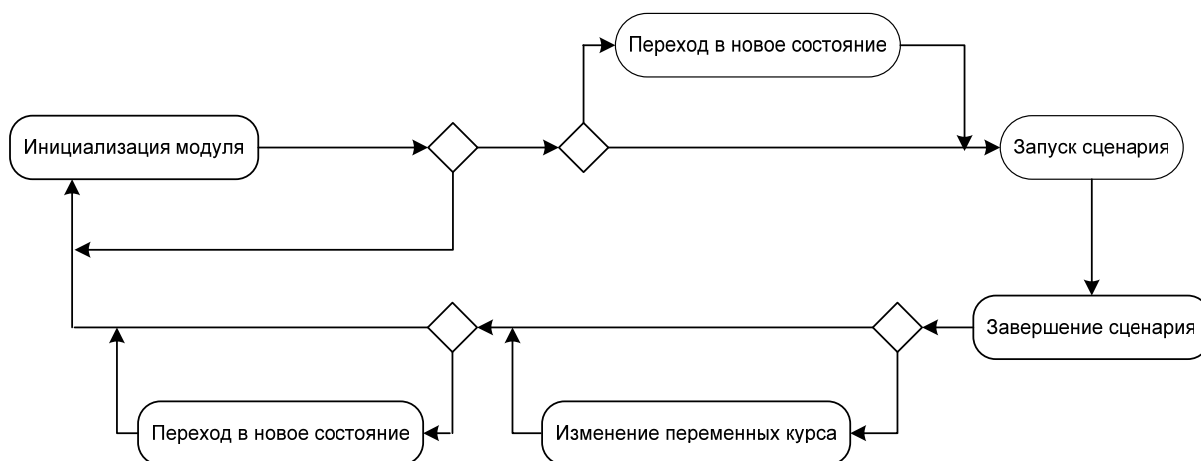


Рис. 2. Процесс инициализации

При работе с курсом студент приобретает определенные компетенции, в зависимости от результатов работы осуществляется переход студента из одного состояния курса в другое. Программирование данных переходов осуществляется на основе конечно-автоматного подхода. При этом автомат переходов имеет одно начальное состояние, которое определяет начало обучения, и одно конечное состояние, которое свидетельствует о получении результирующей компетенции. Переход из состояния в состояние

может осуществляться при инициализации модуля или при завершении сценария. Выходом автомата является новое значение рейтинга [2].

На рис. 3 приведено графическое изображение алгоритма прохождения электронного курса с двумя испытаниями, которые необходимо сдать в заранее определенном порядке. Прямоугольниками обозначены состояния, номер состояния указан в верхнем левом углу. Внутри прямоугольника окружностями обозначены тесты, около окружности стоит номер теста. Если вход в тест в некотором состоянии разрешен, то окружность, находящаяся в прямоугольнике, который соответствует этому состоянию, закрашена. Переходы из состояния в состояние обозначены стрелками. На стрелке указывается номер теста, который обусловил переход. Номера тестов на переходах имеют верхний индекс "*", "+" или "-". Символ "*" указывает на то, что переход вызван входом в тест. Символ "+" указывает на то, что переход вызван завершением работы с тестом, и попытка сдачи теста считается успешной. Символ "-" указывает на то, что переход вызван завершением работы с тестом, и попытка сдачи теста считается неуспешной. На рис. 3 обозначено: x_0 – начальное состояние студента в курсе, в котором доступен для сдачи первый контрольный элемент; x_1 – состояние аттестации, в которое переходит студент при выборе первого элемента, при этом входы во все обучающие элементы курса заблокированы. При неуспешной сдаче элемента осуществляется обратный переход в состояние x_0 , при успешной сдаче – переход в состояние x_3 , в котором доступен для сдачи следующий контрольный элемент; x_4 – состояние аттестации, в которое переходит студент при выборе второго контрольного элемента. При неуспешной сдаче элемента осуществляется обратный переход в состояние x_3 , при успешной сдаче – переход в конечное состояние x_6 , которое свидетельствует о том, что все этапы контроля пройдены. В данном курсе введены еще два дополнительных состояния x_2 и x_5 , которые используются для назначения дополнительных попыток на аттестацию по первому и второму контролю соответственно, переходы студента в эти состояния происходят по желанию преподавателя дисциплины.

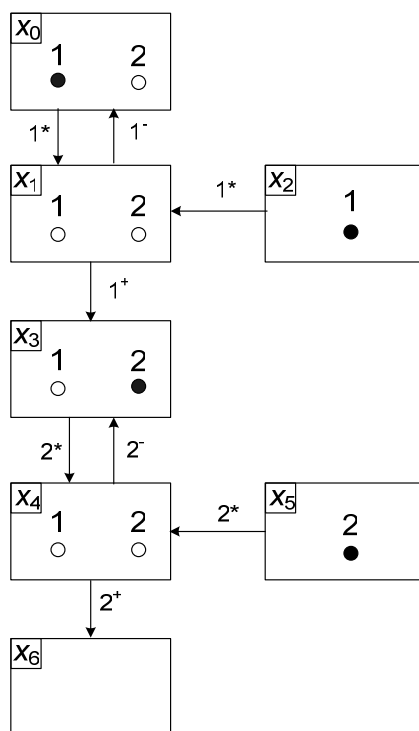


Рис. 3. Диаграмма состояний курса с последовательным прохождением элементов

На рис. 4 приведено графическое изображение алгоритма прохождения электронного курса с тремя испытаниями. В данном курсе испытания можно проходить в любом

порядке. Данный автомат не включает состояний для аттестаций и состояний для назначения дополнительных попыток, но без учета этих нюансов для описания автомата потребовалось 8 состояний. Если бы количество испытаний составляло 10, то потребовалось бы 1024 состояния, так как легко показать, что количество необходимых состояний вычисляется по формуле 2^N , где N – количество испытаний [3]. Быстрый рост состояний затрудняет программирование автомата и управление электронным курсом. Чтобы уменьшить количество состояний, необходимо расширить автомат – ввести дополнительные условия. В данном условии проверяются значения переменной курса, в которую заносится количество успешно сданных испытаний. На рис. 5 приведен пример автомата с использованием условий.

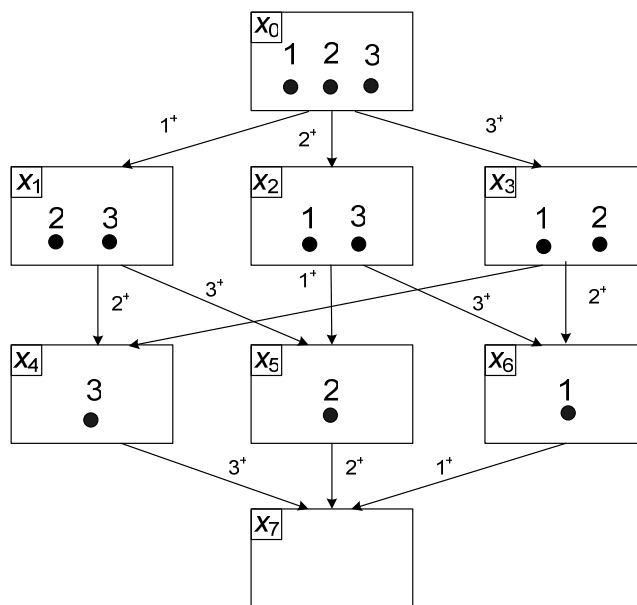


Рис. 4. Диаграмма состояний курса с произвольным прохождением элементов

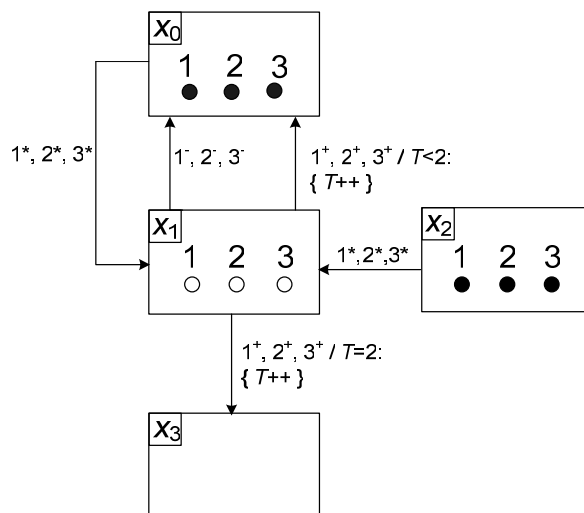


Рис. 5. Диаграмма состояний курса с произвольным прохождением элементов с использованием условий

Здесь x_0 – начальное состояние студента в курсе, в котором доступны для сдачи все контрольные элементы, x_1 – состояние аттестации, в которое переходит студент при выборе одного из элементов, при этом входы во все обучающие элементы курса забло-

кированы. При неуспешной сдаче элемента осуществляется обратный переход в состояние x_0 . При успешной сдаче происходит проверка условия успешной сдачи двух предыдущих испытаний, на рис. 3 условие обозначено как $T=2$, где T – количество испытаний. Если все контрольные элементы были успешно сданы, то T увеличивается на единицу и осуществляется переход в конечное состояние x_3 , в котором все контрольные элементы заблокированы. В противном случае, т.е. при $T < 2$, происходит обратный переход в состояние x_0 , где можно реализовать попытку сдачи других контрольных элементов, при этом переменная T также увеличивается на единицу. В данном автомате введено состояние дополнительной попытки x_2 , в котором по решению преподавателя студенту может быть предоставлена возможность пройти контрольный элемент без учета ограничений. Сравнивая рис. 4 и 5, легко заметить, что использование условий при программировании логики курса значительно уменьшает количество состояний.

В системе AcademicNT существуют электронные курсы, которые комбинируют в себе элементы с последовательным доступом и произвольным доступом. На рис. 6 приведен пример диаграммы курса со смешанным прохождением пяти контрольных точек.

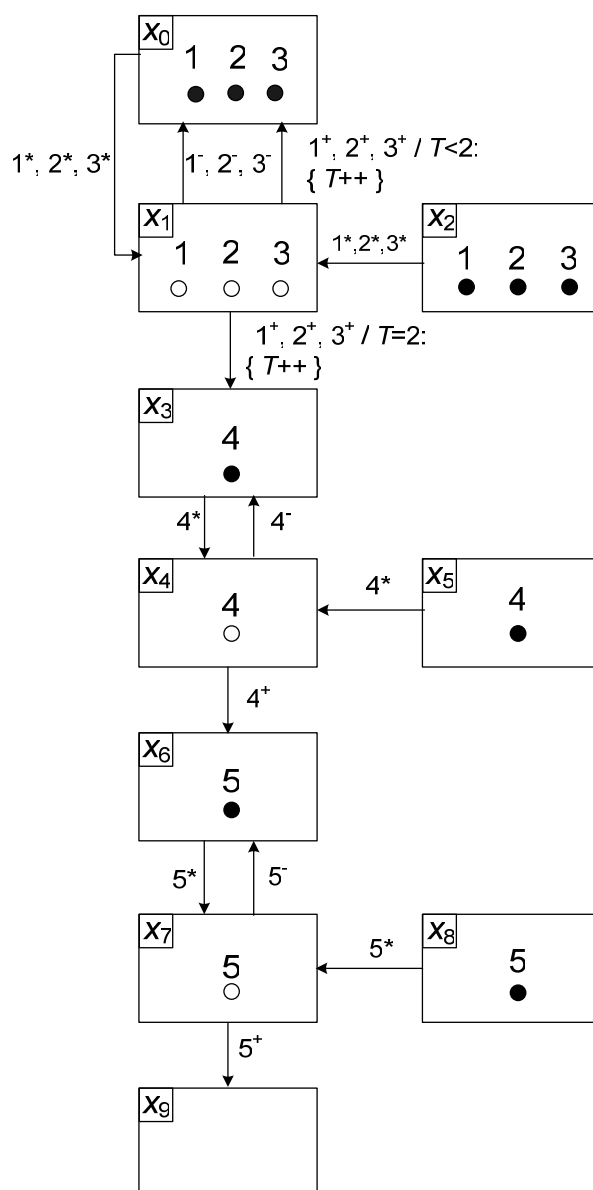


Рис. 6. Диаграмма состояний курса со смешанным прохождением элементов

В начале обучения доступны первые три испытания с произвольным доступом, после успешного прохождения которых можно перейти к последовательной сдаче следующих двух. Здесь x_0 – начальное состояние студента в курсе, в котором доступны для сдачи три контрольных элемента; x_1 – состояние аттестации, в которое переходит студент при выборе одного из элементов, при этом входы во все обучающие элементы курса заблокированы. При неуспешной сдаче элемента осуществляется обратный переход в состояние x_0 . В этом автомате также реализована проверка условий успешной сдачи предыдущих испытаний для уменьшения количества состояний курса.

Таким образом, при успешной сдаче проверяется, были ли успешно сданы остальные контрольные элементы, если не были, то также происходит обратный переход в состояние x_0 , где можно реализовать попытку сдачи других контрольных элементов. Если же все контрольные элементы были успешно сданы, то осуществляется переход в конечное состояние x_3 , в котором доступен для сдачи четвертый контрольный элемент; x_4 – состояние аттестации, в которое переходит студент при выборе четвертого элемента, при этом входы во все обучающие элементы курса заблокированы. При неуспешной сдаче элемента осуществляется обратный переход в состояние x_3 , при успешной сдаче – переход в состояние x_6 , в котором доступен для сдачи последний контрольный элемент; x_7 – состояние аттестации, в которое переходит студент при выборе пятого контрольного элемента. При неуспешной сдаче элемента осуществляется обратный переход в состояние x_7 , при успешной сдаче – переход в конечное состояние x_9 , которое свидетельствует о том, что все этапы контроля пройдены. Также введены специальные состояния дополнительной попытки x_2 , x_5 и x_8 , в которых по решению преподавателя студенту может быть предоставлена возможность пройти контрольные элементы без учета ограничений.

Моделирование курса

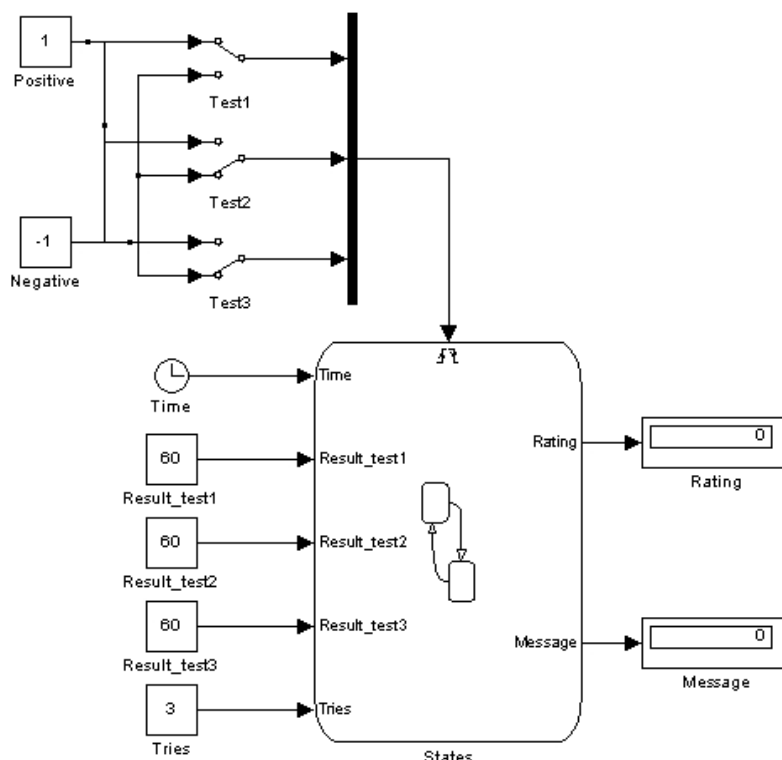


Рис. 7. Схема моделирования курса

Stateflow модуля Simulink системы Matlab представляет собой интерактивную среду, управляемую событиями. Среда Stateflow позволяет создавать модели, основанные на конечных автоматах в естественной и понятной форме, поэтому данный инст-

румент является наиболее подходящим для моделирования логики электронного курса, описанной выше.

Модель Stateflow является частью модели Simulink. На рис. 7 представлена схема моделирования курса, состоящего из трех испытаний. Таким образом, ключи управления выполнением диаграммы Stateflow, внутри которой описана логика курса. На вход модуля Stateflow подаются результаты сдачи испытаний Result_test1, Result_test2 и Result_test3, время сдачи Time, количество попыток Tries. Выходами являются рейтинг по курсу, который выводится на экране Rating, а также сообщения системы, которые вызывают доступность элемента и выводятся на экране Message. Сообщение «0» указывает на то, что в данный момент элемент заблокирован, сообщение «1» – что доступен.

На рис. 8 показана диаграмма Stateflow схемы моделирования курса, показанной выше. Состояниями здесь являются состояния сдачи тестов. Переходы между состояниями указаны стрелками, рядом с которыми определены условия этих переходов. В условиях происходит учет попыток, времени, затраченного на испытание, а также проверка полученного процента баллов. Над переходами указаны условия сдачи и подсчета баллов. Внизу определены функции подсчета рейтинга, которые вызываются внутри состояний, а также пересчитываются при переходах между состояниями.

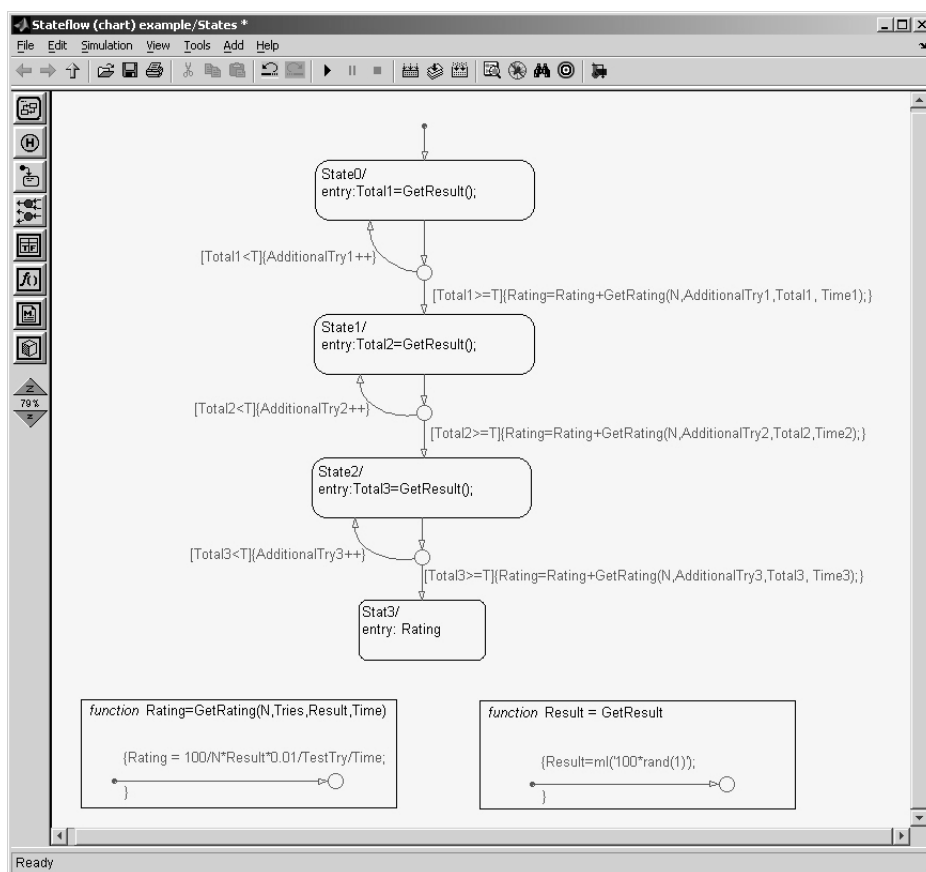


Рис. 8. Диаграмма Stateflow

Заключение

В статье введен формализм, который позволяет описать модель электронного курса. Выявлены наиболее важные условия, которые влияют на состояния компетентности и на рейтинг студента. Произведена оценка количества состояний в зависимости от правил доступа к модулям электронного курса. Разработана расширенная модель описания курса, позволяющая упростить его программирование и сопровождение за

счет уменьшения минимально необходимого количества состояний. Предложена методика моделирования электронных курсов в среде Stateflow системы Matlab.

Представленные в работе результаты создают основу для разработки новых алгоритмов и инструментальных средств, позволяющих автоматизировать процесс программирования и модификации описаний электронных курсов с целью снижения трудоемкости разработки электронных УМК.

Литература

1. Васильев В.Н., Лисицына Л.С., Лямин А.В. Совершенствование СППК в области ИКТ на основе технологий сетевой ИС // Опыт использования сетевых информационных технологий и систем в образовательной и научно-методической деятельности. Сборник научно-методических статей. – СПб: СПбГУ ИТМО, 2007. – С.75–77.
2. Лямин А.В., Плешкова М.В. Разработка среды программирования структуры электронных курсов // Научно-технический вестник СПбГУ ИТМО. Выпуск 45. Информационные технологии. – СПб: СПбГУ ИТМО, 2007. – С.164–166.
3. Лямин А.В., Плешкова М.В. Программирование структуры электронного курса // Труды XIV Всероссийской научно-методической конференции «Телематика'2007». – Санкт-Петербург, 2007. – Т. 1. – С. 211–212.

Плешкова Мария Витальевна

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, maria@cde.ifmo.ru

Лямин Андрей Владимирович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, lyamin@cde.ifmo.ru

УДК 004.942

МЕТОДИКА РАСЧЕТА ВЫЧИСЛИТЕЛЬНОЙ МОЩНОСТИ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА ДЛЯ ПРОВЕДЕНИЯ ЕГЭ В КОМПЬЮТЕРНОЙ ФОРМЕ

А.А. Скшидлевский, А.В. Лямин

Статья посвящена вопросам проектирования и развертывания больших программно-аппаратных комплексов, выбору оптимального варианта конфигурации аппаратной базы. Разработанная методика и алгоритмы позволяют определить ряд характеристик аппаратной части комплекса для проведения ЕГЭ в компьютерной форме, основываясь на заданном количестве пользователей в системе.

Ключевые слова: имитационное моделирование, методика, алгоритмы, программно-аппаратные комплексы.

Введение

Выбор оборудования и аппаратных мощностей для развертывания больших распределенных программных комплексов, таких как система для проведения ЕГЭ в компьютерной форме, всегда очень сложен [1]. Сложность заключается в оценке нагрузки на оборудование при использовании системы большим количеством пользователей. Единственным способом выяснить, сколько пользователей выдержит та или иная конфигурация, являются испытания на реальном серверном оборудовании, а это уже предполагает наличие оборудования, т.е. его приобретение. Какое оборудование необходимо приобрести – в данном случае определя-

ется простыми типовыми подсчетами и предположениями. Такой подход не может гарантировать обеспечение заданной производительности, а причиной сбоя системы может послужить перегрузка, которая будет являться следствием неправильного выбора. Оценить реальную производительность практически невозможно без оценки реакции конкретной системы и прогнозирования ее поведения при различных нагрузках.

Для решения данной проблемы была разработана методика, позволяющая оценить необходимую вычислительную мощность программно-аппаратного комплекса для обеспечения работы заданного количества пользователей в системе, на примере системы для проведения ЕГЭ в компьютерной форме. Оценке подлежит количество вычислительных ядер на сервере, объем оперативной памяти и пропускная способность Интернет-канала. Данная методика дает математическое обоснование для выбора оборудования, основанное на результатах прогнозирования поведения конкретной системы, и необходима при проектировании и развертывании подобных систем. В основе методики лежит имитационное моделирование, которое включает в себя несколько этапов:

- построение структуры имитационной модели серверной части системы;
- апробация системы и сбор статистической информации;
- анализ статистических данных и определение необходимых для моделирования параметров;
- оценка адекватности имитационной модели;
- экспериментальные исследования модели серверной части системы;
- расчет зависимостей на основе данных моделирования.

Построение имитационной модели

Первым этапом в разработке методики является построение имитационной модели [2]. По определению, имитационная модель – это математическая модель изучаемой системы, предназначенная для использования в процессе машинной имитации [3]. От точности построения этой модели, а также от выбора входных и выходных параметров зависит адекватность всех результатов моделирования. Модель должна быть максимально приближена к реальной системе и учитывать все факторы, которые могут влиять на результат работы системы. Чем детальнее проработана модель, тем точнее будут полученные с помощью нее результаты. На рис. 1 приведена построенная имитационная модель системы для проведения ЕГЭ в компьютерной форме.

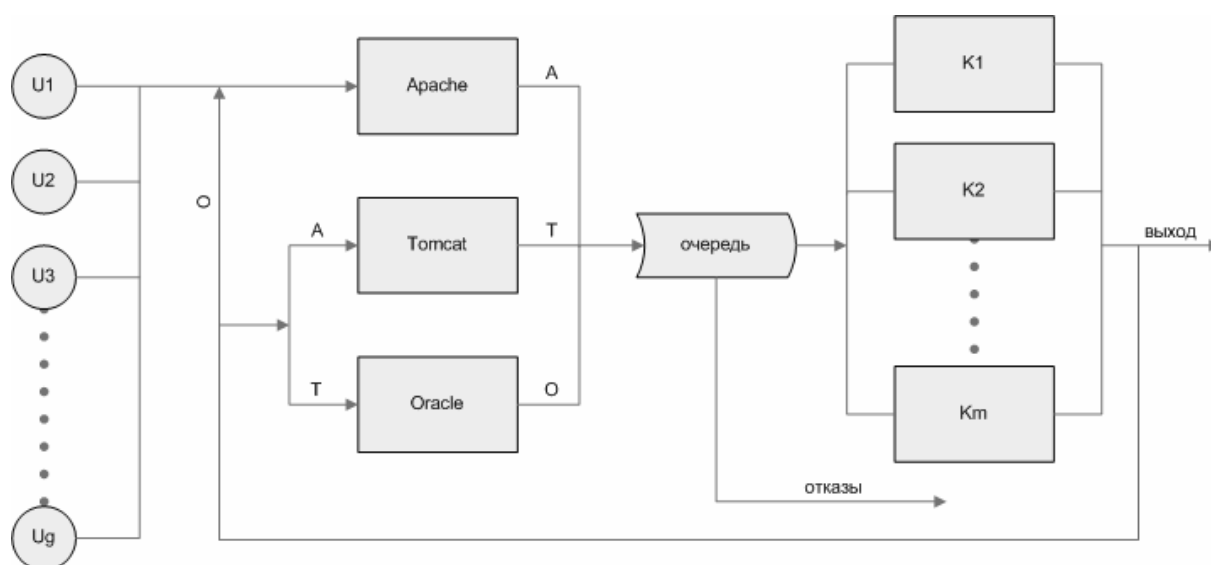


Рис. 1. Схема имитационной модели

Имитационная модель реализована в среде MATLAB [4]. Данная модель учитывает влияние отдельных компонентов на сервере с несколькими вычислительными ядрами. Входными параметрами имитационной модели являются:

- количество пользователей в системе g ;
- количество ядер на сервере m ;
- среднее время обслуживания запросов одним вычислительным ядром μ_R ;
- средний интервал времени между поступлением запросов к системе μ_S .

Параметр μ_S считается по формуле $\mu_S = \frac{\mu_H}{g}$, где μ_H – интервал времени между обращениями к системе одного пользователя.

Данная схема предполагает наличие двух генераторов псевдослучайных последовательностей для времени обслуживания запросов R и интервалов между поступлением запросов S . Время обслуживания каждого запроса является случайной величиной, не коррелированной с интервалами поступления запросов.

Изначально в системе задается множество пользователей g . Каждый пользователь через случайные интервалы времени H генерирует требование со случайным интервалом времени обслуживания R . Каждое требование обрабатывается в три этапа одним из вычислительных ядер m . Только что поступившее требование имеет состояние «А», так как первый этап – это обработка требования веб-сервером Apache. После завершения первого этапа состояние требования меняется на «Т», и оно поступает на обработку веб-сервером Tomcat. Завершающим этапом является обработка требования системой управления базой данных Oracle, перед этим состояние требования меняется на «О». Требования от пользователей поступают и обрабатываются на протяжении всего времени моделирования.

Если при поступлении требования все вычислительные ядра заняты, требование становится в очередь. Очередь организована по принципу «первый пришел – первый ушел». При переполнении очереди происходит отказ в обслуживании требования, и оно исключается из дальнейшей обработки.

Оценке подлежат следующие параметры:

- коэффициент использования системы ρ ;
- средняя задержка в очереди d ;
- среднее время ожидания w ;
- среднее по времени число требований в очереди Q ;
- среднее по времени число требований в системе L .

Сбор и анализ статистической информации

Для вычисления оценок параметров имитационной модели были проведены апробации системы и собраны соответствующие данные. Апробация системы проводилась в двух регионах Российской Федерации по схеме с использованием кластерных технологий для территориально распределенных узлов. Собранные данные относятся к работе системы на серверах с процессором UltraSPARC IV 1.35 ГГц под управлением операционной системы Solaris 10. В качестве веб-серверов использовались Apache 2 и Tomcat 6. Система управления базой данных – Oracle 10g. В расчет принималась следующая информация:

- количество пользователей в системе;
- время обработки запросов сервером;
- загрузка ядер процессора на сервере;
- использование памяти на сервере;
- пропускная способность Интернет-канала.

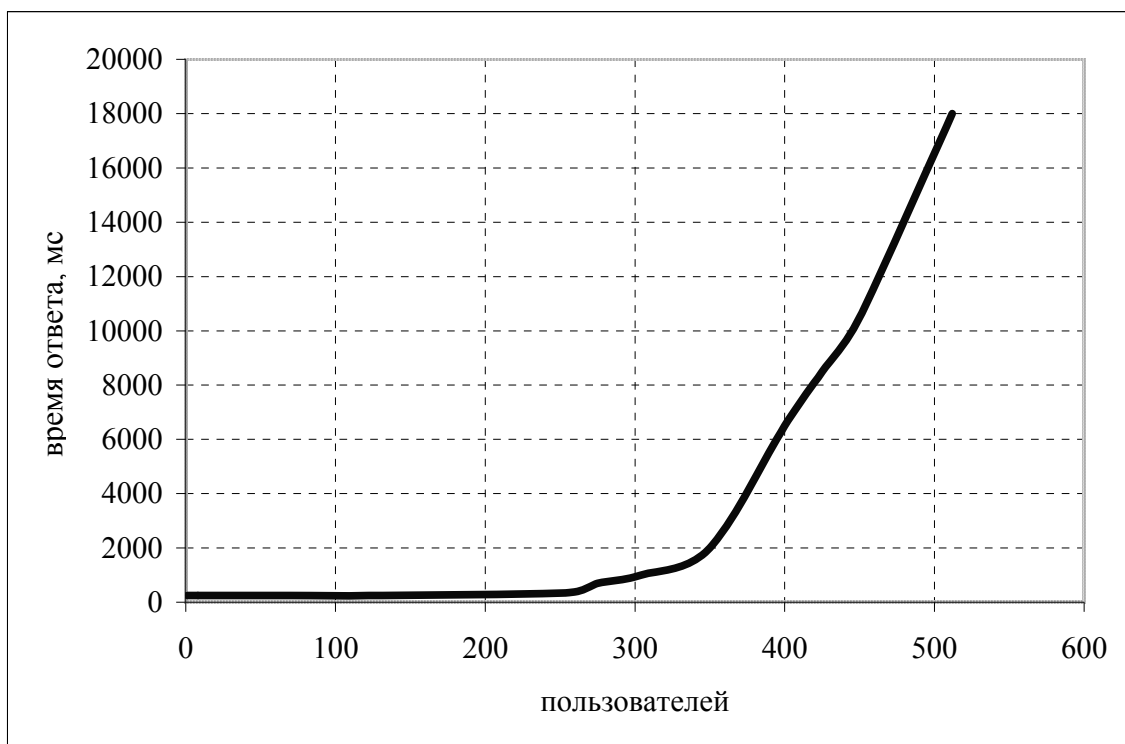


Рис. 2. Зависимость времени ответа сервера от количества пользователей в системе

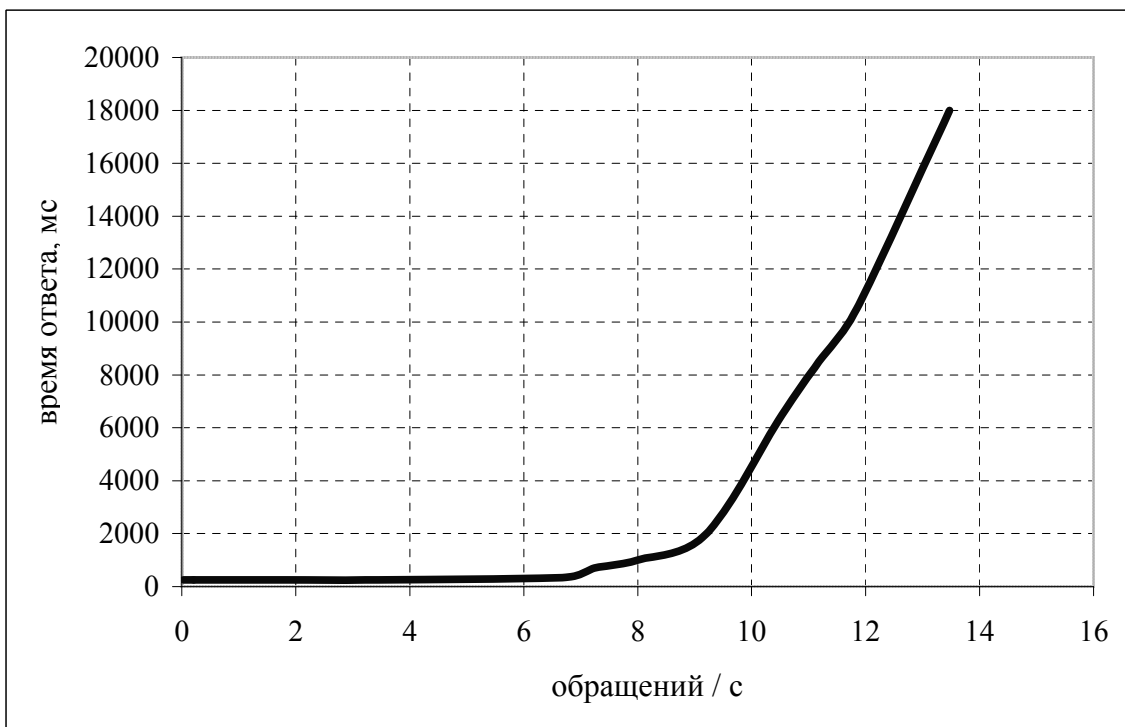


Рис. 3. Зависимость времени ответа сервера от количества обращений к системе за секунду

Исходя из собранных данных, среднее время обработки одного запроса одним вычислительным ядром на сервере с процессором UltraSPARC IV составляет 850 мс. Интервал между поступлениями запросов от одного пользователя в среднем составляет 38 с. На основе полученных данных была выведена зависимость времени ответа сервера от количества обращений к системе за секунду и от количества пользователей в системе. Данные зависимости отображены на рис. 2 и 3. При этом оптимальная нагрузка на один

сервер с четырьмя 2-ядерными процессорами UltraSPARC IV составляет 8 обращений к серверу за секунду, что примерно эквивалентно 300 пользователям в системе.

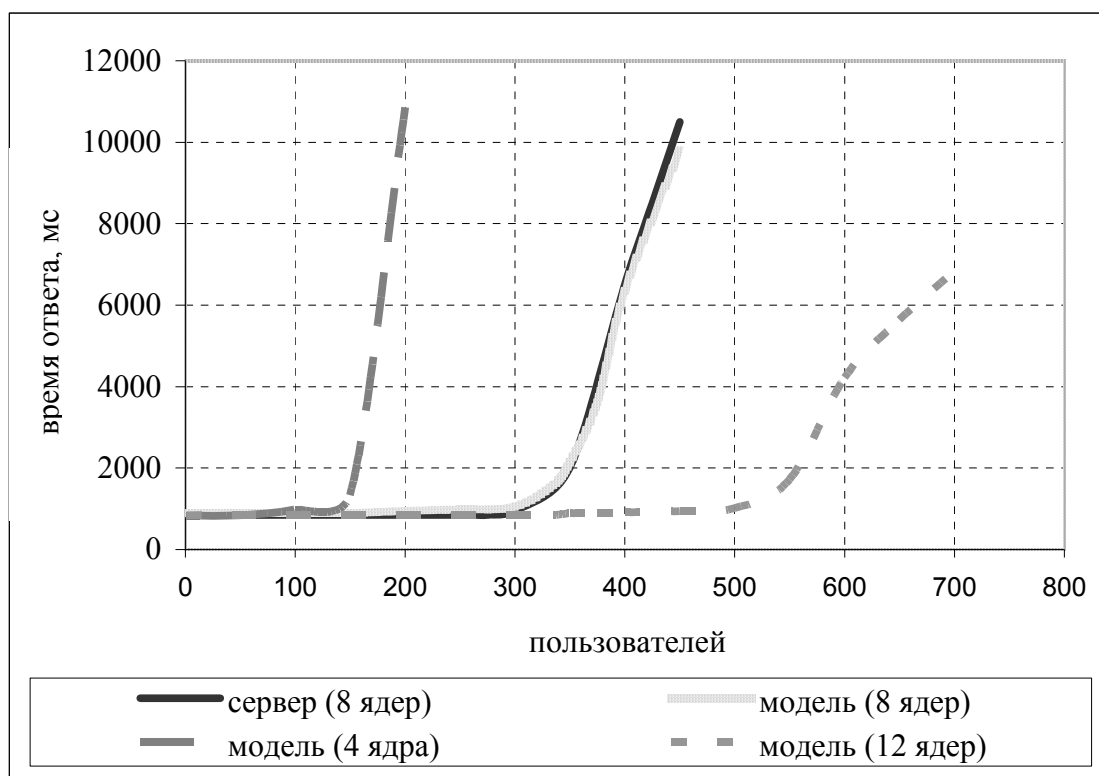


Рис. 4. Сравнение данных по производительности системы для имитационной модели и реального сервера

Оценка адекватности имитационной модели

Далее была проведена оценка адекватности имитационной модели. На рис. 4 показаны графики зависимости времени ответа сервера от количества пользователей в системе для реального сервера и имитационной модели с разными входными данными. Характер реакций реального сервера и имитационной модели совпадает, что говорит об адекватности имитационной модели.

Моделирование серверной части системы и расчет зависимостей

После того, как построена имитационная модель, собраны все необходимые данные, проведена оценка адекватности модели, идет следующий этап – моделирование системы и расчет зависимостей по результатам моделирования [2, 5].

Определение необходимого числа экспериментов

Эксперимент представляет собой прогон созданной компьютерной имитационной модели. Количество необходимых прогонов n определяется по формуле

$$n = \frac{t_{(9; 0,5125)}^2 \sigma^2}{\varepsilon^2},$$

где $t_{(9; 0,5125)} = 2.262$ – квантиль распределения Стьюдента с девятью степенями свободы порядка 0,975, σ^2 – оценка дисперсии наблюдений, ε – точность оценивания, равная 1% от среднего значения параметра. В табл. 1 приведены результаты предвари-

тельных прогонов. Таким образом, необходимое количество прогонов для каждой точки факторного плана будет составлять 18.

	S_0	R_0	ρ	d	w	Q	L
1	1622718208	637808128	0,7046	0,0568	0,9427	0,0389	5,6760
2	1885565696	1954793472	0,7021	0,0565	0,9324	0,0374	5,6546
3	42013824	1636720640	0,7036	0,0565	0,9362	0,0373	5,6665
4	93708672	184094208	0,7052	0,0565	0,9395	0,0372	5,6791
5	1347465280	1700077696	0,7047	0,0565	0,9398	0,0372	5,6748
6	1082365216	221137792	0,7068	0,0580	0,9414	0,0375	5,6916
7	1903783040	19048960	0,7056	0,0574	0,9431	0,0377	5,6821
8	1286922736	1380240160	0,7049	0,0574	0,9416	0,0377	5,6769
9	216738304	347292928	0,7045	0,0557	0,9399	0,0383	5,6745
10	189554432	1182352640	0,7030	0,0548	0,9375	0,0361	5,6603
Оценки							
$\bar{\epsilon}$			0,7045	0,0566	0,9394	0,0375	5,6736
$\bar{\epsilon}^2$			$1,78 \cdot 10^{-6}$	$8,1878 \cdot 10^{-7}$	$1,0743 \cdot 10^{-5}$	$5,4011 \cdot 10^{-7}$	$1,1486 \cdot 10^{-4}$
Количество экспериментов							
ε			0,007	0,0006	0,0094	0,0004	0,0567
n			1	12	1	18	1

Таблица 1. Результаты предварительных прогонов

Планирование экспериментов с помощью полного факторного плана

В терминологии планирования экспериментов входные переменные и структурные допущения, составляющие модель, называются факторами, а выходные показатели работы – откликами. Факторы могут быть либо количественными, либо качественными. Различают управляемые и неуправляемые факторы в зависимости от того, может ли руководство соответствующих реальных систем управлять ими. В имитационном моделировании доступно управление всеми факторами, независимо от их практической управляемости.

При моделировании перед выполнением прогонов модели планирование экспериментов позволяет решить, какие именно конфигурации следует создавать, чтобы получить необходимую информацию при наименьшем объеме моделирования.

Стратегией, с помощью которой можно измерять взаимодействие, является факторный план типа 2^k , называемый еще полным факторным планом. Выбираются два уровня каждого фактора, а затем проводятся имитационные прогоны для каждой из 2^k возможных комбинаций уровней факторов, называемых точками плана. Обычно знак «–» связывается с одним уровнем фактора, а «+» – с другим. В целом уровни должны быть противоположны друг другу, но в реальных пределах. Уровни нельзя указывать так, чтобы они были на большом расстоянии друг от друга, иначе могут скрываться важные аспекты отклика.

Для составления плана эксперимента выделим следующие факторы: количество ядер m , среднее время поступления μ_S , среднее время обработки μ_R . Выбранные значения уровней определяются следующими условиями:

- максимизация установившегося значения коэффициента использования системы,
- сохранение ограниченности средней задержки в очереди, среднего числа требований в очереди и системе для каждого набора значений уровней.

В табл. 2 представлена схема кодирования выбранных уровней факторов.

Фактор	-	+
m	7	8
μ_S	0,125	0,152
μ_R	0,8	0,85

Таблица 2. Схема кодирования для факторов m , μ_S и μ_R

Точка плана	m	μ_S	μ_R	ϕ	$\mathcal{E}$	$\mathbb{E}$	$\mathcal{G}$	$\mathcal{E}$
1	-	-	-	0,9052	0,1794	0,9708	0,9021	7,2386
2	+	-	-	0,7924	0,0843	0,8769	0,1828	6,5223
3	-	+	-	0,7551	0,0680	0,8689	0,0961	5,3816
4	+	+	-	0,6645	0,0423	0,8649	0,0136	5,3298
5	-	-	+	0,964	0,3538	1,2215	2,3304	9,0783
6	+	-	+	0,8506	0,1197	0,9717	0,4014	7,2063
7	-	+	+	0,802	0,1002	0,9747	0,2093	5,8233
8	+	+	+	0,7049	0,0575	0,9391	0,0395	5,6784

Таблица 3. Матрица и результаты моделирования для факторного плана типа 2^3

Результаты эксперимента представлены в табл. 3. Запись такого массива, называемого матрицей плана, обеспечивает вычисление эффектов факторов и взаимодействий между ними. Всего выполнялось $n=18$ независимых повторных прогонов имитационной модели на каждую точку плана. Для каждого прогона использовались разные последовательности случайных чисел, статистические счетчики переводились в исходное состояние в конце каждого прогона, и при каждом прогоне использовались одни и те же исходные условия.

Каждый отклик представляет собой среднее значение 18 независимых прогонов с постоянными для каждой точки плана параметрами m , μ_S и μ_R .

Построение уравнения регрессии

Имитационную модель можно рассматривать как механизм преобразования входных параметров в выходные показатели работы. В этом смысле моделирование является всего лишь функцией с неизвестной явной формой.

Регрессионный анализ результатов моделирования дает возможность построить аналитическую модель, наилучшим образом соответствующую набору данных, полученных в ходе машинного эксперимента с имитационной моделью. Под наилучшим соответствием понимается минимизированная функция ошибки, являющаяся разностью между прогнозируемой моделью и данными эксперимента. Наибольшее применение нашли модели в виде алгебраических полиномов. Для факторного плана 2^3 полиномиальная регрессионная модель будет иметь вид

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_{12}x_1x_2 + a_{13}x_1x_3 + a_{23}x_2x_3 + a_{123}x_1x_2x_3, \quad (1)$$

где y – отклик, x_1, x_2, x_3 – факторы, $a_0, a_1, a_2, a_3, a_{12}, a_{13}, a_{23}, a_{123}$ – неизвестные коэффициенты. В табл. 4 приведены оценки коэффициентов выбранной регрессионной

модели, найденной по методу наименьших квадратов, которые позволяют спрогнозировать реакцию откликов ρ , d , w , Q и L на изменение факторов m , μ_S и μ_R .

Коэффициент	ρ	d	w	Q	L
a_0	0,7924	-16,3677	-18,1130	-143,4729	-142,4250
a_1	-0,2073	1,8235	2,0000	15,8329	14,8358
a_2	-0,8208	-9,1661	-13,8138	-71,4728	-33,5881
a_3	2,8987	24,9710	28,9047	217,2222	231,0124
a_{12}	0,8304	2,4442	3,1991	22,4286	23,5160
a_{13}	-0,0106	-2,7998	-3,1372	-24,3753	-23,2856
a_{23}	-13,1383	-16,1852	-16,5599	-154,3389	-259,3334
a_{123}	-0,0077	0,1571	0,1601	1,4455	1,3685

Табл. 4. Коэффициенты модели регрессионных уравнений

В частности, подставляя полученные коэффициенты в формулу (1), получим уравнение зависимости коэффициента использования системы с фиксированным средним временем обработки требований 0,85 с от количества вычислительных ядер на сервере и среднего интервала времени поступления требований в систему, который обратно пропорционален количеству пользователей в системе:

$$\rho = 3,3 - 0,2 \cdot m - \mu_S \cdot (12 - 0,82 \cdot m). \quad (2)$$

В качестве характеристики вычислительной мощности взят коэффициент использования системы, рекомендуемое значение которого составляет 0,7. Таким образом, используя формулу (2), по количеству пользователей в системе можно определить необходимое число вычислительных ядер на сервере.

Расчет необходимого объема оперативной памяти и пропускной способности Интернет-канала

Для расчета необходимого минимального объема оперативной памяти на сервере V Мбайт и пропускной способности Интернет-канала C кбит/с выведены формулы, полученные на основе апробаций системы при проведении ЕГЭ в компьютерной форме:

$$V = V_0 + 2 \cdot g + 18 \cdot N,$$

$$C = 16 \cdot N,$$

где V_0 – минимальное количество памяти, выделяемое под операционную систему Solaris, систему управления базой данных Oracle, веб-серверы Apache и Tomcat;

$N = \frac{g}{\mu_H}$ – количество обращений к серверу за секунду.

Заключение

Представленная в работе имитационная модель и методика позволяет определить ряд характеристик аппаратной части комплекса для проведения ЕГЭ в компьютерной форме, основываясь на заданном количестве пользователей в системе. Адаптация имитационной модели к конкретной схеме организации системы позволяет применять эту методику и для других возможных конфигураций комплекса.

Полученные в работе результаты являются основой для построения регрессионных моделей программно-аппаратных платформ, отличающихся от рассмотренной в данной статье.

Литература

1. Таненбаум Э., ван Стен М. Распределенные системы. Принципы и парадигмы. – СПб: Питер, 2003. – 877 с.
2. Кельтон В., Лоу А. Имитационное моделирование. – 3-е изд. – СПб: Питер; Киев: Издательская группа BHV, 2004. – 847 с.
3. Трусова П.В. Введение в математическое моделирование: Учеб. пособие. – М.: Логос, 2005. – 440 с.
4. Иглин С.П. Математические расчеты на базе Matlab. – СПб: Издательство «BHV-Санкт-Петербург», 2005. – 640 с.
5. Максимова Ю.Д. Вероятностные разделы математики. – СПб: «Иван Федоров», 2001. – 592 с.

Скшидлевский Антон Алексеевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, anton@cde.ifmo.ru

Лямин Андрей Владимирович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, lyamin@cde.ifmo.ru

УДК 004.02; 004.58

КООРДИНАЦИЯ УПРАВЛЯЮЩИХ ВОЗДЕЙСТВИЙ ПРИ ФОРМИРОВАНИИ ПРОФЕССИОНАЛЬНОЙ КОМПЕТЕНТНОСТИ В ПРОГРАММИРОВАНИИ Н.Ф. Гусарова, Г.О. Котелкова, В.А. Сеницын, Ф.А. Смирнов

Рассматривается преодоление системной неопределенности среды как компоненты профессиональной компетентности программистов (ПКП). Представлены математическая модель и демонстрационная реализация системы поддержки формирования ПКП с использованием визуализации результатов программирования. Разработка сочетает погружение в реальную среду, характерное для метода проектов, с возможностью двухуровневого контроля процесса получения обучающимися профессиональных навыков.

Ключевые слова: системная неопределенность технологической среды программирования, профессиональная компетентность программистов (ПКП), координация управляющих воздействий.

Введение

Качество подготовки программистов и, соответственно, их конкурентоспособность на рынке труда в значительной и все возрастающей степени определяются не столько академическими знаниями, сколько умением решать профессиональные задачи. Как показывает опыт общения со студентами, они в большей степени мотивированы именно на приобретение навыков реального программирования при решении прикладных задач. В то же время структура учебного процесса в техническом вузе ориентирована, в первую очередь, на передачу студентам академических знаний.

Актуальность и комплексный характер проблемы инспирируют множество подходов к ее решению. Так, исследования, проводимые в течение нескольких лет в СПбГУ ИТМО [1, 2], позволили раскрыть социально-психологические (в том числе де-

мографические и мотивационные) и организационно-методические аспекты проблематики. В настоящей статье обращается внимание на те аспекты обозначенной проблемы, которые связаны с системными характеристиками технологической среды программирования. Акцентируется роль опыта преодоления системной неопределенности среды как компонента ПКП. Предлагаются математическая модель и технологическая реализация поддержки формирования ПКП с использованием визуализации результатов программирования.

Системная сложность технологической среды программирования и профессиональная компетентность как средство ее парирования

Технологическая среда для решения прикладных задач программирования (ТСП) формируется, как правило, совокупностью проблемно-ориентированных программных продуктов (ПП) и доступна актору (программисту или пользователю) через пользовательский интерфейс. На формальном уровне [3, 4] интерфейс ПП определяется как тройка

$$IF = (X, Y, S), \quad (1)$$

где X – множество стимулов, Y – множество реакций, S – множество состояний ПП, а взаимодействие актора с ПП с интерфейсом (X, Y, S) – как интерфейсная операция (ИО)

$$IA = (s, x, y, s') \in S \times X \times Y \times S, \quad (2)$$

где s – пресостояние, x – стимул, y – реакция, s' – постсостояние. Согласно (2), понятие взаимодействия интерпретируется следующим образом: в пресостоянии ПП s вызывается ИО с некоторыми значениями параметров x , на что ПП возвращает выходные параметры y и переходит в постсостояние s' . Набор ИО, определяющих функциональность ПП, описывается моделью требований – абстрактным автоматом

$$MR = (S, X, Y, E), \quad (3)$$

где $E \subseteq S \times X \times Y \times S$ – множество переходов. Конкретизация (3) до уровня отдельных параметров производится через понятие спецификации. Сигнатурой ИО I называется тройка

$$Sign_I = (In_I, Out_I, St_I), \quad (4)$$

где $In_I = \{x_i \mid i = 1, \dots, in(I); in(I) \geq 0\}$ – упорядоченный набор входных параметров операции, $Out_I = \{y_i \mid i = 1, \dots, out(I); out(I) \geq 0\}$ – упорядоченный набор выходных параметров операции, $St_I = \{s_i \mid i = 1, \dots, st(I); st(I) \geq 0\}$ – набор переменных состояния, к которым обращается операция. Спецификацией $Spec_I$ ИО I с сигнатурой $Sign_I$ называется пара предикатов

$$Spec_I = (pre_I, post_I), \quad (5)$$

где pre_I – предусловие (предикат на множестве $S_I \times X_I$), $post_I$ – постусловие (предикат на множестве $S_I \times X_I \times Y_I \times S_I$), $In = \{x_i \mid i = 1, \dots, in(I)$. Спецификацией ПП называется непустой набор спецификаций его ИО

$$Spec_{ПП} = \{Spec_{I_i} \mid i = 1, \dots, k; k > 0\}. \quad (6)$$

Решая технологическую задачу P_s , актор задает последовательность интерфейсных операций $\{IA\}_s$, т.е. формирует управление, потенциально приводящее к требуемому выходному набору переменных Out_s . Если это управление не противоречит ограничениям (3), то ПП его выполняет.

Однако в реальном программировании модель (1)–(6) оказывается, как правило, излишне упрощенной, что связано с принципиально неустранимой системной сложностью ТСП [5]. В 1975 г. Ф. Брукс выделил такие факторы системной сложности:

(Ф.1) мощность множества состояний. Число возможных состояний ПП на порядки величин превышает число состояний компьютеров, которое само по себе очень велико, причем в большинстве случаев элементы ПП взаимодействуют между собой нелинейным образом, и сложность целого растет значительно быстрее, чем линейно;

(Ф.2) искусственное происхождение. В отличие от природных объектов, которые подчиняются фундаментальным инвариантам и законам, структура ПП определяется «культурной матрицей» [5] их авторов, т.е. во многом произвольна и, соответственно, неполностью предсказуема для пользователей;

(Ф.3) сложность визуального представления полной структуры ПП, что затрудняет процесс проектирования, общение между разработчиками и работу пользователей в среде ПП.

С развитием ИТ этот список расширился:

(Ф.4) компьютеры, являющиеся физической средой ТСП, включаются в локальные и/или глобальные сети и, соответственно, испытывают непрогнозируемые воздействия извне;

(Ф.5) помимо процесса, регулируемого актором, в компьютере исполняются различные резидентные программы, не контролируемые актором;

(Ф.6) при создании ПП широко используются внешние библиотеки и принцип инкапсуляции, а многие ПП являются проприетарными; в результате множество состояний St_i доступно актору лишь частично.

В результате действия этих факторов множество состояний ПП $S(1)$ становится не полностью наблюдаемым (не полностью измеримым)¹ для актора, и априорно выделить в нем полностью наблюдаемую часть St_i , которая необходима для строгого соблюдения формализмов (4)–(6), не представляется возможным. Различные аспекты организации целенаправленной деятельности в условиях неполной наблюдаемости технологической среды изучаются в рамках когнитивной психологии, искусственного интеллекта, системного анализа и смежных дисциплин (см., например, [9–12]). Применительно к проблематике статьи выделим следующие результаты:

(Р.1) целенаправленные действия в таких условиях возможны при сочетании формальных моделей, выраженных в явных знаниях², и эвристических стратегий (анalogии, ассоциации, предположения и т.п.);

(Р.2) последние формируются естественным интеллектом на основе внутренней активности и предыдущего опыта и во многом являются имплицитными, т.е. не доступными саморефлексии.

Для обозначения соответствующих способностей человека в педагогике [2, 13–15] используется ряд смежных понятий, в том числе *умения*, *профессионализм*, *компетентность*. В содержательном плане они характеризуют способности человека эффективно решать разнообразные задачи, в том числе нестандартные, в реальных ситуациях во всей области профессиональной ответственности.

Для формирования таких способностей оказываются в той или иной мере эффективными различные подходы – тренинги, наставничество, конструктивистские технологии, проектное обучение [16–21] и т.д. Многие из них находят применение в ИТ-компаниях при вводе в выполняемые проекты новых исполнителей. Так, компания Motorola/Bangalore проводит для всех вновь принятых на работу программистов профессиональный тренинг в объеме 42 рабочих дня, что считается особенно важным для погружения в сложный проект [22]. В компании Microsoft вновь принятые программисты приступают к реальной работе в проекте в день приема, однако к каждому из них прикрепляется персональный ментор, который в течение 2–10 месяцев читает все коды,

¹ Несмотря на содержательную близость, понятие *наблюдаемость* (см., например, [6]) в разных предметных областях имеет различные формальные определения. Применительно к программированию, согласно [7], оно эквивалентно понятию *измеримость* [8]: под измеримым понимается [8] такое состояние, которое экспериментатор может определить для каждого момента времени либо непосредственно, либо через известные значения входного и выходного сигналов, не зная начального состояния.

² Явные знания – накопленный опыт, который может быть выделен и представлен в форме методик, инструкций, руководств, рекомендаций к действию. Явные знания являются сформулированными, зафиксированными [13].

написанные новичком, дает профессиональные советы и т.д. [22]. Однако такие подходы ориентированы не на повышение исходного уровня ПКП, а на адаптацию новичка к принятым в конкретном проекте методам программирования, стилю кода и т.п. Кроме того, несмотря на свою потенциально высокую эффективность, такие подходы требуют больших ресурсов (в частности, большого штата высококвалифицированных менторов), а также предполагают возможность отложенного принятия решения о найме. В более мелких же компаниях процедуры ввода в проект новых исполнителей во многом формируются спонтанно, на уровне самоорганизации и, как правило, не имеют организационной и методической поддержки.

Таким образом, для формирования ПКП необходимо получение не только академических знаний и опыта работы в команде, но и профессиональных навыков³, а также опыта преодоления системной неопределенности ТСП, причем достижение этих целей должно быть верифицируемым. Сформулируем требования к системе поддержки ПКП (далее – системе) в свете этих задач.

(Т.1) Индивидуальный вклад обучающегося в работу над заданием должен отслеживаться. Как показывает наша практика организации проектной работы при обучении в области ИТ, нередки ситуации, когда при коллективной работе основной объем выполняют 1–2 талантливых студента, а остальные члены группы пользуются их результатами; при этом переломить ситуацию мотивационными и/или организационными мерами не удается.

(Т.2) Необходимо обеспечить, по возможности, объективный контроль уровня и содержания получаемых обучающимися профессиональных навыков.

(Т.3) Последовательность заданий должна поддерживать постепенное нарастание алгоритмической сложности и, одновременно, уровня неопределенности ТСП.

(Т.4) В каждом задании обучающийся должен видеть пользу от применения конечного результата в своей реальной практике (на уровне повторно используемого кода).

(Т.5) Формирование базовых уровней профессиональной компетентности должно происходить на стороне обучающей организации, а не ИТ-компании. В то же время, учитывая специфику требований к программисту, подготовку следует строить на основе максимального соответствия промышленному процессу. В частности, целесообразно использовать при реализации заданий некоторые принципы методологии экстремального программирования [23], такие как «юнит-тестирование», «рефракторинг» и «небольшие релизы». В целом же система должна быть достаточно легко адаптируемой к различным методологиям разработки.

(Т.6) Учитывая недостаток преподавателей, способных креативно инспектировать код в условиях неполной наблюдаемости ТСП, следует максимально разгрузить преподавателя от этой задачи, используя автоматическое тестирование кода, а также интегральную (в частности, визуальную) проверку результата выполнения задания.

Моделирование системы поддержки формирования ПКП

Как показывает наша практика, удачным подходом к построению системы может служить выполнение обучающимся (актером) набора заданий, состоящих в написании и запуске расширений к существующим прикладным ПП. В частности, эффективную поддержку формирования ПКП можно организовать в процессе освоения языка программирования JavaScript [24], который обладает развитой объектной моделью и является мощным средством разработки информационных систем, предполагающих работу

³ Навыки – двигательные, сенсорные и умственные действия, доведенные до автоматизма путем многократного повторения, упражнений. Применительно к умственной деятельности выделяют интеллектуальные навыки общения (коммуникативные навыки), навыки письма, навыки презентации, управленческие навыки и т.п. [15]

через html-интерфейс, а также важнейшей частью широко используемой технологии AJAX. С точки зрения реализации сформулированных выше требований JavaScript обладает рядом преимуществ:

- полная интеграция с браузером (что требует от обучающегося учета системных неопределенностей конкретного браузера);
- поддержка большинства браузеров (в отличие, например, от технологий ActiveX и VBScript);
- сравнительно низкий «порог вхождения» обучающегося.

Таким образом, выбор JavaScript позволяет удовлетворить требование Т.4, а использование платформы Windows с весьма сложной структурой потока визуализации [25] предоставляет широкие возможности для удовлетворения требования Т.3.

Диаграмма информационных потоков в системе представлена на рис. 1. Набор заданий в системе ранжирован по уровням сложности в соответствии с требованием Т.3 (на рис. 1 n – номер уровня). После регистрации в системе обучающийся получает задание, соответствующее ранее пройденным уровням; в частности, при $n=0$ он проходит входное тестирование. Получив задание, обучающийся пишет код и запускает его на исполнение в системе. Для самоконтроля и проверки работоспособности кода реализована визуальная проверка результата исполнения кода, а также автоматическое тестирование кода (тем самым выполняются требования Т.2 и Т.5). Если обе проверки пройдены, обучающийся получает задание следующего уровня и повторяет цикл «кодирование – визуальная самопроверка – автоматическое тестирование» требуемое количество раз. На старших уровнях сложности в цикл включается проверка задания преподавателем, который указывает в комментариях свои пожелания – например, внести изменения в код, провести дополнительный рефакторинг или отладку (тем самым удовлетворяется требование Т.6).

В качестве методологической основы поддержки формирования ПКП используется концепция координации в иерархических многоуровневых системах [26], развиваемая применительно к сфере ИТ в работах [27, 28], в соответствии с которой построена модель управления в системе (рис. 2). Обучающийся C_i , получив задание P_n , пишет код, т.е. формирует управление $m_{зад}$, потенциально приводящее к требуемому результату y , и запускает его в ТСП, т.е. производит целенаправленное изменение $u_{зад}$ потока параметров ТСП u . Очевидно, что задание P_n может быть различными способами декомпозировано на отдельные подзадачи (рис. 3). Подзадачи (а также выполненное задание с ТСП в целом) взаимодействуют посредством ИО (см. модель (1)–(6)) и/или их комбинаций, которые в [26] обозначаются термином *связующие сигналы*. В соответствии с [26], качество выполнения задания описывается оценочной функцией

$$G: X \times Y \rightarrow V_{\text{глоб}}, \quad (7)$$

причем в общем случае в оценочном объекте $V_{\text{глоб}}$ присутствуют компоненты, допускающие полностью формальное описание (V_{Metr}) и слабую формализацию (V_{NMet}):

$$V_{\text{глоб}} = V_{\text{Metr}} \cup V_{\text{NMet}}. \quad (8)$$

Соответственно, результаты выполнения задания доступны акторам системы посредством сигналов обратной связи z_{MetrUT} , z_{MetrSt} , z_{NMet} , причем функциональная связь между z_{NMet} и V_{NMet} отсутствует. Компоненты V_{Metr} оцениваются автоматически (например, посредством юнит-тестирования в блоке *UnitTest*). К компонентам V_{NMet} относятся, в частности, ряд показателей поддерживаемого обучающимся стиля программирования (объем кода, использованный для решения задачи; наличие повторяющихся фрагментов кода, не выделенных в функции; количество операторов ветвления и т.д.), а также интегральные (общекультурные – см. фактор Ф.2) показатели реализованного расширения. Оценка неформализуемых показателей стиля программирования производится преподавателем (блок *ChStyle*), а интегральная оценка качества решения выполняется акторами системы за счет визуализации части потока $u_{\text{виз}}$ (блок *Viz*).

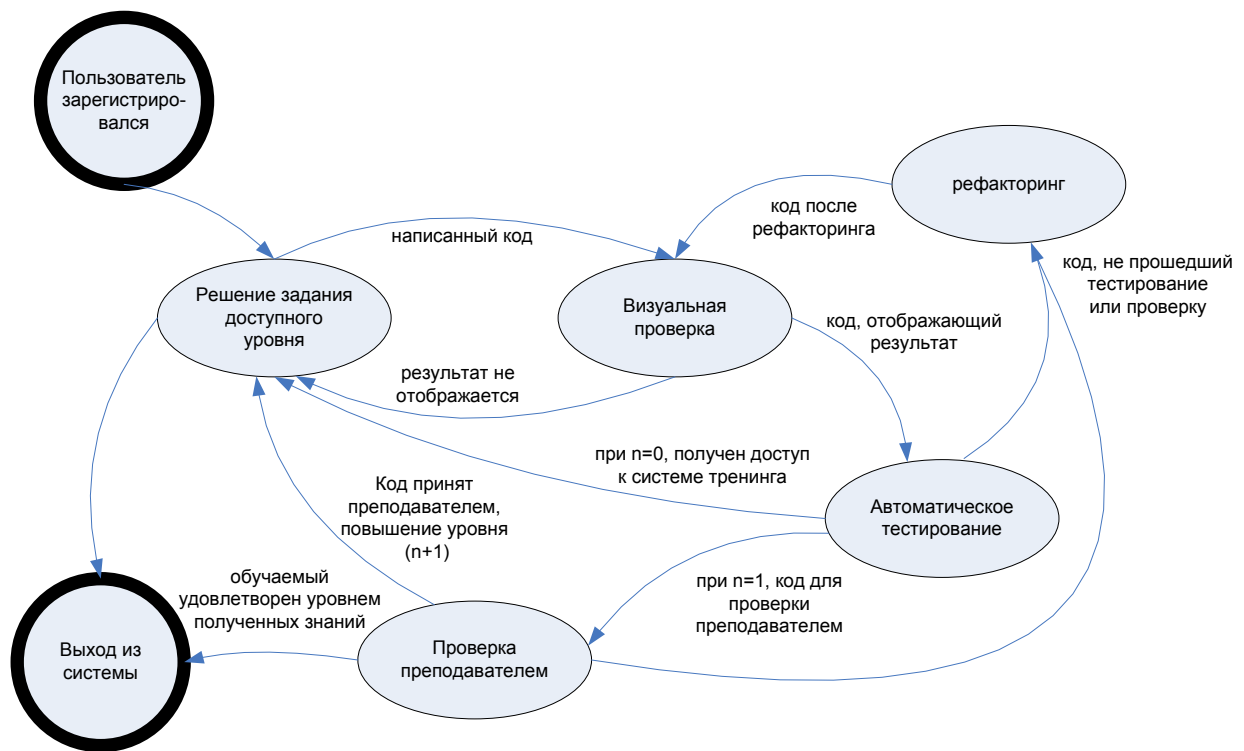


Рис. 1. DFD-диаграмма информационных потоков в системе

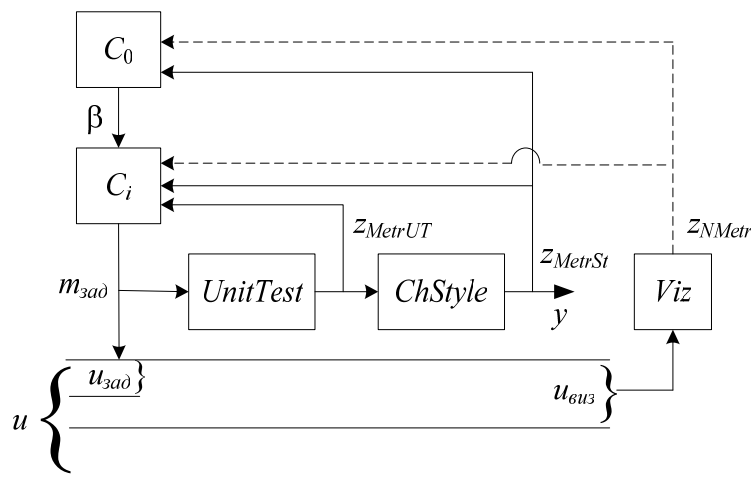


Рис. 2. Модель управления в системе

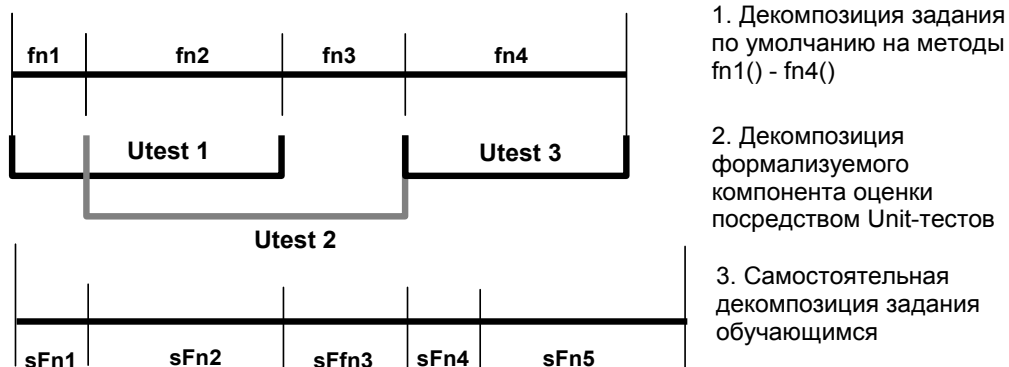


Рис. 3. Примеры декомпозиции выполняемого задания

В силу неполной наблюдаемости множества состояний ТСП, а также наличия слабо формализуемых показателей качества решения V_{NMetr} задание P_n , очевидно, может ставиться только как поиск удовлетворительных решений: для допустимого подмножества управляющих воздействий $M^f \subseteq M$ найти управляющее воздействие $\mu \in M$ такое, что

$$g(\mu, \omega) \in V_{\text{глоб}} \quad (9)$$

для всех возмущений $\omega \in \Omega$, при этом допустимое подмножество управляющих воздействий $M^f \subseteq M$ задается преподавателем через допустимые диапазоны связующих входов и параметров локальных функций качества $U_i^\gamma \times B_i^\gamma$. Другими словами, преподаватель C_0 воздействует на работу обучающегося C_i , параметризуя либо задаваемую целевую функцию G_n решаемой задачи, либо множество связующих входов для задания P_n и/или его подзадач, либо то и другое вместе, с помощью координирующих сигналов

$$\gamma = (\alpha, \beta); \quad \Gamma = A \times B, \quad (10)$$

где α – вектор допустимых значений связующих входов задания, $\alpha \subseteq A$, а β – вектор параметров, конкретизирующих компоненты целевой функции G_n , $\beta \subseteq B$.

Содержательный анализ показывает, что V_{Metr} имеет не более чем кодовую метрику, т.е. определяется над конечным полем, в то время как по крайней мере некоторые компоненты V_{NMetr} должны определяться над бесконечным полем [29]. В связи с этим поиск решения, удовлетворяющего условию (10), не может быть организован в итеративной форме, и его необходимо вести в лексикографическом порядке, а именно: вначале найти $\mu_1 \in M$ такое, что

$$g_1(\mu_1, \omega) \in V_{Metr}, \quad (11)$$

а затем найти $\mu_2 \in M$ такое, что

$$g_2(\mu_1, \omega) \in (V_{Metr} \cup V_{NMetr}). \quad (12)$$

Таким образом, в рамках выполнения (9) обучающийся, выполняя задание в ограничениях (10), самостоятельно и во многом имплицитно (см. результат Р.2) строит матрицу технологических связей (связующих сигналов u_i) между заданием (совокупностью подзадач) и ТСП. При этом сочетание неполной наблюдаемости ТСП и поисковой активности (см. Р.1) актора приводит к следующим ограничениям.

(О.1) При запуске написанного расширения в реальном ТСП происходит проверка равенства сечений полного потока ИО,

$$\{u_{-}\} = \{u_{+}\}, \quad (13)$$

в форме компарирования на программно-аппаратном уровне (здесь $\{u_{-}\}$ и $\{u_{+}\}$ – поток параметров ТСП перед запуском и после него соответственно).

(О.2) Результаты проверки (13) могут быть интерпретированы при визуализации, т.е.

$$\{u_{\text{зад}}\} \subseteq \{u_{\text{виз}}\}. \quad (14)$$

(О.3) Результирующий вектор задания y должен соответствовать требованиям, задаваемым метризуемым компонентом оценочного объекта V_{Metr} , однако их трансляция на поток параметров ТСП

$$y_{\text{доп}} \rightarrow u_y^{\text{доп}} \quad (15)$$

(т.е. задание вектора α) возможна только на уровне

$$u_y^{\text{доп}} \supseteq \{\tilde{u}\}, \quad (16)$$

где $\{\tilde{u}\}$ – наблюдаемый (измеримый) по выходу подпоток полного потока u . Другими словами, требования к потоку параметров, реализуемых при выполнении задания P_n , как правило, могут быть заданы не в виде полного шаблона, а только в виде набора ограничений.

При ограничениях О.1–О.3 можно доказать следующие утверждения.

У.1. Если в системе управление организовано в соответствии с рис. 2 и выполняются ограничения О.1–О.3, то для удовлетворения требований к качеству ее работы в фор-

ме (11) необходимо, чтобы при запуске расширения допустимый набор подмножества параметров $u_y^{\text{доп}}$ принадлежал текущему сечению полного потока параметров $\{u\}$:

$$u_y^{\text{доп}} \supseteq \{u\}. \quad (17)$$

У.2. Если для системы справедливо утверждение У.1, а модификации функций качества, задаваемые вектором β , монотонно связаны с соответствующими сигналами обратной связи $\{z\}$, то возможно удовлетворение требований к качеству работы системы в форме (12).

Очевидно, что для поддержки сходимости ТСП в потоке ИО $\{u\}$ не должны содержаться комбинации, противоречащие требованиям (3). Нарушение этих требований даже в части потока ИО, не входящей в $\{u_{\text{виз}}\}$, проявляется в ошибках переполнения и, соответственно, нарушениях штатного поведения ТСП. Формирование индивидуального банка таких ситуаций и сценариев поведения в них, как показывает практика, является не менее важным аспектом ПКП актора, чем опыт его работы в штатных ситуациях.

Пример реализации системы поддержки формирования ПКП

Пример реализации разработанной системы представлен на рис. 4–5. Обучающиеся имеют доступ к заданиям своего уровня, рейтингу, учебно-справочной литературе и форуму. Каждое задание состоит из учебного материала, непосредственно задания, набора тестов, рекомендаций по тестированию и рекомендаций по установке расширения в основную программу. Задания ранжируются по уровню сложности в соответствии с требованиями утверждения У.2. Некоторые примеры заданий младших уровней доступны для загрузки [30].

На странице задания (рис. 4) отображаются список заданий и результаты решений обучающегося, которые можно просмотреть, исправить, сохранить, протестировать и послать преподавателю на проверку. Напротив задания отображается статус проверки (принято, в очереди на проверку, не принято) и комментарий преподавателя. На странице решения (рис. 5) для удобства обучающихся встроен редактор кода Editarea 0.7.2.3 с номерами строк и подсветкой кода, а также консоль JavaScript для вывода промежуточных результатов и лога ошибок.

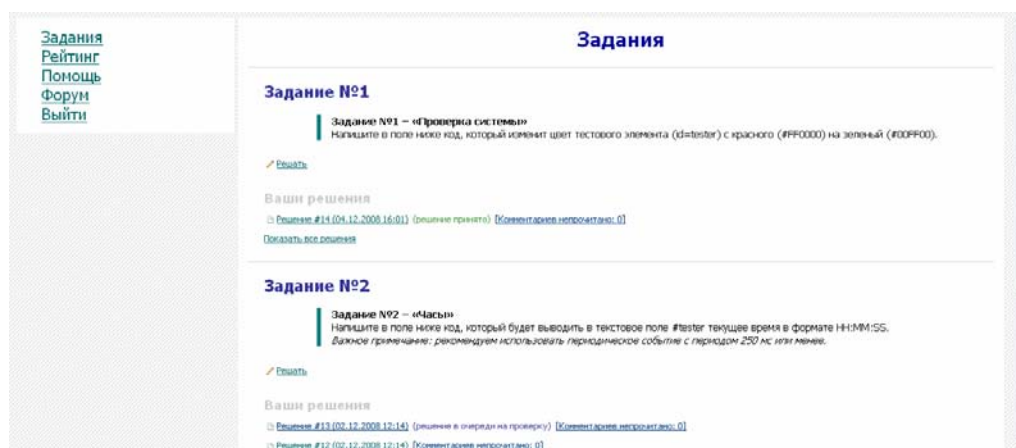


Рис. 4. Интерфейс обучающегося

В настоящее время разрабатывается пакет заданий в форме создания востребованных в реальной практике расширений к open-source веб-браузеру Mozilla Fire Fox, что позволят сформировать постепенный сдвиг локализации сложности и неопределенности заданий – от выполнения задачи в ТСП к парированию неопределенности ТСП.

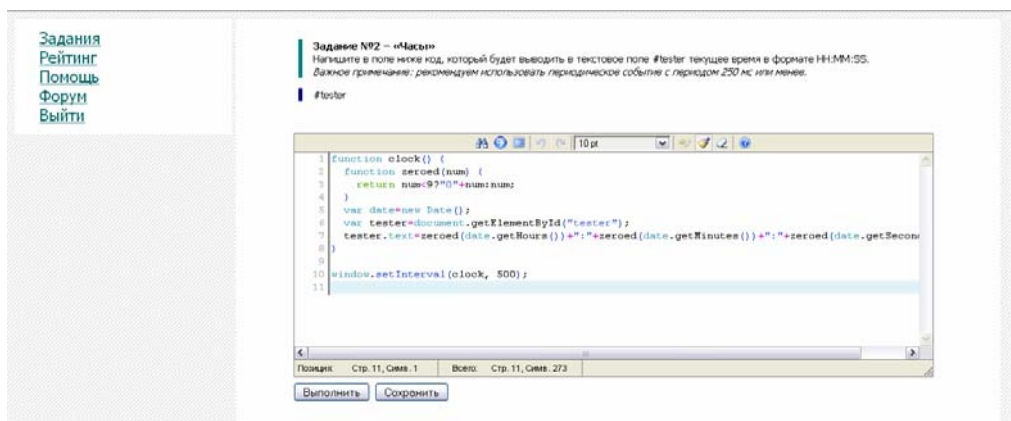


Рис. 5. Визуальный редактор кода

Заключение

В статье акцентируется роль опыта преодоления системной неопределенности среды как компонента профессиональной компетентности программистов (ПКП). Предлагаются математическая модель и технологическая реализация системы поддержки формирования ПКП с использованием визуализации результатов программирования. Предложенный подход сочетает в себе погружение в реальную среду, характерное для метода проектов, с возможностью объективного контроля и самоконтроля уровня и содержания получаемых обучающимся профессиональных навыков, а также разгружает преподавателя от рутинных проверок кода. В качестве перспектив развития концепции планируется дальнейшая разработка программных средств (с особым вниманием к многопользовательскому режиму) и создание учебно-методического обеспечения для внедрения системы в образовательный процесс.

Литература

1. Девятая Всероссийская олимпиада школьников по информатике и программированию / Под ред. В.Н. Васильева, В.Г. Парфенова, А.С. Станкевича. – СПб: СПбГУ ИТМО, 2008. – 244 с.
2. Лисицына Л.С. Концепция и методология управления разработкой образовательного процесса по подготовке компетентных выпускников средствами сетевой информационной системы: Автореферат дис. ... докт. техн. наук. – СПб: СПбГУ ИТМО, 2008.
3. Кулямин В.В., Петренко А.К., Косачев А.С., Бурдонов И.Б. Подход UniTesK к разработке тестов // Программирование. – 2003. – Т. 29. – № 6. – С. 25–43.
4. Хорошилов А.В. Спецификация и тестирование компонентов с асинхронным интерфейсом: Автореферат дис. ... канд. физ.-мат. наук. – М.: ИСП РАН, 2006.
5. Брукс Ф. Мифический человекомесяц или Как создаются программные системы. – М.: Символ-Плюс, 2006. – 304 с.
6. Kalman R.E. On the general theory of control systems // Proc. First Intern. Congr. Automatic Control. – Moscow, 1960. – Vol. 1. – London: Butter Worth & Co.
7. Шалыто А. А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. – СПб: Наука, 1998. – 628 с.
8. Заде Л., Дезоер Ч. Теория линейных систем: Метод пространства состояний. – М.: Наука, 1970.
9. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб: Питер, 2001. – 384 с.
10. Поспелов Д.А. Ситуационное управление: теория и практика. – М.: Наука, 1986. – 288 с.
11. Люгер Дж.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. 4-е издание: Пер. с англ. – М.: Издательский дом «Вильямс», 2003.

12. Джонс Дж.К. Методы проектирования: Пер. с англ. 2-е изд. – М.: Мир, 1986. – 326 с.
13. ГОСТ Р ИСО 10015, Государственный основной стандарт: Менеджмент организации. Руководящие указания по обучению, разработанный на основе международного стандарта: ISO 10015:1999 Quality management – Guidelines for training (IDT).
14. Дружилов С.А. Психология профессионализма субъекта труда: интегративный подход // Ежегодник Российского психологического общества: Материалы 3-го Всероссийского съезда психологов. – СПб: Изд-во СПбГУ, 2003. – Том 3. – С.153–157.
15. Пидкасистый П.И., Фридман Л.М., Гарунов М.Г. Психолого-дидактический справочник преподавателя высшей школы. – М.: Педагогическое общество России, 1999.
16. Пахальян В.Э. Групповой психологический тренинг. – СПб: Питер, 2006.
17. Коммуникативно-ориентированные образовательные среды. Психология проектирования / Под ред. В.В. Рубцова. М., 1996. – Режим доступа: http://ido.rudn.ru/psychology/psychology_of_marketing/ch11_1.html, своб.
18. Пакет Moodle2.0. – Режим доступа: docs.moodle.org, своб.
19. Kilpatrick W. H. The project method. – N.Y., Columbia University Teachers College, 1918.
20. Новые педагогические и информационные технологии в системе образования / Под ред. Е.С.Полат – М., 2000.
21. Лаврентьев М.М., Алсынбаева Л.Г., Васючкова Т.С. Метод проектов в системе непрерывного обучения информатике. Опыт НГУ.
22. Иордан Э. Человеческий фактор в разработке программного обеспечения: привлечение, мотивация и удержание ключевых сотрудников / Материалы семинара. – Режим доступа: <http://www.yourdon.com>, своб.
23. Бек К., Фаулер М. Экстремальное программирование. – СПб: Питер, 2003.
24. Дмитриева М.В. JavaScript. Занятия 1–7 // Компьютерные инструменты в образовании. – 2002. – № 1–5; 2003. – № 1, 2.
25. Таненбаум Э. Операционные системы. – СПб: Питер, 2005. – 1038 с.
26. Месарович М., Мако Д., Такахара И. Теория иерархических многоуровневых систем. – М.: Мир, 1973. – 344 с.
27. Гусарова Н.Ф., Маятин А.В., Смирнов Ф.А. Обратные задачи в компьютеризированных технологических средах // Научно-технический вестник СПбГУ ИТМО. – 2007. – Вып. 44. – С. 284–292.
28. Гусарова Н.Ф. Координация в технологических процессах со слабо формализуемыми критериями. – СПб: СПбГУ ИТМО, 2001. – 271 с.
29. Дударенко Н.А., Слита О.В., Ушаков А.В. Математические основы современной теории управления: аппарат метода пространства состояний: Учебное пособие / Под ред. А.В. Ушакова. – СПб: СПбГУ ИТМО, 2008. – 310 с.
30. Тестовое учебное задание. – Режим доступа: <http://212.116.110.171:8081/share/devln/training.prj/files/task.smpl.zip>, своб.

Гусарова Наталия Федоровна

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, pat-fed@list.ru

Котелкова Галина Олеговна

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, glaya@inbox.ru

Синицын Владислав Андреевич

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, shinisaru@gmail.com

Смирнов Филипп Александрович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, smirnfil@gmail.com

PROGRAM VERIFICATION BASED ON ALGEBRAIC AND TOPOLOGICAL APPARATUS OF COUNTABLE CUBIC COVERINGS

Dmitry Kravchenko (semem@yandex.ru), Leonid Osovetskiy (director@cit.ifmo.ru)

Keywords: program verification, cubic coverings, dead code.

Program verification method based on cubic coverings is described in the article. This method gives an opportunity to prove formally programs compatibility and to make a conclusion about the absence of undeclared facilities and dead codes in programs.

FUNCTIONAL MODELING METHOD OF INFORMATION TECHNOLOGIES SAFETY EVALUATION BY "COMMON CRITERIA"

Oleg Zaytsev (zoe83@mail.ru), Alexander Lyubimov (a_v_l@inbox.ru)

Keywords: model, method, modeling, functional

For "Common Criteria" functional model building the SADT method and its DFD notation are used. SADT method initially is not able to reflect "Common Criteria" model properly due to modeling object certain specificity, that's why for CC functional modeling method the SADT method should be somehow corrected.

QUANTITATIVE AND QUALITATIVE ANALYSIS OF CHARACTERISTICS INFLUENCING THE DISTRIBUTED REPOSITORIES FUNCTIONING

Vladimir Kirillov (kirillovvv@gmail.com), Nikolay Luk'yanov (nikolav@lukianov.net)

Keywords: repository, distributed, characteristics, Internet, service

This article represents the result of comprehensive investigations of the distributed data storehouses and the quantitative and qualitative analysis of their functioning characteristics. As the result of research the author describes the functional model of distributed data storehouse system for Internet services.

THE SOURCE CODE PARSING METHOD ON BASIS OF LANGUAGE SIMPLIFIED GRAMMAR

**Dmitry Aref'ev (av_dmitry@mail.ru), Vladimir Vereshchagin (seckey@yandex.ru),
Alexey Galanov (daiverddd@yandex.ru)**

Keywords: security, information, vulnerability, analysis, grammar, source code

The article involves the source code parsing method on basis of simplified ensemble of grammar production rules. The authors present methodological conception and information about effective practical application.

THE COMPUTER-BASED SYSTEM OF COLLECTION AND PROCESSING OF BOREHOLE MEASUREMENTS DATA USING AN ACOUSTIC FLOWMETER

Tat'yana Voytyuk (voytuk@i-teco.ru), Anatoliy Dyomin (dav_60@mail.ru)

Keywords: oil, mining, field, acoustic flowmeter, fiberoptic.

The article describes new concepts of collection and processing of borehole measurements data using an acoustic flowmeter. Computerized system allows to increase the development profitability of the earlier unpromising oil fields.

COOK-TORRANCE LIGHTING MODEL IMPLEMENTATION USING DEFERRED SHADING

Alexey Bezgodov (demiurgh@gmail.com), Eduard Starodubcev (sevedw@d1.ifmo.ru)

Keywords: computer graphics, lighting model, deferred shading

We consider features of Deferred Shading, a number of advantages and disadvantages of Cook-Torrance lighting model, and implementation of this lighting model using deferred shading. We suggest a layout for g-buffer and number of optimization techniques.

HARDWARE-SOFTWARE COMPLEX FOR AIRCRAFT LANDING CONTROL

Alexander Sauta (demosito@gmail.com), Vladimir Pol'akov (v_i_polyakov@mail.ru)

Keywords: aircraft landing, satellite navigation system, satellite landing system.

The article gives a brief description of satellite landing system and satellite navigation system differential operating mode. The problem of aircraft landing accuracy control for existing systems is considered, and its solution with tools commercially available on avionics market is suggested.

IMAGE SEGMENTATION FOR RECOGNITION OF PRINTED DOCUMENTS

**Anatoly Korobeynikov(korobeynikov_a_g@mail.ru), Hoang Giang
(giangdebug@yahoo.com)**

Keywords: conference, image recognition, image segmentation

This paper describes the newly developed multiscale algorithm for document segmentation, based on the Dynamic Local Connectivity Map (DLCM). Using this algorithm we can extract the hierarchical tree of document blocks with a complicated structure (presence of pictures, sports, text in different fonts and so on)

EXTERNAL ATTACKS EFFECT ON DC COEFFICIENTS OF DCT MATRIX OF STILL SEMITONE IMAGES

**Nikolay Prokhozhev (karaja2@yandex.ru), Olga Mikhailichenko (19791109@list.ru),
Anatoliy Korobeynikov (korobeynikov_a_g@mail.ru)**

Keywords: steganography, DCT-base algorithms, JPEG compression

The subject of this article is an external attacks effect on DC coefficient of semitone image DCT matrix. The goal of this analysis is a convenience determination of using DC coefficients as a data hide area for steganography algorithms. Resizing, Gaussian noise, filtering and JPEG compression are used as an external attacks.

METHOD OF PROCESS DYNAMICS RESEARCH BY QUANTITIVE RECURRENT CALCULATION IN THE WINDOW MOVING ALONG THE MAIN DIAGONAL OF RECURRENT DIAGRAM

Vladislav Kiselyov (contact@impsoft.spb.ru), Boris Krylov ((krylov@mail.ifmo.ru)

Keywords: recurrent analysis, recurrent diagrams, window size.

The article deals with a quantitive recurrent calculation of recurrent diagram in the window moving along the main diagonal. Examples of the method application to model and natural data are given. The results and application facilities are estimated.

STUDY OF COMMUNICATION NETWORKS WITH A GLANCE OF PACKETS LIMITED LENGTH

Taufik Aliev (alive@d1.ifmo.ru), Nguyen Duc Tai (duc_tai@mail.ru)

Keywords: simulation, communication network, packets length, distribution, normalization, Ethernet.

A study of the limited packets length impact on the functioning characteristics of communication networks was carried out using simulation system GPSS. We show that the error of calculation characteristics results in the assumption of unlimited length of packets can reach up to tens of percent.

BUFFER MEMORY SIZE ESTIMATION IN THE TELECOMMUNICATION NETWORK INTERMEDIATE NODES

Taufik Aliev (aliev@d1.ifmo.ru), Vladimir Sosnin (vsosnin@mail.ru)

Keywords: telecommunication network node, buffer memory, upper bound, simulation

The impracticability of analytical methods for buffer memory size estimation in telecommunication network intermediate nodes is shown. The results of multiple simulation experiments are allowed suggesting an empiric inequality that can be used to calculate accurately the upper bound for the buffer memory size. The inequality gives decent results only if utilization factor is above 40%.

OPTIMIZATION ALGORITHMS FOR THE WIRELESS SENSOR NETWORK BASED ON ZIGBEE PROTOCOL

Sergey Trifonov (serxa@mail.ru), Timofei Istomin (tim.ist@gmail.com),

Alexei Chechendaev (a_chechendaev@ipmce.ru),

Maxim Minenko (mminenko2006@yandex.ru),

Yaroslav Kholodov (kholodov@crec.mipt.ru)

Keywords: wireless networks, ZigBee, energy conservation

The wireless networks based on ZigBee protocol and problems of reducing both message delivery time and energy consumption at the same time are considered. The roles of distribution and schedule planning algorithms for routers are described with their theoretical analysis. The results of carried out experiments are given and analyzed.

THE ELECTRONIC COURSE PROGRAMMING ON THE BASIS OF THE AUTOMATA APPROACH

Andrey Lyamin (lyamin@cde.ifmo.ru), Maria Pleshkova (maria@cde.ifmo.ru)

Keywords: electronic course, finite automata theory, transition diagram.

This article covers the description of an electronic course in the AcademicNT distant learning system based on competence set-rating approach. The graphic notation for the description of electronic course models is developed on the basis of the automatic approach. Condition components are defined. Typical transition diagrams are constructed, and the dependence analysis of condition number from electronic course modules quantity and access technique is carried out. Modeling technique of the course logic in the Stateflow environment of Matlab system is offered.

DESIGN PROCEDURE OF COMPUTATIONAL CAPABILITY OF THE HARDWARE-SOFTWARE COMPLEX FOR CARRYING OUT UGE IN THE COMPUTER FORM

Andrey Lyamin (lyamin@cde.ifmo.ru), Anton Skshidleviskiy (anton@cde.ifmo.ru)

Keywords: Simulation, technique, algorithms, hardware-software complexes.

The article is devoted to big distributed hardware-software complexes design and arrangement, an optimal choice of the equipment configuration. The developed technique and algorithms allow defining a number of the complex hardware characteristics for carrying out UGE in the computer form based on the users quantity in system.

CONTROL ACTIONS COORDINATION WHILE PROFESSIONAL QUALIFICATIONS FORMATION IN PROGRAMMING

Natalia Gusarova (natfed@list.ru), Galina Kotelkova (glaya@inbox.ru), Vladislav

Sinitsyn (shinizaru@gmail.com), Filipp Smirnov (smirnfil@gmail.com)

Keywords: system uncertainty of technological programming environment, programmer professional qualifications, control actions coordination

The article deals with overcoming of environment system uncertainty as a part of programmer professional qualifications (PPQ). Mathematical model and demonstration system of PPQ formation are represented with visualization of programming results. This system implies immersion into the real-life environment with the two-level control facility while receiving professional skills by learners.

1. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ	3
Зайцев О.Е., Любимов А.В. Методика функционального моделирования процесса оценки защищенности информационных технологий по «Общим критериям»	3
Лукьянов Н.М., Кириллов В.В. Анализ факторов, влияющих на качественные и количественные показатели функционирования систем распределенного хранилища данных	9
Арефьев Д.Б., Верещагин В.Л., Галанов А.И., Молдовян А.А. Метод разбора исходных текстов по упрощенной грамматике языка	17
Войтюк Т.Е., Демин А.В. Автоматизированная система сбора и обработки данных скважинных измерений с помощью акустического расходомера	25
Безгодов А.А., Стародубцев Э.В. Реализация модели освещения Кука-Торренса с использованием технологии Deferred Shading	34
Саута А.О., Поляков В.И. Программно-аппаратный комплекс для контроля систем посадки воздушных судов	44
Хоанг З., Коробейников А.Г. Сегментация изображения для распознавания печатных документов.....	52
Прохожев Н.Н., Михайличенко О.В., Коробейников А.Г. Влияние внешних воздействий на DC коэффициенты матриц ДКП в полутоновых изображениях	57
Киселев В.Б., Крылов Б.А. Исследование динамики процессов методом вычисления мер количественного рекуррентного анализа в окне, смещаемом вдоль главной диагонали рекуррентной диаграммы	62
2. ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ	73
Нгуен Дык Тай, Алиев Т.И. Исследование сетей передачи данных с учетом ограниченной длины пакетов	73
Соснин В.В., Алиев Т.И. Оценка емкости буферной памяти в промежуточных узлах телекоммуникационной сети	81
Трифонов С.В., Истомин Т.Е., Чечендаев А.В., Миненко М.И., Холодов Я.А. Алгоритмы оптимизации работы беспроводной сенсорной сети на базе протокола ZIGBEE	86
3. ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	96
Плешкова М.В., Лямин А.В. Программирование электронного курса на основе автоматного подхода	96
Скшидлевский А.А., Лямин А.В. Методика расчета вычислительной мощности программно-аппаратного комплекса для проведения ЕГЭ в компьютерной форме.....	103
Гусарова Н.Ф., Котелкова Г.О., Сеницын В.А., Смирнов Ф.А. Координация управляющих воздействий при формировании профессиональной компетентности в программировании	111
SUMMARY.....	121