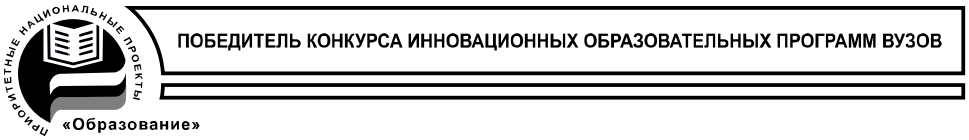
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ



НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК

Выпуск 46

ИНФОРМАЦИОННЫЕ И ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ

САНКТ-ПЕТЕРБУРГ
2008

В научно-техническом вестнике СПбГУ ИТМО, Выпуск 46 «ИНФОРМАЦИОННЫЕ И ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ» представлены работы, выполненные в рамках:

– инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» вузов России на 2007–2008 гг.;
– аналитической ведомственной целевой программы «Развитие научного потенциала высшей школы (2006–2008 гг.) (Федеральное агентство по образованию);
– Федеральной целевой программы развития образования на 2006–2010 гг. (Федеральное агентство по образованию);
– Федеральной целевой программы развития научно-технологического комплекса России на 2007–2012 гг. (Федеральное агентство по науке и инновациям);
– Российского фонда фундаментальных исследований, а также инициативные разработки.

В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

ISSN 1819-222X

1 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

АДАПТИВНАЯ АВТОМАТИЗИРОВАННАЯ СИСТЕМА СБОРА И ОТОБРАЖЕНИЯ ИНФОРМАЦИИ ДЛЯ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ

А.А. Волков (Костромской государственный университет им. Н.А. Некрасова)
Р.А. Набатов (Костромской государственный технологический университет)
О.В. Щекочихин (Костромской государственный технологический университет)
Научный руководитель – д.т.н., профессор В.Н. Шведенко
(Костромской государственный технологический университет)

Основные недостатки современных информационных систем в управлении предприятием или корпорацией заключаются в следующем:

1. высокая жесткость создаваемой информационной системы;
2. значительные финансовые ресурсы для поддержания системы в работоспособном состоянии;
3. большое время на модернизацию приложений, изменение их функциональности.

Предлагаемая технология позволяет создавать гибкие информационные системы, которые позволяют быстро перенастраивать приложения пользователя и структуру базы данных. Система строится из нескольких серверов баз данных (в частном случае система размещается на одном сервере). Объем хранимой информации на каждом сервере может достигать нескольких миллионов записей. Сервер, будучи один раз установленным, не требует постоянного обслуживания: все функции работают в автоматическом режиме, обеспечивая требуемые показатели работы системы. Также подразумевается возможность подключения к уже существующим хранилищам данных и СУБД предприятия (базы данных АСУП, системы бухгалтерского, складского учета и т.д.).

Одним из основных элементом системы является генератор информационных объектов [3]. Пользователь может создавать неограниченное количество объектов и их версий. При этом данные от версии к версии не теряются. Также имеется генератор приложений пользователей, который формирует окно пользователя для работы с коллекцией экземпляров информационного объекта [2]. Сюда же входят подсистемы подготовки отчетов и программирования вычислений.

Следующим элементом системы является модуль проектирования бизнес-процессов и формирования электронного документооборота на основе разработанных ранее информационных объектов и приложений пользователя.

Система обладает свойством расширения масштабов, т.е. увеличение пользователей и информационных объектов в системе и их экземпляров не влияет на время записи, исправления и выборки информации.

Такой инструмент дает возможность создавать информационную систему управления предприятием или корпорацией на следующих принципах.

1. Система управления строится не снизу вверх, как это принято, а сверху вниз. Здесь реализуется принцип «руководитель лучше всех знает, какой показатель ему требуется для управления тем или иным процессом» [4].
2. Показатели собираются и интегрируются из первичных документов – принцип однократного ввода информации в систему».
3. Возможно делегирование полномочий в принятии управленческих решений на нижнее звено управления [4]. На верхнем звене остаются только функции контроля над показателем – принцип индивидуальной, а не коллективной ответственности.

4. Движение показателей строго регламентируется. Возможен оперативный контроль только за теми показателями, которые выходят за установленные пределы – принцип управления по отклонению.
5. Функция ИТ-специалиста связана только с настройкой компьютеров, локальной вычислительной сети и системы телекоммуникации – принцип отчуждения программиста от создания и сопровождения информационной системы управления предприятием.

Проектирование информационной системы начинается с моделирования предметной области путем создания набора информационных объектов [3]. Под информационным объектом понимается некоторая иерархическая структура, отражающая характеристики реально существующих объектов. В системе все объекты делятся на справочники и документы.

Тип объекта однозначно определяется на этапе создания. В процессе проектирования каждый объект наделяется свойствами. В системе поддерживается работа со свойствами трех типов: строка, число, дата. Также возможно включение в структуру информационного объекта других объектов. На рис. 1 представлен внешний вид окна дизайнера объектов.

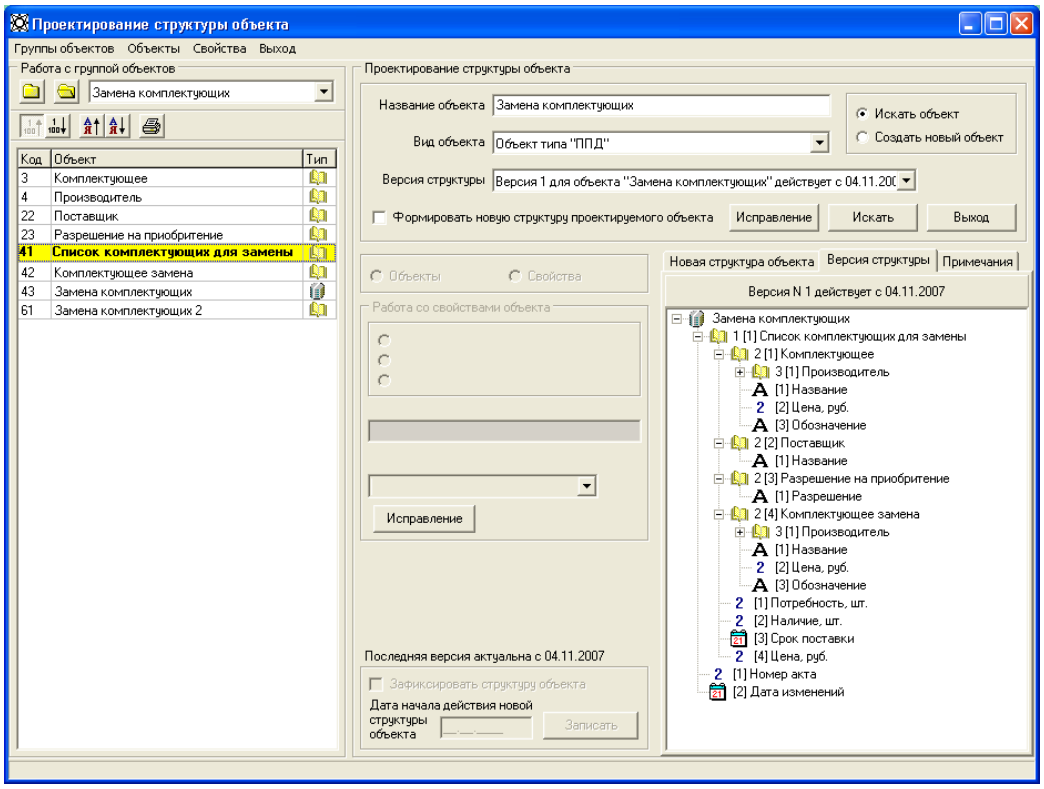


Рис. 1. Внешний вид окна дизайнера объектов

Система работы с объектами предметной области поддерживает гибкое и адекватное изменение структуры информационных объектов. Эта работа осуществляется через механизм управления версиями объектов [3]. Такой подход подразумевает фиксацию определенной версии структуры объекта на каком-либо интервале времени. Любое дальнейшее изменение структуры объекта происходит в следующем порядке: проектируется новая версия (для которой за основу может быть взята одна из предыдущих версий), фиксируется на определенную дату, при этом предыдущая версия сохраняется в истории системы. С момента наступления указанной даты актуальной становится последняя версия объекта [3].

Кроме того, система поддерживает возможность проектирования предметной области по принципу «сверху вниз» [2]. Это реализовано посредством механизма каскадного проектирования объектов, которое заключается в том, что при проектировании объекта самого верхнего уровня нет необходимости заранее проектировать объекты,

которые должны быть включены в его структуру. Это можно сделать на этапе проектирования головного объекта путем создания новых объектов, включения их в структуру головного и проектирования их версий. Эта возможность избавляет от необходимости заранее создавать объекты более низкого порядка.

Проектирование экранных форм является вторым этапом в подготовке информационной системы. На рис. 2 представлен внешний вид окна дизайнера форм.

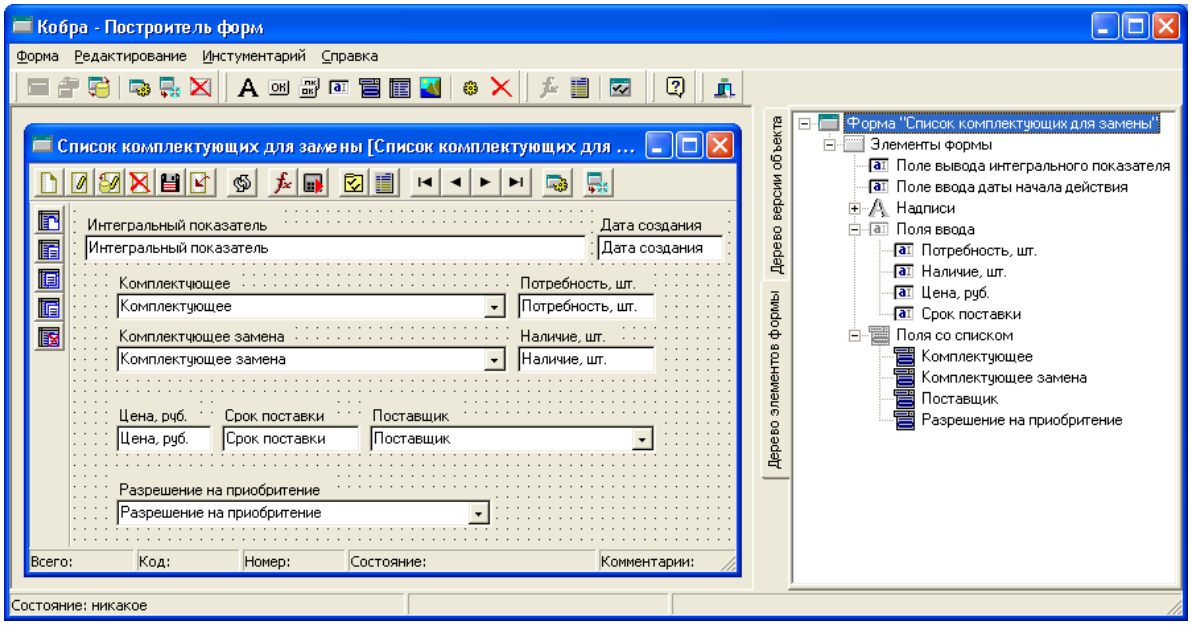


Рис. 2. Внешний вид окна дизайнера форм

Данный процесс полностью опирается на предыдущий, поскольку каждая форма строится на основе одной из версий какого-либо объекта. Набор элементов формы состоит из элементов отображения данных, среди которых выделяются поля ввода, поля со списком, а также таблицы; управляющих кнопок, реализующих необходимый набор методов для работы с формой; стандартной панели управления.

Для каждого элемента формы предусмотрен необходимый перечень настроек, который зависит от характера элемента и доступен через соответствующее диалоговое окно. Кроме того, возможны различные варианты настройки самой формы, среди которых можно выделить параметры загрузки данных, доступные методы работы с данными, а также параметры изменения цветовых характеристик формы в зависимости от режимов работы [1]. Предусмотрена возможность создания отчетов для вывода данных, загруженных в форму на печать, а также возможность программирования формул для вычисления значений свойств.

Следующий шаг в подготовке системы – моделирование бизнес-процессов. На первом этапе подготовки шаблона бизнес-процесса необходимо определить центры ответственности на каждом шаге прохождения бизнес-процесса и создать набор ролей. Роль определяет уровень доступа и режим работы с информацией во время работы бизнес-процесса [4].

Любой бизнес-процесс сопровождается набором документов. Проектирование бизнес-процесса заключается в определении регламента движения этих документов от одного центра ответственности к другому. На рис. 3 представлен внешний вид модуля проектировки бизнес-процессов. Каждый документ в системе обладает набором реквизитов:

- статус – определяет режим работы с документом и порядок исполнения;
- инициатор – роль, разработчик документа или автор исходных данных, используемых в документе;
- исполнитель – роль, центр ответственности, обработчик исходных данных;

- форма работы с документом – экранные формы системы Кобра;
- уровень – этап прохождения бизнес-процесса, который сопровождает документ;
- контрольное время – относительное время обработки документа исполнителем.

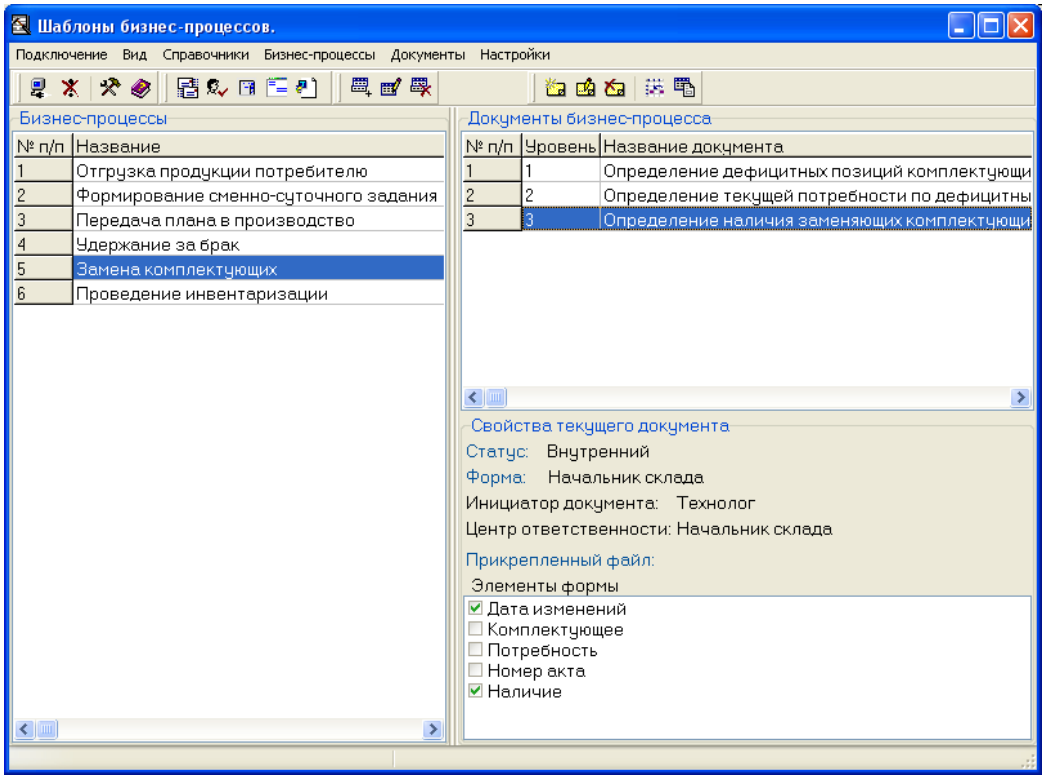


Рис. 3. Формирование регламента документа

При инициализации шаблона бизнес-процесса происходит настройка ролей для конкретных участников бизнес-процесса. Все участники бизнес-процесса работают в локальной сети предприятия. На каждой рабочей станции установлено приложение, позволяющее автоматически получать задания. Исполнитель видит активный бизнес-процесс, регламент его исполнения и соответствующую экранную форму. Каждое действие по работе с системой фиксируется в истории действий всех участников бизнес-процесса [1].

Практическая реализация описываемого программного комплекса показала возможность в несколько раз уменьшить время настройки информационной системы, сократить период адаптации системы к изменениям в системе управления предприятием, а также сократить финансовые затраты на сопровождение.

Литература

1. Когаловский М.Р. Энциклопедия технологий баз данных. – М.: Финансы и статистика, 2002. – 800 с.
2. Козловский А. Объектные СУБД: ситуация смены парадигмы // BYTE/Россия. – 2000. – №8. – С. 16–28.
3. Шведенко В.Н. Временная модель данных на основе объектно-ориентированных технологий / В.Н. Шведенко, Д.А. Иванов. – Кострома: КГТУ, 2003. – 90 с.
4. Шведенко В.Н. Объектно-функциональная система управления предприятием // Изв. вузов. Технология текстильной промышленности. – 2004. – №4. – С.104–110.

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ПРИ ПРОЕКТИРОВАНИИ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ

В.Б. Новосельский

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

В статье рассматривается жизненный цикл проектирования распределенных баз данных (РаБД), описываются этапы проектирования. На основании взаимозависимости этапов фрагментации данных, размещения данных и выбора стратегии исполнения запросов формулируется задача проектирования РаБД и производится оценка ее сложности. Автор рассматривает применимость генетических алгоритмов для решения поставленной задачи и предлагает подход, позволяющий учесть взаимозависимость этапов проектирования.

Введение

Создание распределенных информационных систем является весьма актуальной задачей. Это связано с возрастающими потребностями в приложениях, доступ к которым осуществляется из различных географических местоположений. Увеличиваются требования к оперативности и достоверности информации. Управление информацией происходит с помощью систем управления базами данных. Для достижения высокой производительности распределенных приложений, работающих с базами данных, необходимы эффективные методы проектирования распределенных баз данных (РаБД).

В статье рассматривается жизненный цикл проектирования РаБД, описываются этапы проектирования. На основании взаимозависимости этапов фрагментации данных, размещения данных и выбора стратегии исполнения запросов формулируется задача проектирования РаБД и производится оценка ее сложности. Автор рассматривает применимость генетических алгоритмов для решения поставленной задачи и предлагает подход, позволяющий учесть взаимозависимость этапов проектирования.

Жизненный цикл проектирования РаБД

Жизненный цикл проектирования РаБД состоит из двух фаз: начальное проектирование и репроектирование. Большинство исследователей под начальным проектированием понимают фрагментацию БД и размещение фрагментов по узлам вычислительной сети (ВС). С течением времени возможно ухудшение производительности приложений, работающих с РаБД, вызванное изменениями в инфраструктуре распределенной среды (ИРС). Под инфраструктурой понимаются физические и логические параметры функционирования системы, а именно транзакции и их частоты, топология ВС, характеристики узлов ВС и т.д.

- При возникновении таких изменений требуется репроектирование РаБД для сохранения производительности приложений. Репроектирование приводит к возникновению новых схем фрагментации и размещения и влечет за собой необходимость материализации обновленного дизайна. Жизненный цикл процесса проектирования изображен на

рис. 1.

Изменения ИРС могут быть двух типов: физические и логические. Возможны три сценария репроектирования РаБД: корректирующее, превентивное и адаптивное. Подробнее вопросы репроектирования и материализации рассматриваются в [1–3] и не являются целью исследования в данной работе.

Далее кратко описаны основные этапы проектирования.

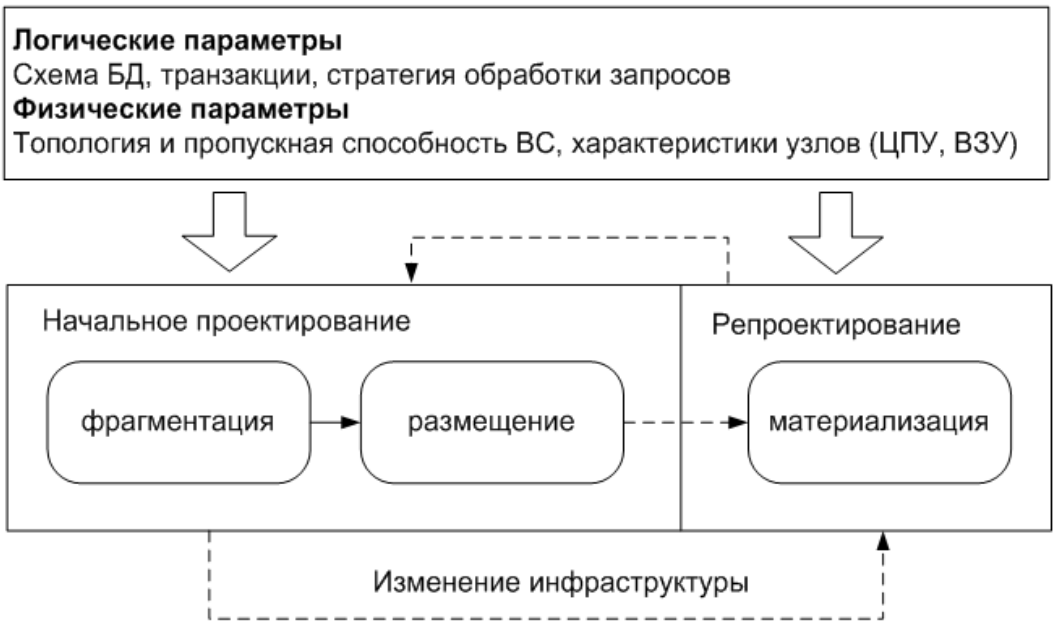


Рис. 1. Жизненный цикл проектирования РаБД

Фрагментация данных

Основной целью фрагментации является сужение пространства поиска при исполнении запроса. Фрагментация данных допускает разбиение отношения на два или более сегмента или фрагмента. Каждый фрагмент может храниться на любом узле ВС. Принято выделять две базовых стратегии фрагментации данных: горизонтальная фрагментация (ГФ) и вертикальная фрагментация (ВФ). Вертикальная фрагментация – это разделение атрибутов на группы, горизонтальная фрагментация – разделение отношения на подмножества таким образом, что каждое подмножество содержит полный набор атрибутов. Примеры горизонтальной и вертикальной фрагментации показаны на

рис. 2 и
рис. 3 соответственно.

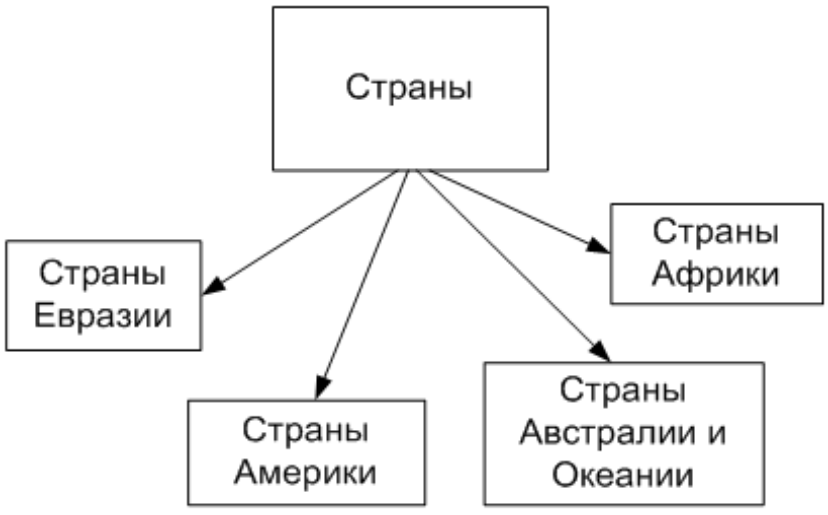


Рис. 2. Горизонтальная фрагментация

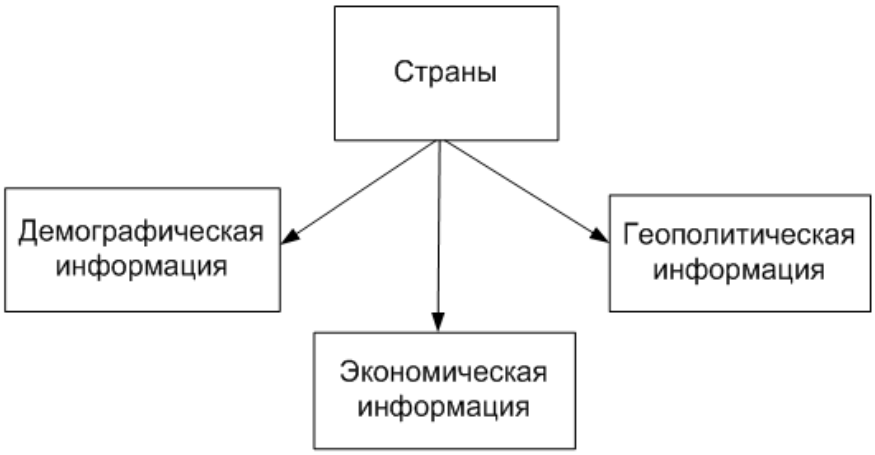


Рис. 3. Вертикальная фрагментация

Горизонтальная фрагментация разделяется на первичную (primary) и вторичную (derived) фрагментацию. Целью первичной горизонтальной фрагментации является оптимизация операций над множествами, позволяющая сузить пространство поиска и обеспечить возможность параллельного выполнения операций. Целью же вторичной горизонтальной фрагментации является повышение скорости навигационных операций, достигаемое за счет объединения множеств объектов различных типов.

Размещение данных

Размещение данных представляет собой процесс принятия решения о месте хранения данных с целью минимизации целевой функции при выполнении запросов. Выделяют следующие типы стратегии размещения данных:

- централизованное размещение данных – вся база данных хранится на одном узле;
- секционированное размещение данных – каждый фрагмент БД хранится на определенном узле;
- реплицированное размещение данных – одна или более копий фрагментов БД хранятся на нескольких узлах.

Репликация данных связана с хранением копий данных на нескольких узлах сети. Поскольку копии фрагментов повышают уровень доступности данных и уменьшают время отклика, репликация уменьшает общие затраты на коммуникации при выполнении запросов.

Стратегия исполнения запроса

Существует две стратегии оптимизации запроса для распределенного исполнения. В первой стратегии сначала строится оптимальный план последовательного исполнения запроса, который затем параллелизуется [4]. Во второй стратегии оптимизация производится с учетом параллельного исполнения [5]. Как показано в [6], для систем без разделения ресурсов (shared-nothing), которые наиболее близки к РаБД, первая стратегия не обеспечивает оптимальности.

Наиболее часто план исполнения запроса представляется в виде дерева операторов, в котором листья представляют отношения, которые участвуют в запросе, а промежуточные вершины – операторы. Параллельная обработка при таком описании разделяется на нескольких видов [7, 8]:

- межоператорная (inter-operation) – параллельное исполнение операторов, лежащих на разных ветвях дерева;
- внутриоператорная (intra-operation) – параллельное исполнение на различных фрагментах одного отношения суб-операторов, полученных в результате декомпозиции оператора;
- конвейерная (pipeline) – узел дерева «потребитель» может начать выполнение до того, как «производитель» завершит свою работу

Сложность решаемых задач

Проектирование схем фрагментации и размещения отношений основывается на информации о способах и методах использования РаБД. Методы использования зависят от стратегии исполнения запросов, которая в свою очередь должна учитывать схемы фрагментации и размещения. Таким образом, по мнению автора, проектирование фрагментации, размещения и стратегии исполнения запросов должны производиться одновременно.

Следовательно, задача проектирования РаБД формулируется так: для данной логической схемы БД, множества запросов и конфигурации ВС описать схему фрагментации, схему размещения фрагментов и стратегии исполнения каждого запроса таким образом, чтобы оптимизировать целевую функцию.

Количество возможных вариантов фрагментации отношения равно n-ному числу Белла $B(n)$, это показано в [9] применительно к ВФ и в [10] для ГФ. Причем для ВФ n означает число атрибутов отношения, а для ГФ – количество минтерм предикатов, используемых во всех запросах, применяемых к отношению. Например, при n = 15 $B(n) \approx 10^9$, для больших n $B(n)$ стремится к n^n.

В [11, 12] показано, что оптимальные алгоритмы фрагментации и размещения данных являются NP-трудными, т.е. с ростом размерности задач их вычислительная сложность растет экспоненциально. Для уменьшения размерности задачи размещения данных в некоторых работах (например, в [13]) применяется алгоритм объединения узлов в группы, после чего размещение фрагментов производится в два этапа – между группами и между узлами внутри группы. Однако, как показано в [14], задача объединения узлов в группы также является NP-полной.

Решение	Количество вариантов
Выбор фрагмента	$\prod r_i$, где r_i – количество реплик i-го фрагмента
Последовательность операции соединения (join)	$(t-1)!$, где t – количество фрагментов
Узел, в котором производится операция соединения	$n(t-1)$, где n – количество узлов
Применение операции semi-join	$3(t-1)$
Применение double-pipelined hash join	$3(t-1)$
Степень внутриоператорного параллелизма	$t(n)!$

Таблица 1. Количество вариантов исполнения запроса

При формировании стратегии исполнения реляционного запроса необходимо принять решения о способе и порядке выполнения операций соединения (join) отношений, а также об используемых методах параллельной обработки. Рассматриваемые решения приведены в табл. 1.

Общее количество возможных стратегий исполнения запроса равно произведению количества вариантов каждого решения. Например, для сети из 4-х узлов и запроса, в котором используются 5 фрагментов, каждый из которых имеет 2 реплики, существует почти 16×10^6 возможных вариантов.

Эвристические алгоритмы

Для решения NP-полных задач применяются эвристические алгоритмы, которые уменьшают вычислительную сложность и позволяют получить близкое к оптимальному решение. Существуют следующие типы эвристических моделей:

- модель слепого поиска, которая опирается на так называемый метод проб и ошибок;
- лабиринтная модель, в которой решаемая задача рассматривается как лабиринт, а процесс поиска решения – как блуждание по лабиринту;
- структурно-семантическая модель, которая считается в настоящее время наиболее содержательной. Она отражает семантические отношения между объектами, составляющими область задачи.

Достаточно полный список существующих эвристических методов приведен в [15]. В [16] отмечено, что для решения задач кластеризации и компоновки наиболее успешно применяются генетические алгоритмы (ГА). В ГА любое решение задачи синтеза представляется хромосомой, состоящей из генов. Значениями генов являются значения проектных параметров. Направленный перебор решений осуществляется с помощью генетических операторов выбора родителей, кроссовера, мутации, селекции, переупорядочения.

Сначала формируется исходное поколение, состоящее из g хромосом. Размер поколения выбирается таким образом, чтобы в достаточной степени был представлен набор существующих решений. Например, поколение из 100 хромосом обычно достаточно для того, чтобы осуществить поиск среди 10^9 доступных решений. Хромосомы оцениваются с использованием функции приспособленности.

Далее случайным образом среди хромосом данного поколения выбираются пары родителей, причем вероятность выбора хромосом с лучшими значениями функции приспособленности должна быть выше. Следующее поколение образуется (селектируется) из g перспективных дочерних хромосом, являющихся результатом ряда операций кроссовера. Кроссовер заключается в разрыве двух родительских хромосом и рекомбинировании образовавшихся хромосомных отрезков. Мутации, т.е. случайные изменения генов, происходят с заданной вероятностью и служат для исключения застревания поиска в ограниченном подпространстве. Разновидности генетических операторов и их сочетаний порождают множество ГА, описание которых можно найти в [17].

Применение генетических алгоритмов

Задача проектирования РаБД формируется тремя NP-полными взаимозависимыми задачами, т.е. входными данными для следующей задачи является решение предыдущей. Таким образом, в методе получения проекта РаБД, близкого к оптимальному, целесообразно применять вложенные генетические алгоритмы. Разрабатываемый автором алгоритм выглядит следующим образом.

Для каждого отношения *R* на основании запросов определяется набор минимальных фрагментов. Каждый минимальный фрагмент характеризуется множеством минтерм предикатов и группой атрибутов. После этого формируется хромосома, описывающая размещение фрагментов на узлах ВС. Если фрагмент размещен в узле, то в соответствующую ячейку ставится 1, в противном случае 0. Структура хромосомы приведена в табл. 2.

Узлы \ фрагменты	Отношение 1				Отношение 2			
	Атрибуты		Предикаты		Атрибуты		Предикаты	
	Атр1	Атр2	Пред1	Пред2	Атр1	Атр2	Пред1	Пред2
Узел1	1	0	1	1	0	1	0	1
Узел2	0	1	1	0	1	1	1	0

Таблица 2. Структура хромосомы схем фрагментации и размещения

Для удовлетворения свойства полноты в каждой колонке должна быть хотя бы одна единица. Каждому атрибуту и предикату назначено значение мощности: для атрибутов мощность определена на основании типа данных, для предикатов – селективности. На основании мощности определяется количество возвращаемых запросом данных.

Для каждой хромосомы из сформированного поколения схем фрагментации и размещения запускается вложенный генетический алгоритм, направленный на получение оптимальной стратегии исполнения запросов. Хромосомы вложенного алгоритма сформированы из решений, принимаемых при исполнении запроса. Функцией приспособленности вложенного алгоритма является критерий оптимальности РаБД. После определенного количества поколений значение функции приспособленности лучшей хромосомы из вложенного алгоритма используется как оценка функции приспособленности внешнего алгоритма. Комбинация хромосом из внешнего и вложенного алгоритмов полностью описывают проект РаБД.

Заключение

В статье рассмотрен жизненный цикл проектирования РаБД и описаны его этапы. Сформулирована задача проектирования РаБД и оценена ее сложность. Обоснован выбор генетических алгоритмов для решения поставленной задачи и предложен подход, позволяющий учесть взаимозависимость этапов проектирования.

Список литературы

1. Stepwise Redesign of Distributed Relational Databases: Technical Report / Hong Kong University of Science & Technology; Kazerouni L., Karlapalem K. – Hong Kong, 1997. – 28 p. – HKUST-CS97-12.
2. Karlapalem K., Navathe S.B., Ammar M. Optimal redesign policies to support dynamic processing of applications on a distributed relational database system // Information Systems. – 1996. – Vol. 21. – № 4. – P. 353–367. – ISSN 0306-4379.
3. Materialization of Redesigned Distributed Relational Databases: Technical Report / Hong Kong University of Science & Technology; Karlapalem K., Navathe S.B. – Hong Kong, 1994. – 44 p. – HKUST-CS94-14.
4. Hong W. Exploiting Inter-Operation Parallelism in XPRS // ACM SIGMOD International Conference on Management of Data. USA. – 1992. – P. 19–28.

5. Lanzelotte R.S.G., Valduriez P., Zait M. Industrial-Strength Parallel Query Optimization: Issues and Lessons // Information Systems. – 1994. – Vol. 19. – № 4. – P. 311–330. – ISSN 0306-4379.
6. Ziane M., Zait M., Hong H.Q. Parallelism and query optimization // International Journal of Computer Science and Engineering. – 1995. – Vol. 10. – № 1. – P. 50–56.
7. Ganguly S., Goel A., Silberschatz A. Efficient and accurate cost models for parallel query optimization (extended abstract) // Fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems. Montreal, Quebec, Canada. – 1996. – P. 172–181.
8. Brunie L., Kosch H. Control strategies for complex relational query processing in shared nothing systems // SIGMOD Rec. – 1996. – Vol. 25. – № 3. – P. 34–39. – ISSN 0163-5808.
9. Hammer M., Niamir B. A heuristic approach to attribute partitioning // ACM SIGMOD international conference on Management of data. Boston, Massachusetts. – 1979. – P. 93–101.
10. Horizontal Class Partitioning for Queries in Object-Oriented Databases: / Hong Kong University of Science & Technology; Bellatreche L., Karlapalem K., Basak G.K. – Hong Kong, 1998. – 27 p. – HKUST-CS98-6.
11. Ahmad I., Karlapalem K., Kwok Y.K., et al. Evolutionary Algorithms for Allocating Data in Distributed Database Systems // Distributed and Parallel Databases. – 2002. – Vol. 11. – № 1. – P. 5–32.
12. Apers P.M.G. Data allocation in distributed database systems // ACM Transactions on Database Systems. – 1988. – Vol. 13. – №3. – P. 263–304.
13. Hababeh I.O., Ramachandran M., Bowring N. A high-performance computing method for data allocation in distributed database systems // Journal of Supercomputing. – 2007. – Vol. 39, N1. – P. 3–18. – ISSN 0920-8542.
14. Graham J.M. Theoretical properties of two problems of distribution of interrelated data // 44th annual Southeast regional conference. Melbourne, Florida. – 2006. – P. 395–398.
15. Джонс Д.К. Методы проектирования: пер. с англ. 2-е изд., доп. // М.: 1986. – 326 с.
16. Норенков И.П. Эвристики и их комбинации в генетических методах дискретной оптимизации // Информационные технологии. – 1999. – №1. – С. 2–7. – ISSN 1994-0408.
17. Батищев Д.И. Генетические алгоритмы решения экстремальных задач. Учеб. пособие. – Воронеж: ВГТУ, 1995. – 69 с.

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЯ

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

Проблематика проецирования данных и доступа к ним для OLAP анализа. Проектирование и разработка уровней агрегирования и обработки данных. Использование распределенных вычислительных систем для обработки данных на различных уровнях. Акцентирование необходимости развития систем автоматической нормализации хранилищ данных.

Введение

История развития систем управления базами данных (СУБД) тесно связана с совершенствованием подходов к решению задач хранения данных и управления транзакциями. Развитый механизм управления транзакциями в современных СУБД сделал их основным средством построения OLTP-систем (систем онлайновой обработки транзакций), основной задачей которых является обеспечение выполнения операций с БД. Современные OLTP-системы позволяют совершать большое количество параллельных изменений, а также предоставляют возможность одновременного обращения множества пользователей к одним и тем же данным для чтения, редактирования и удаления.

Тем не менее, с развитием структур данных и увеличения объема информации в БД, а также повышением требований к уровню, глубине и качеству анализа стало ясным, что OLTP-системы уже не могут эффективно решать возлагаемые на них задачи. Эти системы достаточно успешно решают задачи сбора, хранения и поиска информации, но они не удовлетворяют требованиям, предъявляемым к современным системам поддержки принятия решения (СППР). Основной причиной неудачи развития OLTP-систем послужило противоречие требований к OLTP и СППР. Причина лежит даже не в противоречии требований, предъявленных к OLTP и СППР, а в эволюции требований к самому анализу данных, т.е. что мы хотим получить в ответ от системы.

Несмотря на развитие OLTP-систем, в СППР узким местом является как раз обращение к БД, особенно при удалении и изменении данных. Поэтому в настоящее время при разработке СППР огромное внимание уделяется проектированию БД в зависимости от объема и структуры данных, а также предполагаемых задач по их анализу. Порой изначально неверно спроектированная архитектура БД приводит к неудачному завершению всего проекта.

Обзор средств

Для построения СППР существует множество средств, направленных на упрощение структуры СППР, увеличения производительности и надежности системы. Две основные группы средств, применяемые при реализации, – хранилища данных (ХД) и набор аналитических инструментов. ХД предоставляет собой единую среду хранения корпоративных данных, организованных в структурах и оптимизированных для выполнения аналитических операций. Аналитические средства позволяют конечному пользователю, не имеющему специальных знаний в области информационных технологий, осуществлять поиск, отбор и обработку необходимых данных, а также представляют эти данные в терминах предметной области. Для пользователей различной квалификации СППР располагают различными типами интерфейсов доступа к своим сервисам.

Концепция хранилищ данных

На данный момент существует ряд концептуальных подходов, направленных на снижение нагрузки на БД и увеличения скорости анализа данных. Один из таких под-

ходов, заключающийся в разделении данных по разным БД в зависимости от выполняемых над этими данными операциями, получил название «Хранилище данных» (ХД). Оперативные и часто изменяющиеся данные в этом случае следует хранить в OLTP БД, а данные, предназначенные для глубокого полноценного анализа, – в другой БД. При этом ввиду того, что при анализе выполняются только операции чтения данных и отсутствуют транзакционные задержки, связанные с ожиданием завершения операций записи и изменения данных, скорость анализа существенно возрастает.

Как правило, наиболее удобным способом представления информации для человека является отображение зависимости между различными рассматриваемыми параметрами. Часто возникает потребность в построении зависимостей между большим числом различных параметров, но реляционная БД не удовлетворяет таким требованиям. Такая модель БД, по словам Е. Кодда, «не способна объединять, просматривать и анализировать данные с точки зрения множественности измерений» [3]. Таким образом, было предложено построение многомерной модели данных в виде так называемого «гиперкуба». При построении гиперкуба возникает проблема актуализации всего массива данных в момент добавления новых данных в ХД. Впрочем, как и проектирование БД, проектирование структуры гиперкуба и его физическая реализация влияют как на скорость доступа к данным и скорость их анализа, так и на успешность всего проекта в целом.

По сути, принято создавать гиперкуб, содержащий только агрегированные данные, и лишь при необходимости получения детальных данных выполнять операцию агрегации (Drill Down). При этом следует внимательно выбирать между большим количеством измерений куба и высокой степенью агрегированности данных, так как в первом случае мы получаем дублирование данных, а во втором – набор практических бесполезных данных, требующих извлечения из хранилища, что снижает производительность системы. В случае наличия в структуре слабо связанных или не связанных между собой данных существует возможность снижения нагрузки на ХД путем разнесения таких данных по различным физическим хранилищам, что приведет к минимизации объема обрабатываемых данных и повышению быстродействия ХД в целом. Например, ХД предприятия можно разделить на физические хранилища, содержащие данные не взаимодействующих или мало взаимодействующих отделов предприятия.

По Кодду, многомерное концептуальное представление (multi-dimensional conceptual view) представляет собой множественную перспективу, состоящую из нескольких независимых измерений, вдоль которых могут быть проанализированы определенные совокупности данных. Одновременный анализ по нескольким измерениям определяется как многомерный анализ. Каждое измерение включает направления консолидации данных, состоящие из серии последовательных уровней обобщения, где каждый вышестоящий уровень соответствует большей степени агрегации данных по соответствующему измерению. Так, измерение *Исполнитель* может определяться направлением консолидации, состоящим из уровней обобщения «предприятие – подразделение – отдел – служащий». Измерение Время может даже включать два направления консолидации – «год – квартал – месяц – день» и «неделя – день», поскольку счет времени по месяцам и по неделям несовместим.

Кодд определил 12 правил, которым должен удовлетворять программный продукт класса OLAP (online analytical processing – аналитическая обработка в реальном времени) (таблица 1).

Набор этих требований, послуживших фактическим определением OLAP, следует рассматривать как рекомендательный, а конкретные продукты оценивать по степени приближения к идеально полному соответствию всем требованиям.

№	Правило	Описание
1.	Многомерное концептуальное представление данных (Multi-Dimensional Conceptual View)	Концептуальное представление модели данных в продукте OLAP должно быть многомерным по своей природе, т.е. позволять аналитикам выполнять интуитивные операции «анализа вдоль и поперек» («slice and dice»), вращения (rotate) и размещения (pivot) направлений консолидации.
2.	Прозрачность (Transparency)	Пользователь не должен знать о том, какие конкретные средства используются для хранения и обработки данных, как данные организованы и откуда берутся.
3.	Доступность (Accessibility)	Аналитик должен иметь возможность выполнять анализ в рамках общей концептуальной схемы, но при этом данные могут оставаться под управлением оставшихся от старого наследства СУБД, будучи при этом привязанными к общей аналитической модели. То есть инструментарий OLAP должен накладывать свою логическую схему на физические массивы данных, выполняя все преобразования, требующиеся для обеспечения единого, согласованного и целостного взгляда пользователя на информацию.
4.	Устойчивая производительность (Consistent Reporting Performance)	С увеличением числа измерений и размеров базы данных аналитики не должны столкнуться с каким бы то ни было уменьшением производительности. Устойчивая производительность необходима для поддержания простоты использования и свободы от усложнений, которые требуются для доведения OLAP до конечного пользователя.
5.	Клиент - серверная архитектура (Client-Server Architecture)	Большая часть данных, требующих оперативной аналитической обработки, хранится в мэйнфреймовых системах, а извлекается с персональных компьютеров. Поэтому одним из требований является способность продуктов OLAP работать в среде клиент-сервер. Серверный компонент инструмента OLAP должен быть достаточно интеллектуальным и обладать способностью, строить общую концептуальную схему на основе обобщения и консолидации различных логических и физических схем корпоративных баз данных для обеспечения эффекта прозрачности.
6.	Равноправие измерений (Generic Dimensionality)	Все измерения данных должны быть равноправны. Дополнительные характеристики могут быть предоставлены отдельным измерениям, но поскольку все они симметричны, данная дополнительная функциональность может быть предоставлена любому измерению. Базовая структура данных, формулы и форматы отчетов не должны опираться на какое-то одно измерение.
7.	Динамическая обработка разреженных матриц (Dynamic Sparse Matrix Handling)	Инструмент OLAP должен обеспечивать оптимальную обработку разреженных матриц. Скорость доступа должна сохраняться вне зависимости от расположения ячеек данных и быть постоянной для моделей, имеющих разное число измерений и различную разреженность данных.

8.	Поддержка многопользовательского режима (Multi-User Support)	Зачастую несколько аналитиков имеют необходимость работать одновременно с одной аналитической моделью или создавать различные модели на основе одних корпоративных данных. Инструмент OLAP должен предоставлять им конкурентный доступ, обеспечивать целостность и защиту данных.
9.	Неограниченная поддержка кросс-мерных операций (Unrestricted Cross-dimensional Operations)	Вычисления и манипуляция данными по любому числу измерений не должны запрещать или ограничивать любые отношения между ячейками данных. Преобразования, требующие произвольного определения, должны задаваться на функционально полном формульном языке.
10.	Интуитивное манипулирование данными (Intuitive Data Manipulation)	Переориентация направлений консолидации, детализация данных в колонках и строках, агрегация и другие манипуляции, свойственные структуре иерархии направлений консолидации, должны выполняться в максимально удобном, естественном и комфортном пользовательском интерфейсе.
11.	Гибкий механизм генерации отчетов (Flexible Reporting)	Должны поддерживаться различные способы визуализации данных, т.е. отчеты должны представляться в любой возможной ориентации.
12.	Неограниченное количество измерений и уровней агрегации (Unlimited Dimensions and Aggregation Levels)	Настоятельно рекомендуется допущение в каждом серьезном OLAP-инструменте как минимум пятнадцати, а лучше двадцати, измерений в аналитической модели. Более того, каждое из этих измерений должно допускать практически неограниченное количество определенных пользователем уровней агрегации по любому направлению консолидации.

Таблица 1. Правила оценки программных продуктов класса OLAP

Интеллектуальный анализ данных

В нынешнее время активно используется направление в аналитических технологиях обработки данных – Data Mining, что переводится как «добыча» или «раскопка данных». Нередко рядом с Data Mining встречаются слова «обнаружение знаний в базах данных» (knowledge discovery in databases) и «интеллектуальный анализ данных».

Цель Data Mining состоит в выявлении скрытых правил и закономерностей в наборах данных. Дело в том, что человеческий разум сам по себе не приспособлен для восприятия больших массивов разнородной информации. Человек к тому же не способен улавливать более двух–трех взаимосвязей даже в небольших выборках. Но и традиционная математическая статистика, долгое время претендовавшая на роль основного инструмента анализа данных, также нередко пасует при решении задач из реальной сложной жизни. Она оперирует усредненными характеристиками выборки, которые часто являются фиктивными величинами. Поэтому методы математической статистики оказываются полезными главным образом для проверки заранее сформулированных гипотез (verification-driven data mining).

Современные технологии Data Mining (discovery-driven data mining) обрабатывают информацию с целью автоматического поиска шаблонов, характерных для каких-либо фрагментов неоднородных многомерных данных. В отличие от оперативной аналити-

ческой обработки данных (OLAP) в Data Mining бремя формулировки гипотез и выявления необычных шаблонов переложено с человека на компьютер.

Типы выявляемых закономерностей

Выделяют пять стандартных типов закономерностей, которые позволяют выявлять методы Data Mining.

- *Ассоциация* имеет место в том случае, если несколько событий связаны друг с другом.
- Если существует цепочка связанных во времени событий, то говорят о *последовательности.*
- С помощью *классификации* выявляются признаки, характеризующие группу, к которой принадлежит тот или иной объект. Это делается посредством анализа уже классифицированных объектов и формулирования некоторого набора правил.
- *Кластеризация* отличается от классификации тем, что сами группы заранее не заданы. С помощью кластеризации средства Data Mining самостоятельно выделяют различные однородные группы данных.
- Основой для всевозможных систем *прогнозирования* служит историческая информация, хранящаяся в БД в виде временных рядов. Если удается построить математическую модель и найти шаблоны, адекватно отражающие эту динамику, есть вероятность, что с их помощью можно предсказать и поведение системы в будущем.

Заключение

Обобщая вышеизложенный материал, хочется отметить масштабность и широту текущих исследований вопросов организации хранения и обработки данных, а также большое количество используемых для решения этих задач технологий, каждая из которых даже сама по себе требует существенных ресурсов для изучения и развития. Если методологии и технологии решения задач, связанных с выявлением закономерностей в больших объемах данных, в настоящее время развиваются стремительно, то таким вопросам, как заблаговременное проектирование оптимальных структур ХД и гиперкуба и распределение физического расположения данных, а также поддержка актуальности и оптимальности их состояния и структуры в условиях быстро меняющейся внешней среды, уделяется недостаточно внимания. Однако значимость этих вопросов в скором времени станет существенной в связи с постоянно растущими объемами обрабатываемой информации, а также повышающимися требованиями к оперативности их обработки. Таким образом, в ближайшем будущем необходимо уделять особое внимание созданию подсистем СППР, которые будут в состоянии решать подобные задачи.

Список литературы

1. Power D.J. Web-based and model-driven decision support systems: concepts and issues. Americas Conference on Information Systems, Long Beach, California, 2000.
2. Гасанов Э.Э. О сложности поиска в базах данных // Искусственный интеллект: Межвузовский сборник трудов. – Саратов, Изд-во Саратовского университета, 1993. – С. 41–56.
3. Codd E.F. Relational Database: A Practical Foundation for Productivity // Commun. of ACM. – 1982. – V. 25/ – № 2. – P. 140–155.
4. Литвак Б.Г. Разработка управленческого решения – М.: Издательство «Дело», 2004. – 392 с.
5. Thieranf R.J. Decision Support Systems for Effective Planing and Control. – Englewood Cliffs, N.J: Prentice Hall, Inc, 1982.

ПРИМЕНЕНИЕ СТОХАСТИЧЕСКИХ ФРАКТАЛОВ К НЕКОТОРЫМ ЗАДАЧАМ ИНФОРМАЦИОННОГО ПОИСКА

Я.А. Седова (Астраханский государственный технический университет)
Научный руководитель – к.т.н., профессор И.Ю. Квятковская (Астраханский государственный технический университет)

В статье рассматриваются особенности применения теории стохастических фракталов к некоторым задачам информационного поиска, требующим анализа Web-пространства. Рассматриваются особенности процесса кластеризации при решении этих задач.

Введение

Информационный поиск – наука, изучающая поиск информации в документах, расположенных в базе данных. При этом база данных может быть как автономной, так и сетевой, такой как «Всемирная паутина» (World Wide Web).

Сущность всех задач информационного поиска сводится к поиску информации по некоторому запросу с дальнейшей обработкой результатов. В настоящее время в связи с широким распространением сети Интернет задачи подобного рода стали особенно актуальными. Самый наглядный пример систем, решающих эти задачи, – информационно-поисковые системы, позволяющие пользователю ввести запрос на естественном языке и получить более или менее релевантные результаты, т.е. документы, семантически связанные с запросом.

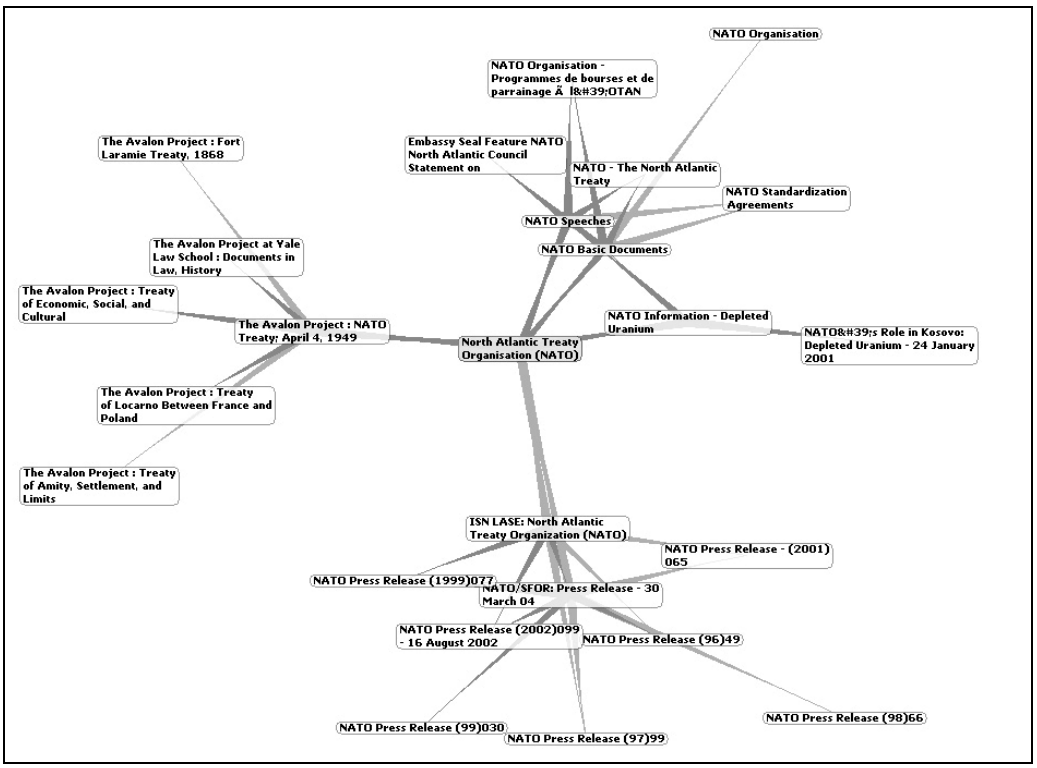


Рисунок. Пример работы Touchgraph Google Browser (взято из [4])

Кластеризация в информационном поиске [1] является одним из способов повысить эффективность поиска. Поскольку ни один из существующих алгоритмов семантического анализа текстовой информации не может работать безошибочно, особую важность приобретает возможность распределить результаты поиска по группам (кластерам), чтобы облегчить для пользователя процесс интерпретации результатов. Один из примеров информационно-поисковой системы, реализующей кластеризацию резуль-

татов поиска, – это система Nigma, разработка которой началась в 2005 г. и продолжается до сих пор. Бета-версия данной системы доступна на сайте www.nigma.ru.

Несколько лет назад рядом исследователей было предложено применить для исследования кластерных систем фрактальный подход. В работе [2] была произведена идентификация информационных объектов как фрактальных множеств на примере множества кластеров статей, имеющих общие цитированные статьи. В 2002 г. российский исследователь С.А. Иванов, обобщив результаты опубликованных работ и собственных исследований, предложил теорию стохастических фракталов для кластерных тематических образований в мировой системе периодических и продолжающихся изданий [3]. Совсем недавно Д.В. Ландэ определил некоторые фрактальные свойства информационных потоков из Интернета, используя в качестве базы данных для эксперимента систему мониторинга сетевых новостей InfoStream [4]. Сервис компании Google Touchgraph (www.touchgraph.com), разработанный для визуализации Web-сайтов, позволяет выявить самоподобие сайтов (рисунок).

В данной работе предлагается применить фрактальный подход для решения некоторых задач информационного поиска, таких как определение рейтинга понятия, заданного пользователем.

Постановка задачи

В общем случае задача определения рейтинга понятия заключается в том, чтобы для некоторого понятия найти все его упоминания в коллекции Web-документов, которая может представлять собой каталог сайтов по определенной тематике, и произвести анализ всех отзывов.

Каждый сайт может рассматриваться как виртуальный эксперт, который обладает рядом критериев, позволяющих определить его вес. Примером таких критериев может быть индекс цитируемости (индекс значимости сайта или отдельной страницы сайта для поисковой системы), ранг страницы PageRank, индекс Хирша (для сайта – максимальное количество дней в месяце, в течение которых было зафиксировано не менее h внешних ссылок на данный сайт). Помимо общепринятых критериев, могут быть использованы также критерии популярных поисковых систем: Яндекс тИЦ, Google PR, Webalta WR/WTR.

Пусть имеется множество из n виртуальных экспертов, каждый из которых обладает m критериев. Тогда определим рейтинг i-го эксперта α_i как

$$\alpha_i = \sum_{j=1}^{m} t_j \ .$$

Благодаря этому получаем возможность определить суммарный рейтинг понятия r как

$$r = \sum_{i=1}^{n} \alpha_i \cdot b_i \ , \qquad (1)$$

где b_i – это оценка, данная понятию i-м сайтом.

Методы исследований

Рассмотрим особенности кластеризации с применением фрактального анализа. В качестве основы будем использовать алгоритм построения иерархической классификации, приведенный в работе [5].

Построим матрицу образов. Теоретически она представляет собой матрицу размерностью $p \times q$, каждая строка которой соответствует наименованию признака Z_i ($i = 1,2,..., p$), а столбец – наименованию образа S_j ($j = 1,2,..., q$).

В задачах информационного поиска для определения признаков может быть использована мера TF·IDF (интегральная значимость):

$$tf \cdot idf = \log_2\left(\frac{N}{n_t}\right) \times \left(\frac{1}{2} + \frac{1}{2} \times \frac{m_{dt}}{m_d}\right),$$

где N – общее количество документов, n_t – количество документов, в которые входит термин t, m_{dt} – число вхождений термина t в документ d, m_d – максимум из m_{dt}. С помощью меры TF·IDF пространство документов, по которым необходимо произвести поиск, может быть представлено в виде числовых векторов (векторная модель VSM) [6].

По матрице образов могут быть вычислены меры сходства и различия. Например, мера сходства Чекановского-Серенсена определяется с помощью следующей формулы:

$$C(S_j, S_k) = \frac{2m(S_j \cap S_k)}{m(S_j) + m(S_k)},$$

где $m(S_j)$ – число элементов множества S_j.

Значения мер сходства документов записываются в матрицу размерностью $q \times q$, строки и столбцы которой соответствуют наименованиям образов S_j ($j = 1,2,..., q$). Матрица симметрична относительно главной диагонали. Чем больше мера сходства двух документов, тем ближе они находятся друг к другу в пространстве термов. Это обстоятельство позволяет построить дендрограмму, а затем с помощью формулы (1) определить рейтинг понятия.

Одним из примеров, в котором целесообразно применение вышеописанных алгоритмов, является задача определения добросовестности поставщиков. На ряде предприятий при закупке товаров и услуг объявляется конкурс среди фирм-поставщиков, претендующих на поставку определенного товара. Проблемой является определение добросовестности каждого из поставщиков на основе анализа информации о нем, найденной в Web. В настоящее время эта задача решается иными способами, когда добросовестность поставщика определяется либо документально, либо в результате опроса, когда превалируют случайные или субъективные факторы. В других случаях организатор конкурса берет на себя проверку. Применение фрактального подхода позволит автоматизировать процесс анализа добросовестности поставщика.

Заключение

В работе исследованы особенности применения фрактального подхода к некоторым задачам информационного поиска, а также особенности процесса кластеризации в применении к задачам подобного рода.

Результаты работы используются автором статьи при разработке автоматизированной интеллектуально-поисковой системы кластерного и фрактального анализа. Система разрабатывается в среде программирования Microsoft Visual Studio 2005 на языке C# и предоставляет возможность анализировать и производить визуализацию результатов выполнения запроса пользователя.

Литература

1. Berry M.W. Survey of Text Mining. Clustering, Classification, and Retrieval. – Springer-Verlag, 2004. – 244 p.

2. Van Raan A.F.J. Fractal geometry of information space as represented by cocitation clustering // Scientometrics. – 1991. – Vol. 20. – № 3. – P. 439–449.
3. Иванов С.А. Стохастические фракталы в информатике // Научно-техническая информация. – 2002. – Сер. 2. – № 8. – С. 7–18.
4. Ландэ Д.В. Фрактальные свойства тематических информационных потоков из Интернет // Регистрация, сбор и обработка данных. – 2006. – Т 8. – № 2. – С. 93–99.
5. Андрейчиков А.В., Андрейчикова О.Н. Компьютерная поддержка изобретательства (методы, системы, примеры применения). – М.: Машиностроение, 1998. – 476 с.
6. Salton G., Wong A., Yang C.S. A vector space model for automatic indexing // Communications of the ACM. – 1975. – Vol. 18. – № 11. – P. 613–620.

МЕТОДЫ ИЗМЕРЕНИЯ ОШИБОК В ЗАДАЧАХ СЕГМЕНТАЦИИ

А.Л. Зеленковский
Научный руководитель – к.т.н., доцент А.В. Меженин

Рассматривается возможность использования RMSE как метода оценки качества сегментации изображений. Реализован алгоритм подсчета ошибок сегментации. Произведено тестирование сегментации, осуществленной методом линейной фильтрации.

Введение

Проводимые исследования относятся к области компьютеризированных процедур анализа и обработки изображений.

Сегментация подразделяет изображение на составляющие его области или объекты [1]. Важным аспектом определения эффективности сегментации изображения является количественная оценка искажений. Был рассмотрен метод вычисления среднеквадратичной ошибки (Root Mean Square Error, RMSE) как один из способов оценки качества сегментации. В результате проведенных исследований получены количественные оценки искажений для сегментации с использованием метода обнаружения перепадов яркости.

Постановка задачи

Необходимо количественно оценить, насколько элементы контура исследуемого объекта в исходном изображении отличаются от контура, полученного с помощью компьютерной сегментации. Один из методов вычисления среднеквадратичной ошибки (Root Mean Square Error, RMSE) – вычисление расстояния между двумя точками [2, 3]. В случае сегментации изображения оно может быть измерено между начальной точкой (до детектирования краев) и конечной точкой. В качестве наблюдаемых для исходного изображения фиксируется несколько крайних точек, затем вычисляется усредненное значение RMSE, которое выражается в единицах исходной системы координат.

Теоретическое обоснование

Один из методов сегментации – линейная пространственная фильтрация. При данном подходе задача сегментации формулируется как задача поиска границ регионов. Полутоновое изображение рассматривается как функция двух переменных (x, y), и предполагается, что границы регионов соответствуют максимумам градиента этой функции. Для их поиска применяется аппарат дифференциальной геометрии – определяем точку изображения как точку перепада, если ее двумерная производная первого порядка превышает некоторый заданный порог. Детектор Собела [1] для обнаружения перепадов использует маски для численного приближения производных G_x и G_y:

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad S_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

Другими словами, значение градиента в центральной точке окрестности вычисляется по формуле

$$g = \sqrt{G_x^2 + G_y^2} = \left\{ \begin{matrix} \left[(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \right]^2 + \\ \left[(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \right]^2 \end{matrix} \right\}^{1/2}, \qquad (1)$$

где z_i – элементы маски Собела. Пиксель с координатами (x, y) является пикселем перепада, если для него $g \geq T$, где T – это выбранный порог.

Ошибка для двух точек в единицах исходной системы координат вычисляется следующим образом:

$$RMSE = \sqrt{[(x_r - x_i)^2 + (y_r - y_i)^2}\ ,$$

где индексом i обозначаются начальные координаты, а r – конечные [3]. В случае совпадения координат $RMSE = 0$.

Результаты экспериментов

Обработка изображений выполнялась в среде MATLAB Image Processing Toolbox (IPT).

Рассмотрим простейший пример определения RMSE для двух точек – исходной, с координатами $a_i(2,2)$ (на рис. 1 обозначена звездочкой), и после преобразования $a_r(3,3)$ (на рис. 1 обозначена кружком).

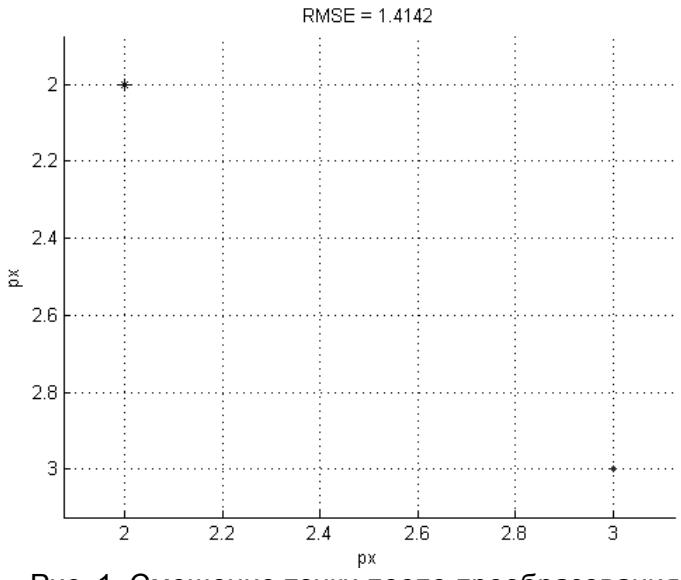


Рис. 1. Смещение точки после преобразования

В этом случае получаем величину $RMSE = 1.4142$.

Для проведения исследований в качестве исходного изображения был использован квадрат, повернутый на 45°, со стороной размером 50 точек (рис. 2).

Рис. 2. Исходное изображение

Для получения координат исходных (калиброванных точек) был разработан алгоритм поиска вершин исходного изображения. Поиск идет от границы изображения до элемента с определенным значением, превышающим некоторый порог. После этого рассматривается столбец и находятся крайние точки.

Пример сбора информации о левой границе объекта:

```
function [Left] = detectLeft(IMG, THRS)
RowSize = size(IMG, 1);
ColSize = size(IMG, 2);
```

```
Left = [];
X = 0;
for n = 1:ColSize
    for m = 1:RowSize
        if (IMG(m,n)<=THRS)
            X = n;
            break
        end
    end
    if (X>0)
        break
    end
end
% Сбор информации об интенсивности в столбце
Left = travelY(IMG,X,THRS);
```

К изображению было применено преобразование Собела. В результате получен контур, к которому применяется рассмотренный выше алгоритм. Получаем два набора точек. Производим оценку RMSE (рис. 3).

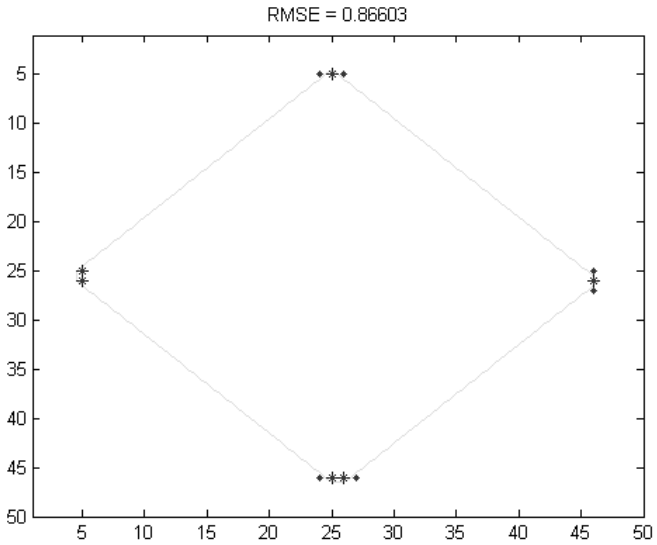


Рис. 3. Результаты экспериментов

Далее к исходному изображению применили двумерный фильтр типа motion (размытие изображения при движении камеры), для этого использовали функцию fspecial из набора Matlab Image Processing с параметрами: расстояние $LEN = 5$ и поворот по часовой стрелке на угол $\theta = 6°$ (рис. 4).

```
LEN = 5;
THETA = 6;
PSF = fspecial('motion',LEN,THETA); % create PSF
BlurredIMG = imfilter(SrcIMG3,PSF,'circular','conv');
```

Рис. 4. Изображение после применения фильтра

Результат работы алгоритма на обработанном преобразованием Собела изображении представлен на рис. 5. Получили новое значение RMSE, которое, как и ожидалось, превысило значение ошибки, полученное в предыдущем эксперименте.

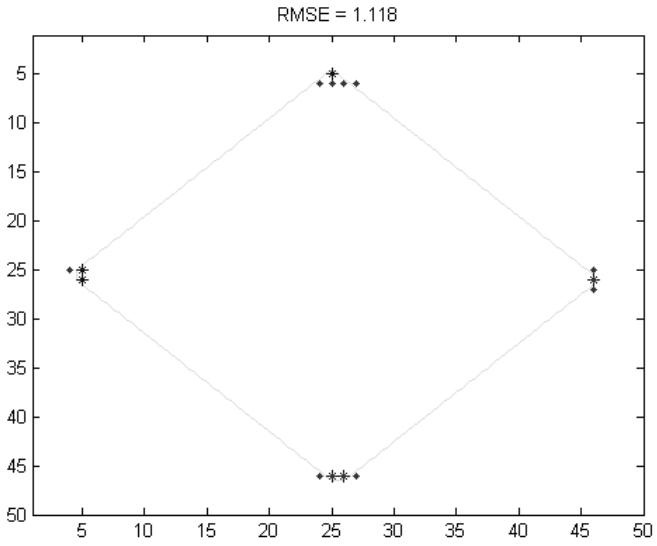


Рис. 5. Результаты экспериментов

Получены зависимости RMSE от порога фильтра Собела. На рис. 6 представлен график, полученный в результате эксперимента.

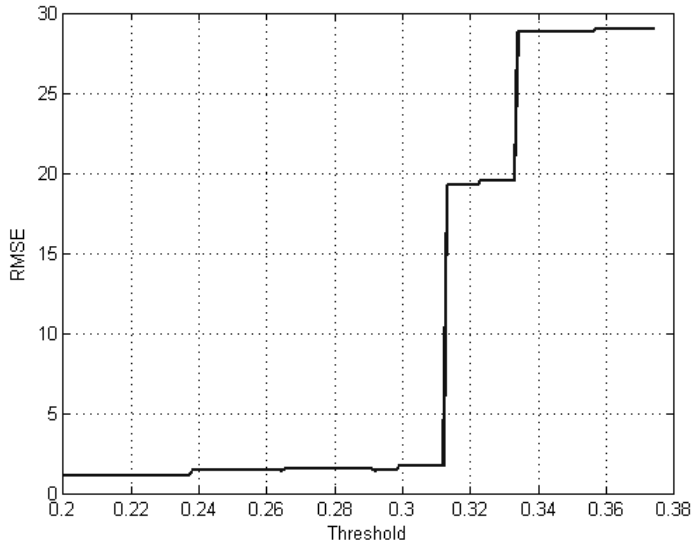


Рис. 6. Зависимость RMSE от выбранного порога для детектора Собела

Дискретный характер графика объясняется тем, что при сегментации точки контура смещаются, появляются или исчезают группами. Таким образом, при данных условиях эксперимента применение оператора Собела показывает нелинейную зависимость, что необходимо учитывать при применении фильтра.

Заключение

Был рассмотрен метод оценки качества сегментации посредствам вычисления среднеквадратичной ошибки (RMSE). В результате проведенных исследований получены количественные оценки искажений для сегментации с использованием метода обнаружения перепадов яркости.

Проведенные исследования показали возможность использования RMSE как одного из методов оценки качества сегментации. Предполагается проводить дальнейшие исследования по уточнению рассмотренного метода.

Литература

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – С. 812–813.
2. Меженин А.В., Тозик В.Т. Оценка погрешности в задачах реконструкции трехмерных моделей // Труды международных научно-технических конференций «Интеллектуальные системы» (AIS'07) и «Интеллектуальные САПР» (CAD-2007). – М.: Физматлит, 2007, Т.3. – С. 79–84.
3. Геоинформационные системы и Дистанционное зондирование Земли [Электронный ресурс] / Среднеквадратичная ошибка (RMSE) . – Режим доступа: http://gis-lab.info
4. D. Martin, C. Fowlkes, D. Tal, J. Malik/ A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms/ – Department of Electrical Engineering and Computer Sciences University of California, Berkeley.

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧАХ ОБРАБОТКИ ТЕКСТОВЫХ ДАННЫХ

Р.О. Белоус, Ю.А. Чернятина
Научный руководитель – к.т.н., доцент И.А. Бессмертный

В докладе рассматриваются вопросы анализа текстов на естественном языке с помощью нейронных сетей. Очерчивается круг задач, требующих анализа текстовых данных, рассматриваются сопутствующие вопросы, проблемы и решения. Обосновывается необходимость развития данного направления.

Введение

Широкое развитие компьютерных технологий и коммуникаций привело к появлению мощных информационных массивов и потоков, аналогов которым в прошлом не существовало. Например, ежедневный объем публикаций электронных средств массовых информаций не позволяет даже бегло с ними ознакомиться одному человеку. Распределение этой задачи среди нескольких исполнителей приведет к проблеме коммуникации и координирования работ между ними. В то же время анализ публикаций может быть очень ценным средством получения актуальной информации, ценной для получения компанией конкурентного преимущества. Многие фирмы имеют специальные отделы информационной безопасности, которые отслеживают и анализируют публикации, посвященные компании и конкурентам. Те же методы могут использоваться, например, для проверки новых сотрудников. Изучая след человека, оставленный в интернет-форумах, блогах и социальных сетях, можно сделать определенные выводы о его личности.

Довольно очевидно, что постоянное изучение всего потока информации по конкретной тематике, более того, даже формирование такого потока без применения автоматизированных средств – практически безнадежная задача. И – конечно же – эти средства появляются. Они могут принимать разнообразные формы, но нас интересуют методы, лежащие в основе этих средств. В течение последних 20 лет активно развиваются методы интеллектуального анализа и получения информации из массивов данных. Этот класс методов называют Data Mining, что на русский язык можно привести примерно как «добыча данных», «извлечение данных». Это междисциплинарная область, возникшая и развивающаяся на базе таких наук, как прикладная статистика, искусственный интеллект, теория баз данных и др. [1]. Изначально Data Mining возник для обработки высокоструктурированных реляционных данных, но развитие технологий требует применения подобных средств для менее структурированных данных.

Примером слабо структурированных данных может служить простой текст. Другим примером (так называемых полуструктурированных данных [2]) служат XML-документы. Текст является универсальным средством представления, накопления и передачи знаний в человеческом обществе. Кроме того, преобразовать текстовый документ в реляционное представление практически невозможно без потери семантики текста и отношений между сущностями, не говоря уж о накладных расходах на подобное преобразование. В то же время в тексте скрыто огромное количество информации, но ее анализ методами Data Mining невозможен в силу неструктурированности. Решением этой проблемы занимаются методы анализа текстовой информации – Text Mining [2, 3].

Задачи Text Mining

Рассмотрим, каковы типичные задачи анализа текстов [2]. Среди них присутствуют как задачи, характерные для Data Mining в целом, так и более специфичные. К первой группе относятся задачи классификации и кластеризации, ко второй – автоматиче-

ское аннотирование, извлечение ключевых понятий, навигация по тексту, анализ трендов, поиск ассоциаций.

Одной из наиболее распространенных задач является классификация. Цель классификации состоит в определении некоторого набора категорий из заранее заданного множества, к которым относится документ. Классификация может использоваться для определения тематики документов, установки тегов и тому подобное.

Второй задачей является кластеризация документов. Целью этого процесса является автоматическое выделение групп семантически похожих документов среди заданного фиксированного множества. Группы формируются исключительно на основе попарной схожести описаний документов. Никакие характеристики групп не задаются заранее.

Автоматическое аннотирование позволяет сократить текст, сохраняя его смысл. При этом, объем аннотации, как правило, регулируется пользователем, что позволяет выбрать желаемый уровень детализации.

Извлечение ключевых понятий позволяет идентифицировать факты и отношения в тексте. Часто такими понятиями оказываются собственные имена существительные: имена людей, названия организаций и другие.

Навигация по тексту предоставляет в распоряжение пользователя компактное интерактивное изображение массива документов с выделенными темами и значимыми терминами. Пользователь может выбрать интересующее его подмножество документов и перемещаться между связанными текстами, например, с помощью автоматически сгенерированных гиперссылок.

Анализ трендов позволяет идентифицировать определенные тенденции в наборах документов. Это может быть использовано, например, для обнаружения перемещения интересов компании от одного сегмента рынка к другому.

Поиск ассоциаций заключается в выявлении ассоциативных отношений между ключевыми понятиями текста.

Нейросетевые методы

Для решения вышеприведенных задач разработан ряд методов. Как правило, это обычные алгоритмические методы. С другой стороны, существует класс мощных методов, которые хороши для решения, как минимум, некоторых из указанных задач. Это нейросетевые методы. Теория нейронных сетей возникла и развилась, как попытка объяснить принципы функционирования мозга. Нейронные сети оказались весьма гибким и эффективным методом решения некоторых задач, которые плохо решаются другими методами.

Нейронная сеть состоит из формальных нейронов. Каждый нейрон является простым вычислительным элементом со многими входами и одним выходом. Каждый вход нейрона с номером i обладает синаптическим весом w_i и уровнем сигнала x_i. Как правило, нейроны вычисляют взвешенную сумму входов, и результат подвергается воздействию некоторой нелинейной функции f (функции активации нейрона): $y = \mathrm{f}(\sum_i x_i w_i)$ (см. рис. 1).

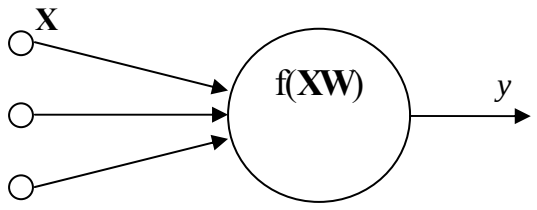


Рис. 1. Формальный нейрон с векторным входом ***X***, вектором весов ***W***, функцией активации *f* и выходным сигналом *y*

Нейроны организуются в слои. Векторный вход $\mathbf{X} = \{x_i\}_0^N$ подается на каждый нейрон слоя (т.е. каждый нейрон имеет *N* входов). Выход всех нейронов одного слоя служит входом для следующего слоя. Нейронные сети обычно классифицируется по количеству слоев (однослойные и многослойные), по наличию обратных связей (реккурентные или прямого распространения), по способу обучения (с учителем или без учителя) [5].

Применение нейронных сетей для анализа текстовых данных не слишком распространено по ряду причин. Одна из основных причин та, что нейронные сети не приспособлены к работе с символьными данными, а требуют числовых входов. Тем не менее, существуют определенные возможности для такой обработки и, возможно, они могут послужить ключом к объяснению того, как наш мозг работает с текстовыми данными. Как известно, наше мышление носит символьный характер (логика, мы не говорим здесь о явлении интуиции). Существует значительный пробел в знаниях о том, как работает мозг на «низком» уровне (нейронные сети представляют собой неплохую модель этих процессов) и как работает наше сознание. Возможно, изучение возможностей нейронных сетей для обработки текстовых данных позволит когда-нибудь пролить свет на этот вопрос.

Наиболее интересными архитектурами нейронных сетей, пригодными для обработки текстовых данных, являются самоорганизующиеся карты Кохонена и сети рекурсивной автоассоциативной памяти. Расскажем о них подробнее.

Конкурентные сети

Принципы работы и обучения самоорганизующихся карт были сформулированы финским ученым Тойво Кохоненом в 1982 г. Основной идеей Т. Кохонена является введение в правило обучения нейрона информации о его расположении. По Кохонену, нейронная сеть имеет один входной слой с числом нейронов, равным числу входов, и единственный скрытый (выходной) слой нейронов, образующий одномерную (линия) или двухмерную (прямоугольник) решетку. По аналогии с топографическими картами такую сеть также называют картой Кохонена. Архитектура сети изображена на рис. 2.

Для этой парадигмы обучение проводится без учителя, т.е. в процессе обучения нет сравнивания выходов нейронов с эталонными значениями. В процессе обучения на вход такой нейронной сети последовательно подаются обучающие примеры. После подачи очередного примера определяется наиболее схожий нейрон, т.е. нейрон, у которого скалярное произведение весов и поданного на вход вектора минимально. Такой нейрон объявляется победителем и будет являться центром при подстройке весов у соседних нейронов. Правило обучения, предложенное Кохоненом, предполагает соревновательное обучение с учетом расстояния нейронов от «нейрона победителя» и записывается в виде $\Delta \mathbf{w}_i^r = \beta \Lambda(|i - i^*|)(\mathbf{x}^r - \mathbf{w}_i)$, где $\Lambda(|i - i^*|)$ – функция соседства, определяющая величину корректировки веса нейрона, $\mathbf{w}_i$ – веса i-го нейрона, β – скорость обучения. Для нейрона-победителя функция соседства равна 1 и затем плавно (по линейному или экспоненциальному закону) уменьшается при удалении от него. Таким образом, в процессе обучения, подстройка весов происходит не только в одном нейроне-победителе, но и в его окрестностях.

После окончания процесса обучения карта Кохонена классифицирует входные примеры на группы схожих. Вся совокупность нейронов в выходном слое точно моделирует структуру распределения обучающих примеров в многомерном пространстве. Уникальность технологии самоорганизующихся карт состоит в преобразовании *N*-го пространства в двух- или одномерное. Единственное, что надо помнить – такое преоб-

разование сопряжено с некоторыми ошибками. Две точки, близко лежащие на карте Кохонена, будут близки и в *N*-мерном входном пространстве, но не наоборот.

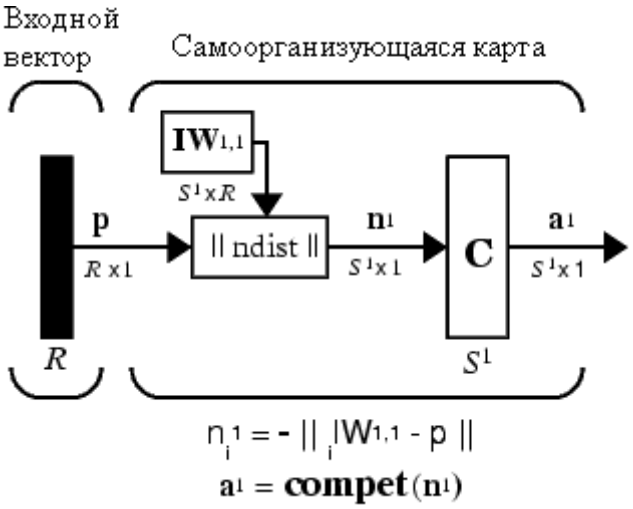


Рис. 2. Архитектура самоорганизующейся карты Кохонена

На основе данной архитектуры финскими учеными были предложены решения задач кластеризации документов и навигации в текстовых массивах [4]. Разработанный ими метод WEBSOM предназначен для представления массивов полнотекстовых документов в виде двумерной карты, раскраска которой отражает плотность распределения образов документов. Конкретные документы при этом связываются со своими областями карты, причем к каждой области может относиться множество близких по содержанию документов – тематический класс. В свою очередь, близким областям обычно соответствуют близкие классы документов, что является основной особенностью карты. Области карты именуются в зависимости от содержания документов, к ним относящихся. Пользователь выбирает на карте интересующую область и получает класс соответствующих ей документов близкого содержания. Если же ищутся документы, включающие некоторые слова, то результаты поиска также могут быть отражены на карте, что достигается выделением областей, которым принадлежат найденные документы. В итоге пользователь получает возможность оценить тематическое распределение искомой информации.

Развитием конкурентных сетей, подобных сетям Кохонена, являются сети LVQ (Linear Vector Quantization). Нейронная сеть LVQ состоит из двух последовательно соединенных слоев нейронов: конкурирующего слоя и линейного слоя. Оба слоя НС LVQ содержат по одному конкурирующему на каждый кластер и одному линейному нейрону на каждый целевой класс. Обозначим S1 – количество кластеров, S2 – количество целевых классов (S1 всегда будет больше, чем S2). Конкурирующий слой производит разделение множества входных векторов X на классы, выделяя центры сосредоточения входных векторов **mi**. Для этого определяются расстояния между входными векторами **x** и начальными значениями центров кластеров. Линейный слой преобразует класс входного вектора, определенный конкурирующим слоем (кластер a1), в класс, определенный пользователем – целевой класс a2. Соответствующие n2 подаются на выходы всех линейных нейронов, образуя двоичный вектор a2, все элементы которого равны 0, за исключением элемента, который соответствует целевому классу (этот элемент равен 1). Таким образом, LVQ – сеть имеет два слоя: соревновательный – выполняющий кластеризацию входных элементов; линейный – соотносящий кластеры входных элементов с выходными классами. Архитектура сети LVQ изображена на рис. 3.

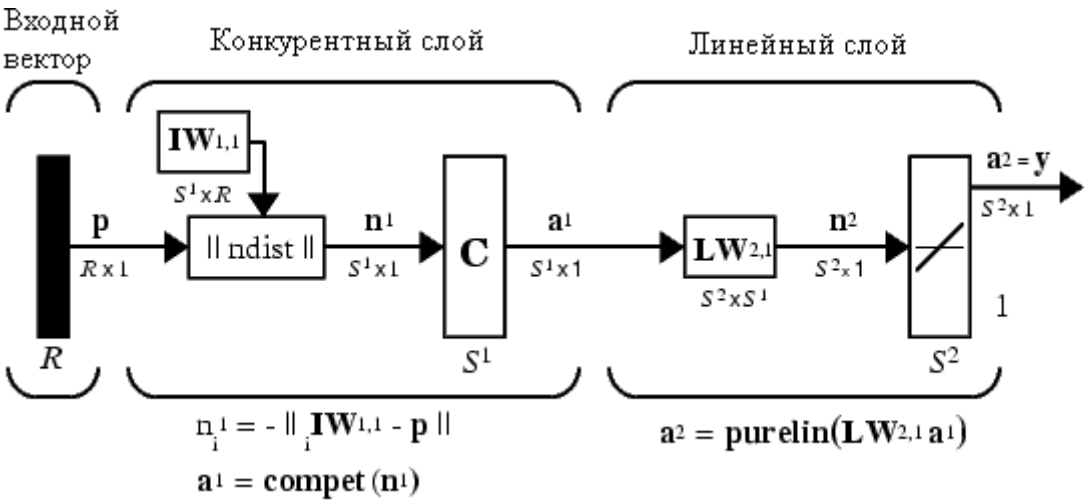


Рис. 3. Архитектура сети LVQ

Рекурсивная автоассоциативная память

Третьей интересующей нас архитектурой является рекурсивная автоассоциативная память (RAAM), предложенная в 1990 г. Поллаком [6]. Целью RAAM является представление символьных структур средствами нейронных сетей. Символьные структуры представляют собой деревья фиксированной валентности. По существу, сеть RAAM является автоассоциативной сетью с обратным распространением ошибки. Во входном и выходном слоях RAAM элементы организованы в поля, где каждое поле содержит одинаковое число элементов. Число полей определяется валентностью кодируемых деревьев, а число элементов в скрытом слое соответствует числу элементов одного поля. Если взять сеть RAAM, обученную представлять некоторое множество деревьев, то эту сеть можно рассматривать как два автомата: первый слой весовых значений является автоматом для создания представлений деревьев (его называют кодером), а второй слой – автоматом, разбивающим созданное представление на составные части (его соответственно называют декодером). Схематичное изображение архитектуры RAAM представлено на рис. 4.

Представление некоторой структуры в сети RAAM порождается значениями скрытого активности слоя. В процессе создания представления всего дерева RAAM генерирует представления для каждой внутренней вершины (поддерева). Деревья являются рекурсивными структурами, и RAAM формирует их представления некоторым рекурсивным образом с помощью обратной подпитки входного слоя ранее сформированными представлениями частей, происходящей в определенные моменты времени.

Конкретные символы в сети представляются векторами. Как правило, применяются ортогональные векторы. Размерность векторов, представляющих символы, определяет минимальное число элементов, образующих поле RAAM. Поскольку число элементов скрытого слоя меньше числа входных элементов, то им порождается сжатое представление. Но показатель сжатия зависит от конкретного набора входных данных, в связи с чем может возникнуть необходимость дополнить векторы нулями.

Значения активации скрытого слоя формируют распределенное пространственное представление символов, поданных на вход сети. Пространственное размещение разных символьных структур может быть проанализировано с помощью самоорганизующейся карты с целью выявления их топологической структуры. Пока сложно сказать, насколько полученный результат будет содержателен. Но мы уверены, что подобная операция может быть применена, например, для выделения определенных структурных шаблонов в тексте.

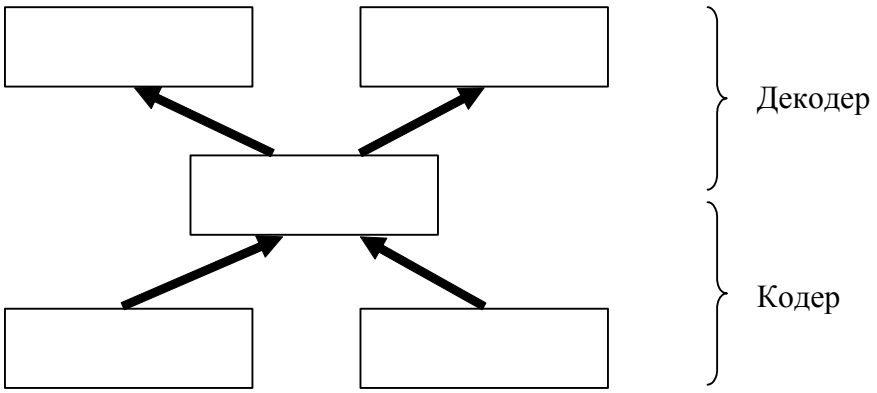


Рис. 4. Схематичная архитектура двоичной сети RAAM

Заключение

Представление символьных структур и их анализ с помощью нейронных сетей представляется интересным и полезным направлением в теории нейронных сетей. Мы рассмотрели несколько архитектур нейронных сетей, которые, с нашей точки зрения, заслуживают более подробного изучения в данном контексте. Наша дальнейшая работа будет посвящена конкретным экспериментам в этой области. Таким образом, очерчен круг проблем для дальнейшего изучения и экспериментирования и обозначены потенциальные возможности применения некоторых нейросетевых моделей для анализа текстовой информации.

Литература

1. Чубукова И.А. Data Mining. – М: БИНОМ, 2006. – 382 с.
2. Барсегян А.А. и др. Технологии анализа данных. – СПб: БХВ-Петербург, 2007. – 384 с.
3. Башмаков А.И., Башмаков И.А. Интеллектуальные информационные технологии. – М: Издательство МГТУ им. Н.Э. Баумана, 2005. – 304 с.
4. WEBSOM – Self-organizing Maps for Internet Exploration [электронный ресурс]. – Режим доступа: http://websom.hut.fi/
5. Хайкин С. Нейронные сети: полный курс. – М: Вильямс, 2006. – 1104 с.
6. Каллан Р. Основные концепции нейронных сетей. – М: Вильямс, 2001. – 288 с.

СИСТЕМА АВТОМАТИЗИРОВАННЫХ ИССЛЕДОВАНИЙ PICLAB

Г.А. Коченятов, С.И. Сутковой
(Ярославский государственный университет им. П.Г. Демидова)
Научный руководитель – к.т.н., доцент В.В. Хрящев
(Ярославский государственный университет им. П.Г. Демидова)

Для анализа, разработки и оптимизации различных алгоритмов цифровой обработки изображений, а также для проведения лабораторных работ по соответствующему курсу для студентов физического факультета ЯрГУ им. П.Г. Демидова, обучающихся по специальности «Радиофизика и электроника», разработана исследовательская среда PicLab (Picture Laboratory). В ее основу положены исследования, проводившиеся на протяжении нескольких лет в лаборатории «Цифровые цепи и сигналы» ЯрГУ.

Введение

Цифровая обработка изображений в настоящее время широко используется в системах телекоммуникаций, медицине, в оборонной сфере, в системах формирования изображений. В связи со скорым переходом России на цифровое телевещание и присоединение к общеевропейской системе DVB (распоряжение Правительства РФ от 25.05.04 №706-р «О внедрении в Российской Федерации европейской системы цифрового телевизионного вещания DVB») вопросы оценки качества передающих и приемных систем, восстановления искаженного цифрового сигнала, моделирования реальных искажений, возникающих в каналах передачи цифровой информации, а также разработки новых алгоритмов обработки цифровых изображений становятся особенно актуальными. С появлением новых стандартов сжатия и с увеличением объемов передаваемой мультимедийной информации возникает задача автоматического выбора параметров сжатия для текущих условий передачи и хранения информации с целью получения максимально возможного качества на выходе системы [1].

Представленный в данной работе программный комплекс позволяет автоматически оценивать качество цифровых изображений с помощью широко известных, а также разрабатываемых нами алгоритмов оценки, получать различные математические и статистические характеристики изображения, а также исследовать различные методики и алгоритмы обработки изображений (восстановление, сжатие, моделирование шумов, изменения в пространственной и частотных областях) с целью определения параметров для их максимальной эффективности.

Еще одной практической задачей, которую решает комплекс, является обучение по курсу «Цифровая фильтрация» посредством проведения лабораторных работ, так как процесс встраивания и тестирования, исследуемых в ходе работы алгоритмов максимально упрощен и понятен.

Общие сведения о программном комплексе PicLab

PicLab можно разделить на несколько основных составляющих, среди которых можно выделить библиотеку алгоритмов цифровой обработки изображений, средства определения качеств изображений (СКО, ПОСШ, УИК, спектр, гистограмма и др.), средства автоматического проведения исследований, библиотеку тестовых изображений. На сегодняшний день в программе поддерживаются два вида цифровых изображений: в градациях серого (8 бит/пиксель) и полноцветные (24 бита/пиксель).

Подавляющее большинство алгоритмов фильтрации последовательно применяет некоторую операцию к каждому пикселю изображения, поэтому для добавления подобного алгоритма в *PicLab* потребуется запрограммировать только преобразование над одним пикселем, которое будет автоматически применено ко всем каналам всего изображения. В случае, когда пользователь должен иметь возможность настройки па-

раметров алгоритма, нет необходимости программировать диалоговое окно, поскольку *PicLab* способен автоматически генерировать подобные окна.

Добавление нового алгоритма в исследовательскую среду может быть осуществлено всего за несколько минут. Алгоритмы хранятся в динамически подключаемых библиотеках (*DLL*), поэтому добавление алгоритма в *PicLab* осуществляется простым копированием соответствующего файла в соответствующую папку – не требуется перекомпиляция всей программы.

В процессе работы в среде *PicLab* ведется журнал произведенных действий, поэтому можно без дополнительных усилий сравнивать результаты работы различных алгоритмов. Например, последовательность действий: открыть изображение, добавить Гауссов шум, обработать изображение фильтром Винера, обработать изображение билатеральным фильтром – будет отражена в журнале работ в виде таблицы. Также предусмотрен экспорт этой таблицы в *MS Excel* или в *MS Word*.

Для сравнения получаемых результатов с результатами других исследователей *PicLab* включает необходимый набор стандартных тестовых изображений (512×512 пикселей), которые часто встречаются в различных источниках. Исследовательская среда *PicLab* может быть использована в системах телекоммуникаций, радио- и гидролокации, сейсмологии, роботехнике, радиоастрономии, медицине.

Тестовые изображения

В среду *PicLab*, помимо множества алгоритмов обработки изображений, моделей шумов, алгоритмов сжатия и средств автоматизации исследований, входит набор тестовых изображений, среди которых наиболее известным является изображение «Lenna» (рис. 1).

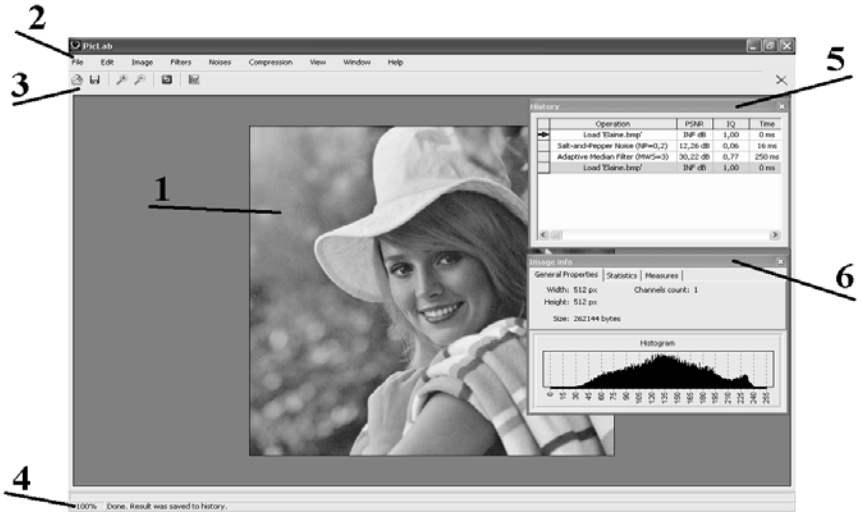


Рис. 1. Рабочее окно среды PicLab

Все тестовые изображения представлены в градациях серого и имеют размер 512×512 пикселей. PicLab позволяет также обрабатывать любые 8-битные (в градациях серого) и 24-битные (цветные) изображения в форматах BMP и JPEG.

Пример проведения исследований в среде PicLab

Предположим, необходимо проанализировать качество удаления из изображения импульсного шума «соль-и-перец» (вероятность искажения пикселя 0,40) двумя фильтрами:

– адаптивным прогрессивным переключающимся медианным фильтром (размер окна детектора 5×5, размер окна фильтра 3×3);
– адаптивным медианным фильтром (максимальный размер окна 22×22, рекурсия).

В качестве тестового можно использовать полутоновое изображение «Lenna» размером 512×512 пикселей.

Ниже представлена последовательность действий для проведения эксперимента.

1. Запустить среду PicLab, используя ярлык на рабочем столе или меню «Пуск».

2. Открыть изображение командой **File ► Open Image** или нажать соответствующую кнопку на панели инструментов. При этом появится диалоговое окно **Open Image** (рис. 2). Следует выбрать изображение «Lenna» (тестовые изображения находятся в папке C:\Program Files\PicLab\Samples) и нажать кнопку **Открыть**. При выделении файла изображения в диалоговом окне справа появляется его миниатюра.

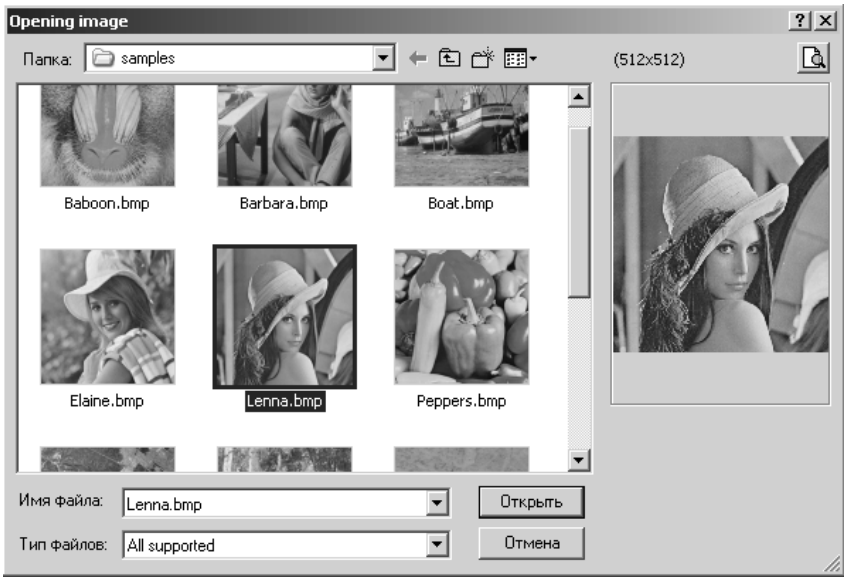


Рис. 2. Диалоговое окно Open Image

3. Выбрать модель шума в меню **Noises ► Impulse Noise ► Salt-and-Pepper Noise.** В открывшемся диалоговом окне нужно установить вероятность искажения пикселя (**Noise percent**) равной 0,40 (рис. 3). При установленной галочке **Предварительный просмотр** (**Full Preview**) будет видно, как шум действует на все изображение.

4. Провести обработку зашумленного изображения выбранными фильтрами:
– выберем адаптивный прогрессивный переключающийся медианный фильтр (Filters ► Detector Median Filters ► APSM Filter) с параметрами: размер окна детектора (Detector window size) – 5, размер окна фильтра (Filter window size) – 3 (рис. 4, а);
– вернемся с помощью панели History к изображению с шумом «соль-и-перец» и используем адаптивный медианный фильтр (Filters ► Median Filters ► Adaptive Median Filter) с размером окна (Window size) – 22 и установленным флажком Рекурсивное выполнение (Recursive executing) (рис. 4, б).

5. Анализ полученных результатов. Для этого используется окно **History** (рис. 5). Можно сравнить параметры ПОСШ (**PSNR)**, УИК (**IQ)** и время (**Time**), затраченное на обработку изображения разными фильтрами.

Двигаясь по списку преобразований (окно **History**), можно визуально оценить изображение до и после воздействия фильтров. Можно видеть, что адаптивный переключающийся медианный фильтр превосходит адаптивный медианный фильтр для заданной степени зашумления по всем критериям.

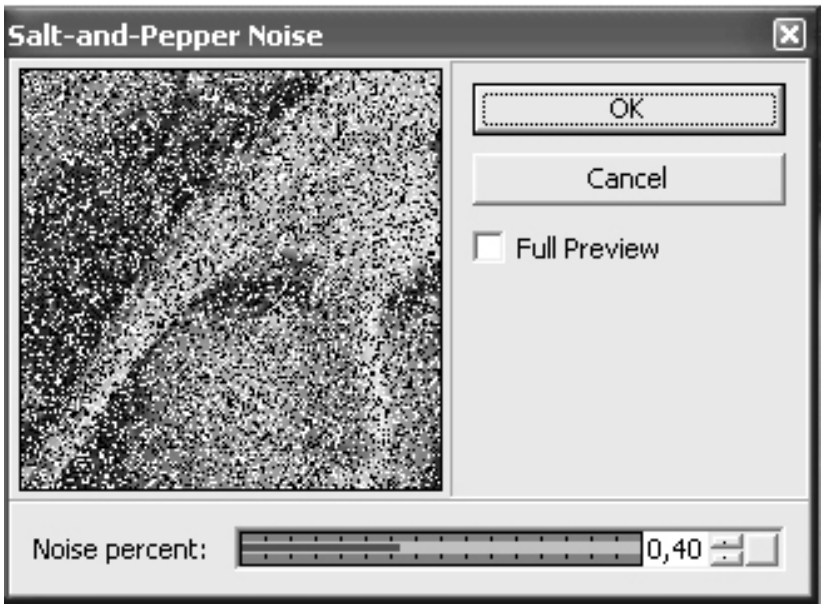
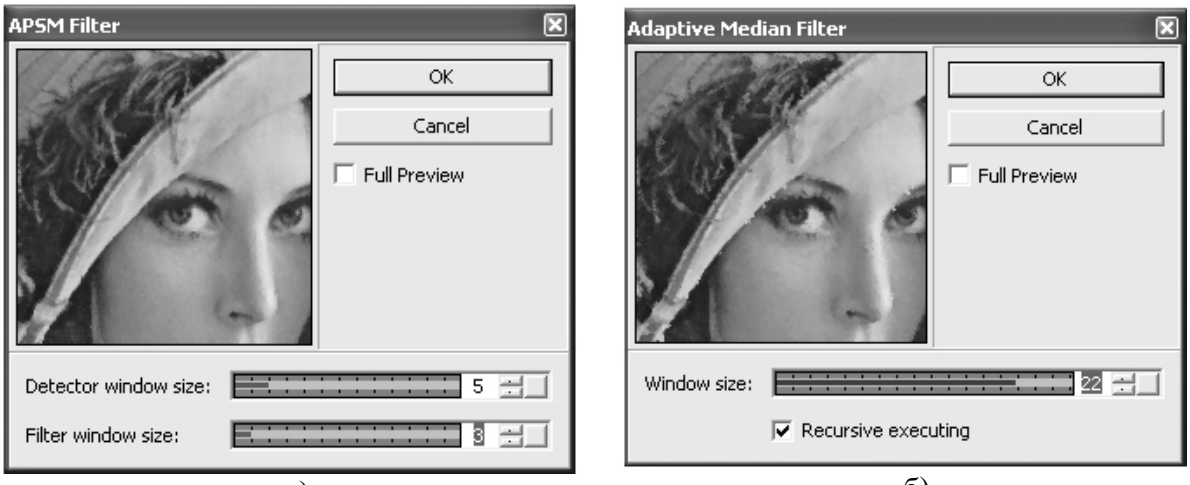


Рис. 3. Установка параметров шума

В среде PicLab предусмотрена возможность автоматического построения экспериментальных зависимостей поведения различных фильтров при обработке одного и того же изображения, искаженного некоторым набором степеней зашумления (рис. 6). Для проведения подобного эксперимента понадобится следующая последовательность.

1. Запустить анализатор алгоритмов **File ► Start Analyzer**.
2. Выбрать искажение, в роли которого могут выступать модели шумов или, например, алгоритмы сжатия изображений.
3. Выбрать параметр искажения, который будет меняться в процессе эксперимента, задать пределы изменения параметра и количество шагов.
4. Задать оставшиеся параметры искажения.
5. Выбрать один или несколько алгоритмов фильтрации и задать их параметры.
6. Выбрать критерий (ПОСШ, СКО или др. [2]).
7. Запустить эксперимент.



а) б)

Рис. 4. Установка параметров фильтров APSM Filter и Adaptive Median Filter

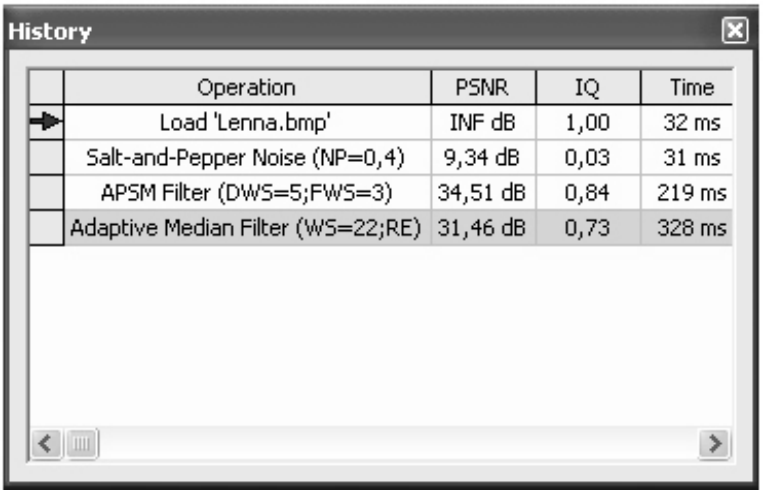


Рис. 5. Окно History после проведения эксперимента

В результате проведения эксперимента будет получена необходимая зависимость. Результат эксперимента можно также экспортировать в Microsoft Word или Microsoft Excel.

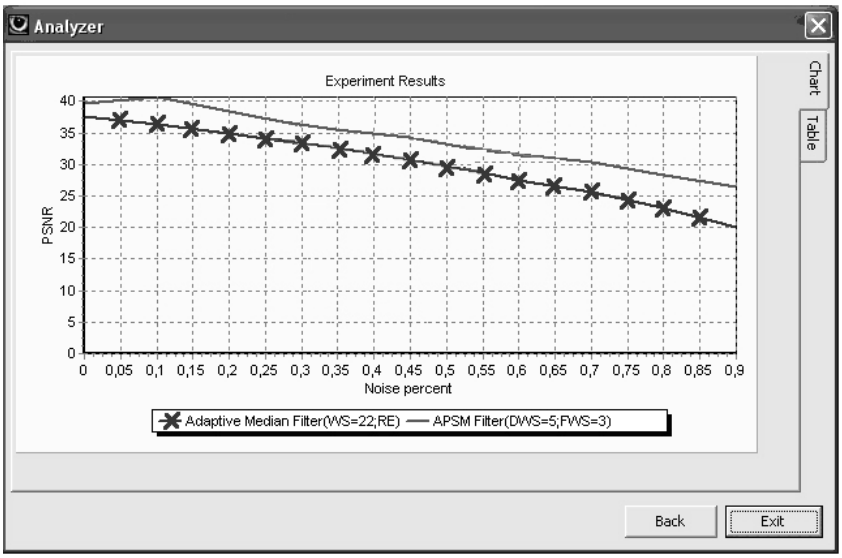


Рис. 6. Окно Analyzer после проведения эксперимента

Особенности представления алгоритмов в среде PicLab

В среде PicLab существует простая процедура добавления новых алгоритмов обработки цифровых изображений. Такая возможность предусмотрена исходя из следующих соображений. Во-первых, необходим механизм для быстрого изменения набора алгоритмов в среде при решении различных задач. Во-вторых, алгоритмы обработки цифровых изображений должны образовывать непротиворечивую и интуитивно понятную объектно-ориентированную иерархию, которую можно легко расширять новыми алгоритмами.

Все алгоритмы хранятся в динамически подключаемых библиотеках, функционирующих по принципу плагинов (plug-in). Идея плагинов состоит в том, что для расширения функциональности основной программы не требуется ее переработка и перекомпиляция, достаточно поместить специальным образом подготовленную библиотеку в

некоторое место (папку на диске), и основная программа при запуске загрузит все подобные библиотеки, расширяя, таким образом, свою функциональность. Такое решение позволяет легко изменять набор возможностей программы без особых усилий.

До непосредственного рассмотрения примеров встраивания алгоритмов в программу необходимо обсудить некоторые особенности иерархии внутренних классов.

Классификация алгоритмов

В среде PicLab все алгоритмы обработки изображений разделены на три типа:
- фильтры (медианный, усредняющий и др.);
- шумы (аддитивный гауссов, импульсный и др.);
- искажения при сжатии с потерями (JPEG, JPEG2000).

Большой класс алгоритмов обработки изображений сводится к применению некоторого преобразования к каждому пикселю. В случае аддитивного шума, например, это добавление случайной компоненты. В результате применения такого преобразования ко всем пикселям получается выходное изображение [3].

По причине принципиального сходства алгоритмов выделена базовая абстракция «Operation» (операция, процесс), общая для всех этих типов. Перечислим основные черты указанной абстракции:
- операция применяется к цифровому изображению, после завершения операции получаем другое цифровое изображение;
- операция имеет уникальное название (например, Gaussian Noise – гауссов шум);
- для операции определено название родительской группы (например, Additive Noises – аддитивные шумы);
- определен тип операции (фильтр, шум или алгоритм сжатия).

Кроме того, предложенная абстракция содержит указания о возможности применения конкретного алгоритма только к цветным изображениям (например, для операции перевода цветного изображения в градации серого) и указания о том, что конкретный алгоритм полностью изменяет изображение, а вычисление объективных оценок теряет смысл (например, операция отражения).

Заключение

Разработанный программный комплекс для проведения автоматизированных исследований цифровых изображений позволяет максимально быстро и просто тестировать новые алгоритмы обработки цифровых сигналов, подбирать оптимальные параметры передачи по каналам связи, получать все необходимые для исследований статистические характеристики сигналов. Для студентов специальности «Радиофизика» данная система предоставляет удобную платформу для курса «Цифровая фильтрация».

Система PicLab постоянно обновляется и дополняется разрабатываемыми алгоритмами обработки.

Литература

1. Приоров А.Л., Ганин А.Н., Хрящев В.В. Цифровая обработка изображений: Учеб. пособие. – Ярославль, 2001.
2. Арляпов С.А., Приоров А.Л., Хрящев В.В. Модифицированный критерий оценки качества восстановленных изображений // Цифровая обработка сигналов. – 2006. – №2. – С. 27–33.
3. Гонзалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005.

МЕТОД ФОРМИРОВАНИЯ ЭЛЕКТРОННОГО ОПИСАНИЯ ИЗДЕЛИЯ

Ю.В. Донецкая (Центральный научно-исследовательский институт «Электроприбор»)
Научный руководитель – д.т.н., профессор Ю.А. Гатчин

В статье представлен метод формирования описания изделия приборостроения. Его описание формируется из структуры изделия, на основе которой автоматически создаются табличные документы.

Введение

Техническая информация является важным активом предприятия. За счет конкурентной борьбы возрастает номенклатура выпускаемых изделий, что обусловливает появление большого количества их модификаций и исполнений.

Учитывая количество информации, выпускаемой на изделия, а также количество самих изделий, конкурентоспособность предприятия возможна лишь при его постоянной адаптации под изменяющиеся условия ведения бизнеса. В связи с этим происходит серьезное переосмысление подходов к организации промышленного производства, способов взаимодействия с заказчиками, борьбы за них и реализация их требований. Это невозможно без системы, связывающей все сведения об изделии на этапах его жизненного цикла. Такой системой является система управления жизненными циклами изделий, или PLM система. Первые такие системы появились в начале 80-х гг. Их появление было обусловлено необходимостью создания среды поддержания целостной, непротиворечивой и актуальной информации внутри группы разработчиков.

Постоянно совершенствуясь, системы управления жизненными циклами изделий позволяют решать следующие задачи:

1) взаимодействие с автоматизированными системами;
2) управление конфигурацией изделия;
3) управление вариантами состава изделия;
4) управление версиями изделия;
5) управление изменениями;
6) хранение документов и т.п. [1, 2].

Другими словами, основным объектом PLM системы является управление составом или структурой изделия. Структурой изделия является конструкторский документ, содержащий состав сборочной единицы, комплекса или комплекта и иерархические отношения (связи) между его составными частями и другие данные в зависимости от его назначения [3]. Таким образом, структура содержит все основные данные об изделии, что свидетельствует о важности этого информационного объекта. Однако необходима дальнейшая унификация использования PLM систем, так как в настоящее время не определена последовательность действий при формировании структуры изделия, а также возможность дальнейшего использования этого объекта, исключая его использование для технологической подготовки производства.

Учитывая это, в работе представлен способ формирования электронной структуры изделия и возможность ее последующего применения.

Жизненный цикл изделия приборостроения

Жизненный цикл изделия представлен стадиями: проектирование, подготовка производства, производство и реализация, эксплуатация и утилизация.

Проектирование – стадия, в процессе которой производится формирование структуры изделия, его конструкции и схемотехнических решений, создается программная документация. Подготовка производства – стадия, в процессе которой производится техноло-

гическая подготовка производства, сопровождаемая выпуском соответствующей документации. На данной стадии может производиться корректировка конструкторской документации, выпущенной на предыдущей стадии. Производство и реализация – стадия, в процессе которой производится непосредственно изготовление изделия, его сборка, а также испытания и поставка заказчику. В процессе изготовления и испытаний могут отрабатываться замечания по конструкторской, программной и технологической документации. Перед испытаниями создается комплект эксплуатационной документации.

Эксплуатация – стадия, в процессе которой, производится сопровождение изделия у заказчика. В данном случае могут проводиться корректировки конструкторской, программной и технологической документации с последующим изготовлением составных частей изделия. Последней стадией является утилизация. Она предполагает списывание изделия, прекращение его производства, сопровождения и разработку его модификаций.

Особенности разработки средств автоматизации

Для сопровождения изделия на всех представленных стадиях разработаны разнообразные системы управления, каждая из которых представляет свои функции по управлению составом изделия. К сожалению, в настоящий момент создание таких структур производится вручную, а автоматизация их создания возможна только в случае использования систем трехмерного проектирования. Помимо этого, разработаны средства автоматизации формирования некоторых видов табличных документов. Следует отметить, что указанные средства разработаны для определенных систем проектирования.

Таким образом, выделим следующие особенности разработки средств автоматизации проектирования.

1. Не на все составные части изделия можно спроектировать трехмерную модель. Следовательно, часть структуры изделия, не созданная конструктором, может быть утеряна, что приведет к отсутствию информации об изделии и задержке на последующих этапах его жизненного цикла.

2. Как показывает практика, большая часть работы по формированию электронного описания изделия производится вручную конструктором, после оформления его документов. Это приводит к эффекту «двойной работы» и ошибкам, выявляющимся на этапе технологической подготовки производства.

3. Автоматизация определенных типов документов (спецификация, сводная спецификация, ведомость, перечень элементов) осуществлена для определенных систем. Это приводит к сложностям при передаче таких документов другим организациям, так как там невозможно прочитать переданные документы.

4. Разработка подобных средств, производится специалистами каждой организации отдельно. При этом возникают проблемы, описанные в п. 3.

Виды электронной структуры изделия

В зависимости от того, какие данные содержатся в структуре, выделяют несколько ее видов. Это функциональная, конструктивная, производственно-технологическая, физическая, эксплуатационная и совмещенная структуры [3].

Функциональная ЭСИ определяет назначение изделия и его составных частей и предъявляемые к ним функциональные требования. Данный тип ЭСИ разрабатывается на стадии технического предложения на изделие.

Конструктивная ЭСИ отображает конкретные технические решения на изделие: детали, сборочные единицы, комплекты и комплексы. ЭСИ разрабатывается на стадиях эскизного и технического проектов и рабочей конструкторской рабочей документации.

Производственно-технологическая ЭСИ отображает особенности технологии изготовления и сборки изделия. Структура формируется на основе конструктивной ЭСИ и содержит технологические составные части. Разрабатывается на этапах технологической подготовки производства и в процессе производства изделия.

Физическая ЭСИ содержит информацию о конкретном экземпляре изделия. Выполняется на стадии производства изделии и дополняет конструктивную ЭС в части составных частей изделия такими признаками, как заводской номер, серийный номер и др.

Эксплуатационная ЭСИ содержит данные о составных частях изделия, подлежащих обслуживанию и (или) замене в ход использования изделия. Является частью конструктивной ЭСИ и выполняется на стадии эскизного и технического проектов и рабочей конструкторской документации, уточняется в соответствии с физической ЭСИ.

Совмещенная ЭСИ отображает комплексную информацию об изделии и содержит отдельные разновидности ЭСИ. Содержит составные части изделия, рассматриваемые в различных аспектах, требующих набор разной информации.

Таким образом, в зависимости от типа содержащихся данных и стад жизненного цикла может быть построена структура изделия определенного типа. Они могут дополняться и изменяться в соответствии с изменением документации, содержащейся в них.

На основе рассмотренного выше материала предлагается следующий метод формирования структуры изделия.

Метод формирования структуры изделия

Совмещенная структура изделия может использоваться для анализа данных, содержащихся в структурах нескольких видов.

В процессе составления и подписания договора на разработку изделия должна формироваться функциональная структура изделия. Она содержит данные о составе изделия и тех требованиях, которые определены для каждой такой части и изделия в целом. Далее, при разработке схемотехнических данных, формируется конструктивная структура. В структуру вносятся данные об используемых покупных элементах и материалах. На основе этих данных из структуры автоматизированным способом предлагается получать перечень элементов.

Следующим этапом формирования такой структуры является внесение в нее конструкторских данных – сборочных чертежей и моделей. На основе полученной структуры предлагается автоматически получать спецификацию. Также данная структура может быть дополнена программными документами.

На основе этих данных формируется производственно-технологическая структура. В нее помещаются технологические документы и модели, необходимые для создания программ для станков с числовым программным управлением и сами эти программы. На данном этапе может изменяться и конструктивная структура.

Следующим этапом является формирование физической структуры. Эта структура представляет собой дополненную конструктивную структуру данными производства. На данном этапе может изменяться как конструктивная, так и производственно-технологическая структуры.

После сборки изделия формируется эксплуатационная структура. При этом конструктивная структура дополняется эксплуатационной документацией. На данном этапе могут автоматически формироваться ведомости и описи.

В процессе эксплуатации документам и структурам могут присваиваться литеры при внесении в них изменений.

Заключение

В статье описан метод последовательного формирования электронного описания изделия. Предложенный метод позволяет формировать информацию об изделии на всех этапах жизненного цикла изделия и автоматически формировать текстовые табличные документы. При этом уменьшается количество допускаемых в документах ошибок и время разработки документации, что позволяет повысить качество продукции.

Литература

1. Норенков И.П., Кузьмик П.К. Информационная поддержка наукоемких изделий. CALS-технологии. – М.: Издательство МГТУ им. Н.Э. Баумана, 2002. – 302 с.
2. Яцкевич А., Страузов Д. Построение интегрированной информационной среды предприятия на основе системы управления данными об изделии PDM STEP SUITE // САПР и графика. – 2002. – №6.
3. ГОСТ 2.053-2006 Электронная структура изделия.

ПОВЫШЕНИЕ ТОЧНОСТИ ЛОКАЛИЗАЦИИ НЕОДНОРОДНОСТЕЙ ВОЛН ПРИ ИСПОЛЬЗОВАНИИ КОРРЕЛЯЦИОННЫХ РЕФЛЕКТОМЕТРОВ

Т.С. Беляева (Санкт-Петербургский государственный университет телекоммуникаций им. профессора М.А. Бонч-Бруевича)
Научный руководитель – к.т.н., профессор С.Ф. Глаголев (Санкт-Петербургский государственный университет телекоммуникаций им. профессора М.А. Бонч-Бруевича)

В статье рассмотрен принцип работы корреляционного рефлектометра, особенности выбора зондирующего сигнала, рассмотрены возможности повышения точности определения неоднородностей и сокращения мёртвой зоны до минимума при использовании корректирующих кодов.

Введение

Среди многих средств измерения параметров волоконных трактов особую роль играют оптические рефлектометры, работающие по принципу измерения обратного рассеяния [1–4]. Оптические рефлектометры позволяют определять длину и затухание оптических кабелей, параметры локальных неоднородностей, возвратные и вносимые потери, коэффициенты отражения в разъемных соединениях оптических волокон, расстояние до оптических муфт, разъемов и точек повреждения оптического кабеля.

Для обеспечения высокой точности измерений каждого из перечисленных параметров необходимо варьировать длительностью зондирующего оптического импульса рефлектометра, полосой пропускания измерительного тракта и временем накопления сигнала обратного рассеяния [5]. Так как для измерения параметров фрагментов волоконного тракта, расположенных на различном расстоянии от его начала, необходимо устанавливать различные длительности зондирующих импульсов, различную полосу пропускания и различное время накопления, то при использовании традиционных рефлектометров точное измерение всех параметров возможно лишь при проведении ряда независимых измерений. Это приводит к значительным затратам времени, что не всегда допустимо.

Сокращение времени, и повышение точности измерений может достигаться за счет увеличения мощности источника излучения, формирующего зондирующий импульс, уменьшения оптических потерь в измерительном тракте, увеличения чувствительности фотоприемника и уменьшения его шумов. Максимально допустимая мощность излучения составляет сотни милливатт и ограничивается нелинейными эффектами в одномодовом оптическом волокне. Выпускаемые в настоящее время полупроводниковые лазеры практически подошли к этому пределу. Оптические потери могут быть уменьшены на 3–6 дБ заменой ответвителей на оптические переключатели на основе кольцевых циркуляторов. Чувствительность фотоприемников практически достигла своего предельного значения, а уменьшение шумов возможно только благодаря снижению их рабочей температуры. Более эффективное использование рассмотренной элементной базы позволяет несколько уменьшить время измерений, но не обеспечивает его кардинального сокращения.

Одним из наиболее перспективных путей уменьшения времени измерения, повышения точности измерения и расширения динамического диапазона оптических рефлектометров при сохранении высокого пространственного разрешения является использование сложных псевдослучайных, зондирующих сигналов с большой энергией [6, 7]. Под сложным сигналом в этих случаях понимается пачка импульсов одинаковой формы, полярность которых изменяется по закону кодовой последовательности.

Рефлектометр со сложным зондирующим сигналом

При классическом рассмотрении свойств псевдослучайных сигналов предполагается, что они содержат импульсы с положительной амплитудой, с отрицательной амплитудой и отсутствие импульсов за пределами пачки. Такое математическое представление зондирующего сигнала необходимо для более удобного рассмотрения способов формирования зондирующего сигнала и регистрации сигнала обратного рассеяния. Однако лазерный источник излучения не может формировать импульсы с положительной и отрицательной интенсивностью, поэтому сначала формируется пачка импульсов, амплитуда которых пропорциональна амплитуде положительных импульсов зондирующего сигнала, затем формируется инверсная пачка импульсов, амплитуда которых пропорциональна амплитуде отрицательных импульсов зондирующего сигнала, но имеет обратный знак. Эти пачки импульсов поочередно поступают на лазерный источник, формирующий две пачки импульсов излучения с положительной интенсивностью, а принятые от этих пачек сигналы обратного рассеяния суммируются: значения принятые от первой пачки, со знаком плюс, а от второй – со знаком минус [8].

В традиционной оптической рефлектометрии в качестве зондирующего сигнала используется одиночный импульс. Получаемая при этом рефлектограмма, представляет собой отклик волоконного тракта на одиночный импульс. Зарегистрированная рефлектограмма сравнительно легко интерпретируется, позволяя получить информацию о распределении затухания вдоль волоконно-оптического тракта, наличии и расположении различных неоднородностей.

В корреляционных рефлектометрах со сложным зондирующим сигналом для получения формы традиционной рефлектограммы в волоконно-оптический тракт вводится сложный зондирующий оптический сигнал, принимается и регистрируется отраженное от неоднородностей и рассеянное излучение, а затем вычисляется взаимно корреляционная функция принятого и опорного сигналов с помощью коррелятора. Эта функция описывает отклик коррелятора на принятый сигнал, поэтому взаимно корреляционная функция зондирующего и опорного сигналов является важной характеристикой метода измерения.

При использовании в качестве зондирующего сигнала кодовых последовательностей, представляющих собой фрагмент случайной последовательности, относительная погрешность регистрации рефлектограммы будет недопустимо велика. Причина возникновения этой погрешности обусловлена наличием больших боковых лепестков у автокорреляционной функции кодовой последовательности зондирующего сигнала. Для создания рефлектометра, обладающего высокими техническими характеристиками, необходимо использовать кодовые последовательности, корреляционные функции которых имеют основной лепесток максимальной амплитуды, боковые лепестки минимальной амплитуды (либо вообще их не имеют), и согласованные с ними способы обработки сигналов обратного рассеяния, использующие малые объемы памяти и простые алгоритмы обработки.

Наибольшее распространение для формирования сложных зондирующих сигналов оптических рефлектометров нашли комплементарные последовательности Голея [9, 10], которые представляют собой пару кодовых последовательностей одинаковой длины. Две пачки прямоугольных импульсов, полярности которых изменяются по закону одной из этих последовательностей, изображены на (рис. 1, а, б).

Автокорреляционная функция каждой пачки импульсов (рис. 1, в, г) имеет основной лепесток максимальной амплитуды и значительные боковые лепестки. Расположенные аналогично друг другу боковые лепестки разных последовательностей имеют одинаковую амплитуду, но противоположный знак. В результате суммарная автокорре-

ляционная функция (рис. 1, д) имеет удвоенный основной лепесток и не имеет боковых лепестков.

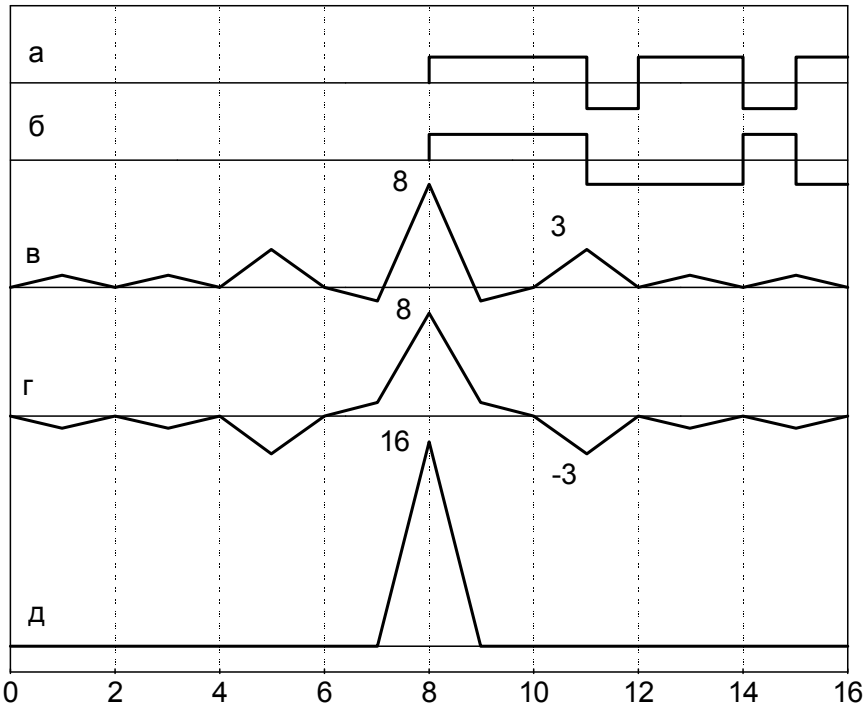


Рис. 1. Автокорреляционные функции комплементарных последовательностей Голея

Выражения для пары комплементарных кодовых последовательностей

$A_K = \{a(k)\}, \quad k = 1,2,\cdots K$, $\qquad B_K = \{b(k)\}, \quad k = 1,2,\cdots K$,

где K – количество элементов кодовой последовательности.

Дискретные автокорреляционные функции этих последовательностей

$$V_A = \{v_A(u)\} = \left\{ \sum_{k=1}^{K} a(k)a(k+u) \right\}, \quad u = (1-K),(2-K),\cdots(K-1),$$

$$V_B = \{v_B(u)\} = \left\{ \sum_{k=1}^{K} b(k)b(k+u) \right\}, \quad u = (1-K),(2-K),\cdots(K-1).$$

Суммарная дискретная автокорреляционная функция

$$V_{AB} = \{v_A(u) + v_B(u)\} = \left\{ \sum_{k=1}^{K} a(k)a(k+u) + b(k)b(k+u) \right\} \quad ,$$

$$u = (1-K),(2-K),\cdots(K-1),$$

где u – фазовый сдвиг.

Характеристики комплементарных последовательностей Голея идеально подходят для оптических рефлектометров со сложным зондирующим сигналом. Суммарная автокорреляционная функция этих последовательностей не имеет боковых лепестков, а стационарная форма зондирующего сигнала предусматривает его многократное повторение, что позволяет суммировать и накапливать в одних и тех же ячейках памяти результаты разных измерений одних и тех же мгновенных значений сигнала обратного рассеяния. Существенным достоинством этих последовательностей является возможность формировать последовательности неограниченной длины, которая реализуется при использовании правил «присоединения» или «чередования» [11].

При исследовании волоконно-оптических трактов оптическими рефлектометрами со сложным сигналом процесс измерения состоит из трех частей: генерирование зондирующего сигнала, интервала ожидания и регистрация сигнала обратного рассеяния. На рис. 2, а, изображен генерируемый на интервале $(t_З - t_0)$ зондирующий сигнал, а на рис. 2, б – сигнал обратного рассеяния, сформированный этим зондирующим сигналом. Видно, что амплитуда сигнала обратного рассеяния во время генерации зондирующего сигнала и в течение некоторого времени после ее окончания имеет величину, значительно превышающую значение сигнала в конце регистрации. Это объясняется отражением потока излучения от торца оптического тракта и мощным сигналом обратного рассеяния его ближней зоны. Такая разница в амплитудах приводит к насыщению фотоприемного устройства и усилителей фототока во время генерации и в течение некоторого времени после ее окончания, что не позволяет в это время проводить регистрацию сигнала обратного рассеяния. Поэтому регистрация (рис. 2, в) начинается через некоторый интервал ожидания $L = (t_P - t_З)$ после окончания генерации и заканчивается в выбранный момент времени, определяемый шкалой расстояний $(t_K - t_P)$.

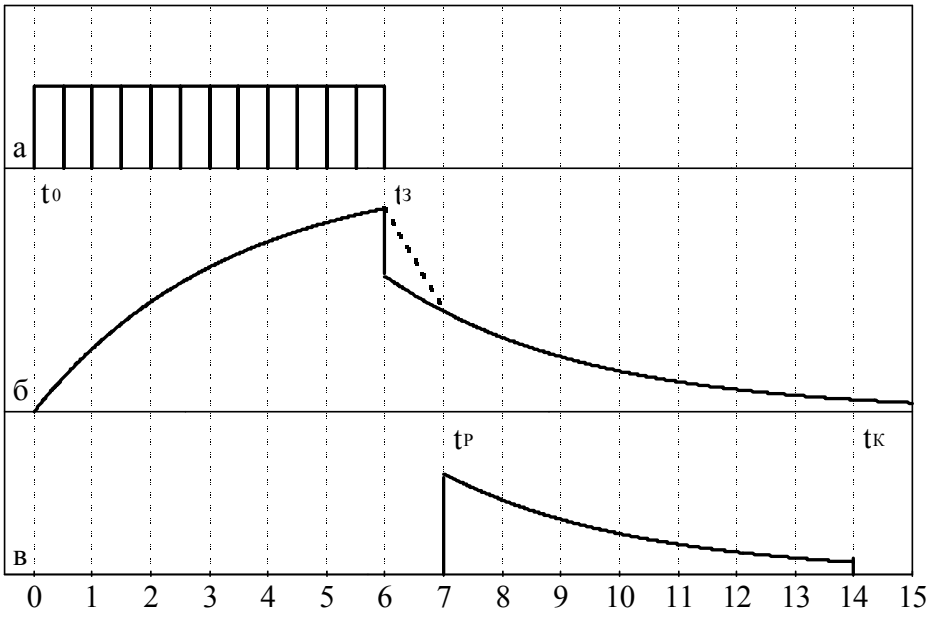


Рис. 2. Процесс измерения рефлектометром со сложным зондирующим сигналом

Длительность зондирующей пачки $(t_З - t_0)$ выбирается исходя из расстояния до исследуемого участка волоконного тракта, необходимого соотношения сигнал/шум и интервала ожидания. Интервал ожидания $(t_P - t_З)$ определяется техническими характеристиками фотоприемного устройства и усилителей фототока и может меняться при использовании фотоприемника и усилителей фототока с переключаемыми коэффициентами передачи.

Интервал времени $(t_K - t_P)$, в течение которого регистрируется сигнал, больше или в пределе равен длительности пачки зондирующего сигнала и определяется расстоянием до исследуемого участка оптического тракта, его длиной, протяженностью исследуемого участка и реально существующим отношением сигнал-шум. После обработки зарегистрированного сигнала корреляционными методами получается рефлектограмма участка волоконно-оптического тракта, соответствующего интервалу $(t_K - t_З - t_P)$. От каждого элемента этого участка принимается сигнал обратного рассеяния длительностью, равной длительности пачки зондирующего сигнала. Ближняя

точка этого участка отстоит от начала тракта на величину мертвой зоны, определяемой длительностью пачки зондирующего сигнала и интервалом ожидания.

Отношение сигнала к шуму и динамический диапазон корреляционного рефлектометра увеличивается в $\sqrt{K}/2$ раза по сравнению с традиционным рефлектометром с такими же параметрами.

Для увеличения отношения сигнал/шум необходимо увеличивать длительность зондирующей пачки, однако это приводит к сокращению протяженности исследуемого участка и увеличению мертвой зоны, а желание увеличить протяженность участка ведет к необходимости сокращения длительности зондирующей пачки и, следовательно, уменьшению отношения сигнал/шум.

От ряда элементов волоконного тракта, находящихся ближе зарегистрированного участка, принимаются усеченные сигналы обратного рассеяния, длительность которых короче длительности зондирующей пачки. Если эти сигналы обработать корреляционными методами, то полученный фрагмент рефлектограммы от этих элементов будет искажен. Это объясняется появлением боковых лепестков у суммарной автокорреляционной функции из-за усечения сигналов обратного рассеяния.

Для увеличения протяженности исследуемого участка и сокращения мертвой зоны могут использоваться наборы кодовых последовательностей Голея. Для этого из пары исходных комплементарных последовательностей Голея с помощью однократного или многократного использования правила присоединения формируется полный набор возможных кодовых последовательностей, из которых формируется набор зондирующих сигналов, регистрируется набор сигналов обратного рассеяния и вычисляется сумма взаимно корреляционных функций каждого зарегистрированного сигнала и соответствующей исходной кодовой последовательности. Для каждого зондирующего сигнала выбирается исходная кодовая последовательность, совпадающая с его последним фрагментом. Однако при таком способе регистрации рефлектограммы приходится измерять, регистрировать и обрабатывать большое число сигналов обратного рассеяния волоконно-оптических трактов. В реальных условиях это весьма затруднительно, так как требует больших объемов памяти оптических рефлектометров и больших затрат времени на обработку зарегистрированных сигналов обратного рассеяния.

Для уменьшения потерь времени зондирующий сигнал может модулироваться одной парой комплементарных последовательностей Голея, одна из которых имеет обратный порядок следования символов. Выражения для такой пары последовательностей при однократном использовании правила присоединения:

$$E_{AB} = \{a(k)|b(k)\} = \{a(k), b(k-K/2)\}, \quad k = 1,2,\cdots K,$$

$$F^0_{AB} = \{-b^0(k)|a^0(k)\} = \{-b(K/2-k+1), a(K-k+1)\}, \quad k = 1,2,\cdots K.$$

Так как автокорреляционные функции последовательностей симметричны относительно основного лепестка, то эти последовательности тоже будут комплементарными.

Выражения для зарегистрированных сигналов обратного рассеяния при использовании этих последовательностей с учетом интервалов ожидания:

$$Z_{EAB} = \{z_{EAB}(j)\} = \left\{ \sum_{k=1}^{K/2} a(k)x(j-k) + \sum_{k=K/2+1}^{K} b(k-K/2)x(j-k) \right\},$$

$$j = (1+K+L),(2+K+L),\cdots(R+K),$$

$$Z^0_{FAB} = \{z^0_{FAB}(j)\} = \left\{ -\sum_{k=1}^{K/2} b(K/2-k+1)x(j-k) + \sum_{k=K/2+1}^{K} a(K-k+1)x(j-k) \right\},$$

$j = (1+K+L),(2+K+L),\cdots(R+K)$, где $R = t_K$.

Для корреляционной обработки зарегистрированных сигналов используются опорные сигналы в виде последних фрагментов кодовых последовательностей зондирующих сигналов:

$$B_{K/2} = \{b(n - K/2)\}, \quad n = (1 + K/2), (2 + K/2), \cdots K,$$

$$A^0_{K/2} = \{a(K - n + 1)\}, \quad n = (1 + K/2), (2 + K/2), \cdots K.$$

Взаимно корреляционные функции зарегистрированных сигналов обратного рассеяния и их опорных сигналов:

$$W_{EB} = \left\{ \sum_{l=1-K/2}^{K/2-1} \sum_{k=1}^{K/2} \left[a(k)b(k+l)x(u+l+K/2) + b(k)b(k+l)x(u+l) \right] \right\},$$

$$u = (1 + L), (2 + L) \cdots (R + K/2 - 1),$$

$$W^0_{FA} = \left\{ \sum_{l=1-K/2}^{K/2-1} \sum_{k=1}^{K/2} \left[-b(k)a(k+l)x(u+l+K/2) + a(k)a(k+l)x(u+l) \right] \right\},$$

$$u = (1 + L), (2 + L) \cdots (R + K/2 - 1).$$

Суммарная взаимно корреляционная функция:

$$W_{EF} = W_{EB} + W^0_{FA} = \left\{ \sum_{l=1-K/2}^{K/2-1} \sum_{k=1}^{K/2} \left[a(k)a(k+l) + b(k)b(k+l) \right] x(u+l) \right\},$$

$$u = (1 + L), (2 + L), \cdots (R + K/2 - 1).$$

Это выражение является функцией фазового сдвига u между зондирующими и опорными сигналами и описывает мгновенные значения рефлектограммы в точках, определяемых этим сдвигом.

Интервал изменений фазового сдвига, которым определяется протяженность исследуемого участка оптического кабеля, лежит в пределах $(K/2 + L) \le u \le (K + L - 1)$.

Окончательное выражение для зарегистрированной рефлектограммы:

$$W_I = K\{x(u)\}, \quad u = (K/2 + L), (K/2 + L + 1), \cdots (K + L - 1).$$

Дальнейшее увеличение протяженности исследуемого участка оптического кабеля и уменьшение мертвой зоны может быть достигнуто, если для их формирования несколько раз воспользоваться правилом присоединения.

Выражения для такой пары последовательностей при двукратном использовании правила присоединения:

$$E_{AB} = \left\{ a(k) \middle| b(k) \middle| a(k) \middle| -b(k) \right\} = \{a(k), b(k - K/4), a(k - K/2), -b(k - 3K/4)\},$$

$$k = 1,2, \cdots K,$$

$$F^0_{AB} = \left\{ b^0(k) \middle| -a^0(k) \middle| b^0(k) \middle| a^0(k) \right\} =$$

$$= \{b(K/4 - k + 1), -a(K/2 - k + 1), b(3K/4 - k + 1), a(K - k + 1)\},$$

$$k = 1,2, \cdots K.$$

Зарегистрированные сигналы обратного рассеяния при использовании этих последовательностей с учетом интервалов ожидания:

$$Z_{EAB} = \left\{ \begin{array}{l} \sum_{k=1}^{K/4} a(k)x(j-k) + \sum_{k=K/4+1}^{K/2} b(k - K/4)x(j-k) + \\ + \sum_{k=K/2+1}^{3K/4} a(k - K/2)x(j-k) - \sum_{k=3K/4+1}^{K} b(k - 3K/4)x(j-k) \end{array} \right\},$$

$$j = (1 + K + L), (2 + K + L), \cdots (R + K),$$

$$Z^0_{FAB}=\left\{\begin{array}{l}\sum\limits_{k=1}^{K/4}b(K/4-k+1)x(j-k)-\sum\limits_{k=K/4+1}^{K/2}a(K/2-k+1)x(j-k)+\\+\sum\limits_{k=K/2+1}^{3K/4}b(3K/4-k+1)x(j-k)+\sum\limits_{k=3K/4+1}^{K}a(K-k+1)x(j-k)\end{array}\right\},$$

$j=(1+K+L),(2+K+L),\cdots(R+K)$.

Если для корреляционной обработки зарегистрированных сигналов используется опорные сигналы в виде последних фрагментов соответствующих последовательностей зондирующих сигналов,

$B_{K/4}=\{-b(n-3K/4)\},\quad n=(1+3K/4),(2+3K/4),\cdots K$,

$A^0_{K/4}=\{a(K-n+1)\},\quad n=(1+3K/4),(2+3K/4),\cdots K$,

то взаимно корреляционные функции зарегистрированных сигналов обратного рассеяния и их опорных сигналов равны

$$W_{EB}=\left\{\begin{array}{l}-\sum\limits_{n=1+3K/4}^{K}\sum\limits_{k=1}^{K/4}a(k)b(n-3K/4)x(u+n-k)-\\-\sum\limits_{n=1+3K/4}^{K}\sum\limits_{k=1+K/4}^{K/2}b(k-K/4)b(n-3K/4)x(u+n-k)-\\-\sum\limits_{n=1+3K/4}^{K}\sum\limits_{k=1+K/2}^{3K/4}a(k-K/2)b(n-3K/4)x(u+n-k)+\\+\sum\limits_{n=1+3K/4}^{K}\sum\limits_{k=1+3K/4}^{K}b(k-3K/4)b(n-3K/4)x(u+n-k)\end{array}\right\}=$$

$$=\left\{\sum_{l=1-K/4}^{K/4-1}\sum_{k=1}^{K/4}\left[\begin{array}{l}-a(k)b(k+l)x(u+l+3K/4)-b(k)b(k+l)x(u+l+K/2)-\\-a(k)b(k+l)x(u+l+K/4)+b(k)b(k+l)x(u+l)\end{array}\right]\right\},$$

$u=(1+L),(2+L)\cdots(R+K/4-1)$,

$$W^0_{FA}=\left\{\begin{array}{l}\sum\limits_{n=1+3K/4}^{K}\sum\limits_{k=1}^{K/4}b(K/4-k+1)a(K-n+1)x(u+n-k)-\\-\sum\limits_{n=1+3K/4}^{K}\sum\limits_{k=1+K/4}^{K/2}a(K/2-k+1)a(K-n+1)x(u+n-k)+\\+\sum\limits_{n=1+3K/4}^{K}\sum\limits_{k=1+K/2}^{3K/4}b(3K/4-k+1)a(K-n+1)x(u+n-k)+\\+\sum\limits_{n=1+3K/4}^{K}\sum\limits_{k=1+3K/4}^{K}a(K-k+1)a(K-n+1)x(u+n-k)\end{array}\right\}=$$

$$=\left\{\sum_{l=1-K/4}^{K/4-1}\sum_{k=1}^{K/4}\left[\begin{array}{l}a(k)b(k+l)x(u+l+3K/4)-a(k)a(k+l)x(u+l+K/2)+\\+a(k)b(k+l)x(u+l+K/4)+a(k)a(k+l)x(u+l)\end{array}\right]\right\},$$

$u=(1+L),(2+L)\cdots(R+K/4-1)$.

Суммарная взаимно-корреляционная функция имеет вид

$$W_{EF} = W_{EB} + W_{FA}^{0} = \left\{ \begin{array}{l} \sum\limits_{l=1-K/4}^{K/4-1} \sum\limits_{k=1}^{K/4} \left[a(k)a(k+l) + b(k)b(k+l)\right] x(u+l) - \\ - \sum\limits_{l=1-K/4}^{K/4-1} \sum\limits_{k=1}^{K/4} \left[a(k)a(k+l) + b(k)b(k+l)\right] x(u+l+K/2) \end{array} \right\},$$

$u = (1+L), (2+L), \cdots (R+K/2-1)$.

Это выражение является функцией фазового сдвига u между зондирующими и опорными сигналами и описывает мгновенные значения рефлектограммы в точках, определяемых этим сдвигом.

Интервал изменений фазового сдвига, которым определяется протяженность исследуемого участка волоконного тракта, лежит в пределах $(K/4+L) \le u \le (K/2+L-1)$.

Окончательное выражение для зарегистрированной рефлектограммы:

$$W_{II} = \frac{K}{2}\{x(u) - x(u+K/2)\}, \quad u = (K/4+L), (K/4+L+1), \cdots (K/2+L-1),$$

Из выражения видно, что рефлектограмма искажена из-за присутствия второго члена. Величина $x(u+K/2)$ в большинстве случаев меньше величины $x(u)$, потому что описывает обратное рассеяние более удаленных элементов оптического кабеля, и в зависимости от отношения этих величин искажениями рефлектограммы иногда можно пренебречь. Кроме того, можно осуществить коррекцию зарегистрированного фрагмента рефлектограммы, так как фрагмент рефлектограммы, на котором расположена величина $x(u+K/2)$, зарегистрирован и обработан ранее, при использовании опорных сигналов большей протяженности.

Корректирующие коды

В зависимости от кратности применения правила присоединения при формировании зондирующего сигнала и протяженности используемых опорных сигналов искажения рефлектограммы будут различными. Например, если три раза воспользоваться правилом присоединения, то пара комплементарных последовательностей имеет вид

$$E_{AB} = \left\{ a(k) \left| b(k) \right| a(k) \left| -b(k) \right| a(k) \left| b(k) \right| -a(k) \left| b(k) \right. \right\},$$

$$F_{AB}^{0} = \left\{ -b^{0}(k) \left| a^{0}(k) \right| -b^{0}(k) \left| -a^{0}(k) \right| -b^{0}(k) \left| a^{0}(k) \right| b^{0}(k) \left| a^{0}(k) \right. \right\}.$$

Если используются опорные сигналы в виде $\{b(k)\}$, $\{a^{0}(k)\}$, то зарегистрированный фрагмент рефлектограммы будет

$$W_{III} = \frac{K}{4}\{x(u) + x(u+K/4) - x(u+K/2) + x(u+3K/4)\},$$

$u = (K/8+L), (K/8+L+1), \cdots (K/4+L-1)$.

Для определения алгоритма проведения коррекции удобно пользоваться корректирующими кодами, которые имеют вид последовательности

$$C_V = \{c(m)\}, \quad m = 1,2, \cdots 2^V.$$

Корректирующий код может быть получен последовательным сравнением опорного сигнала с фрагментами той же длительности, соответствующей кодовой последовательности зондирующего сигнала. Если фрагмент последовательности зондирующего сигнала совпадает с опорным сигналом, записывается (+1), если не совпадает, записы-

вается (0), а если все элементы фрагмента последовательности зондирующего и опорного сигнала имеют противоположный знак, то записывается (-1). Для сравнения может использоваться любая из комплементарных кодовых последовательностей зондирующего сигнала, а сравнения проводится справа налево. Рассмотренная пара кодовых последовательностей при использовании опорных сигналов $\{b(k)\}$ и $\{a^0(k)\}$ имеет корректирующий код [1,0,1,0,-1,0,1,0].

Зарегистрированный фрагмент рефлектограммы, записанный с помощью корректирующего кода, имеет вид

$$W_V = 2^{1-V} K \sum_{m=1}^{2^V} c(m) x\left[u \pm K(m-1)/2^V\right].$$

В таблице приведены выражения, по которым может быть скорректирована форма различных фрагментов зарегистрированной рефлектограммы при зондирующем сигнале, модулированном парой последовательностей, сформированных четырехкратным использованием правила присоединения.

Регистрация и обработка сигналов обратного рассеяния при модуляции зондирующего сигнала парой комплементарных последовательностей Голея, одна из которых имеет обратный порядок следования символов, и коррекции рефлектограммы позволяет применять простые зондирующие сигналы, эффективно использовать компенсационный накопитель, расширить протяженность исследуемого участка волоконно-оптического тракта и максимально сократить мертвую зону оптического рефлектометра.

V	Коррект. код	Выражения для корректировки фрагментов рефлектограммы	Интервалы фрагментов
0	1	$x(u) = W_0 / 2K$	$(K+L)\cdots R$
1	1,0	$x(u) = W_1 / K$	$(K/2+L)\cdots(K+L-1)$
2	1,0,-1,0	$x(u) = 2W_{II} / K + x(u+K/2)$	$(K/4+L)\cdots K/2+L-1)$
3	1,0,1,0, -1,0,1,0	$x(u) = 4W_{III} / K - x(u+K/4) +$ $+ x(u+K/2) - x(u+3K/4)$	$(K/8+L)\cdots(K/4+L-1)$
4	1,0,1,0, 1,0,-1,0, -1,0,1,0, 1,0,-1,0	$x(u) = 8W_{III} / K - x(u+K/8) -$ $- x(u+K/4) + x(u+3K/8) +$ $+ x(u+K/2) + x(u+5K/8) -$ $- x(u+3K/4) + x(u+7K/8)$	$(K/16+L)\cdots(K/8+L-1)$

Таблица. Выражения для корректировки рефлектограммы

Заключение

Применение сложных сигналов в оптической рефлектометрии дает возможность увеличения протяженности исследуемого участка ВОЛС, уменьшения времени регистрации рефлектограммы , повышения точности измерения и расширения динамического диапазона оптических рефлектометров при сохранении высокого пространственного разрешения. Модуляция зондирующего сигнала парой комплементарных последовательностей Голея, одна из которых имеет обратный порядок следования символов, и коррекция рефлектограммы позволяет применять простые зондирующие сигналы и максимально сократить мертвую зону оптического рефлектометра.

Литература

1. Былина М.С., Глаголев С.Ф., Кочановский Л.Н., Пискунов В.В. Измерение параметров волоконно-оптических линейных трактов. Учебное пособие. – СПб: СПб ГУТ, 2002. – 68 с.
2. Измерения на ВОЛП методом обратного рассеяния. Учебное пособие для ВУЗов / Андреев В.А., Бурдин В.А., Баскаков В.С., Косова А.Л. – Самара: СРТТЦ ПГАТИ, 2000. – 107 с.: ил.
3. Свинцов А.Г. Рефлектометрические методы измерения параметров ВОЛС // Метрология и измерительная техника связи. – 2002. – № 5. – С. 64–65.
4. Хрычев В.Е. Разработка и исследование методов и приборов оптической рефлектометрии во временной области. Дис. ... канд. тех. наук. / Санкт-Петербургский государственный ун-т телекоммуникаций им. проф. М.А. Бонч-Бруевича. – СПб, 1998. – 135 с.
5. Архангельский В.Б., Глаголев С.Ф., Семин А.В. Методические особенности регистрации и обработки сигналов обратного рассеяния / Тезисы докладов 14 научно-технической конференции «Фотометрия и ее метрологическое обеспечение», М, 2004. С. 53.
6. Финкельштейн М.И. Основы радиолокации. Учебник для вузов. 2-е изд., перераб. и доп. – М.: Радио и связь, 1983. – 536 с., ил.
7. Варакин Л.Е. Системы связи с шумоподобными сигналами. – М.: Радио и связь, 1985. – 384 с., ил.
8. Семин А.В., Архангельский В.Б., Глаголев С.Ф. Патент на полезную модель № 37209 «Оптический корреляционный рефлектометр» от 10.04.2004 г. бюл. №10 по заявке 2003137925/20 от 18.12.2003.
9. Newton S. A new technique in OTDR // Electron. and Wireless World. – 1988. – V. 94. – № 627. – P.496–500.
10. Jones M. Using simplex codes to improve OTDR sensitivity .// IEEE Photon. Technol. Lett. – 1993. – V.5. – № 7. – P.822–824.
11. Семин А.В., Архангельский В.Б. Способы формирования сложных зондирующих сигналов для оптических рефлектометров / Труды учебных заведений связи. – СПб ГУТ. – СПб, 2003. – № 169. – С. 200–213.

ОПЫТ И ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ КОНТРОЛЛЕРОВ СЕМЕЙСТВА SDK

Д.Р. Ковязина, Е.В. Петров
Научный руководитель – к.т.н., доцент А.Е. Платунов

В статье обсуждаются опыт и перспективы применения контроллеров семейства SDK в учебном процессе, организованном согласно предлагаемой модели обучения, для подготовки специалиста в области микропроцессорных систем.

Введение

На сегодняшний день точка зрения о том, что эффективно разрабатывать микропроцессорные системы способен «интегральный» специалист, владеющий аппаратными и программными аспектами вычислительной техники [1], получает все больше подтверждений. Однако не все программы обучения специалистов в этой области отвечают потребностям рынка. В данной статье предлагается модель подготовки такого «интегрального» специалиста.

Одним из основных принципов предлагаемой модели обучения основам организации микропроцессорных систем (МПС) является параллельное освоение различных микропроцессорных ядер, периферийных блоков, инструментальных средств и средств проектирования [2].

Для эффективного освоения материала в рамках предложенной модели необходимы микропроцессорные стенды и инструментальные средства. В качестве стендового оборудования обычно используются либо платы развития (evaluation board), наборы прототипирования (development kit, starter kit), предоставляемые фирмами-производителями микропроцессорных компонентов для демонстрации их работы [3], либо специально созданные для учебного процесса учебно-лабораторные стенды [1]. Однако наборы прототипирования из первой группы в большинстве случаев не очень приспособлены для учебных целей, а учебные стенды предлагаются в ограниченной номенклатуре. В последнем случае исключением является семейство микропроцессорных стендов учебного назначения SDK, разрабатываемое фирмой ООО «ЛМТ» [1, 4, 5].

В данной статье обсуждаются опыт и перспективы применения данного семейства стендов в учебном процессе, организованном согласно предлагаемой модели обучения. Предлагаемая модель обучения находится на стадии формализации и развития в рамках научно-образовательного направления «Встроенные вычислительные системы» кафедры вычислительной техники СПбГУ ИТМО [6].

Модель обучения принципам организации МПС

Модель обучения включает три уровня:

1. базовые элементы и основы программирования микропроцессорных систем;
2. принципы организации и программирования микропроцессорных систем;
3. основы проектирования микропроцессорных систем.

Первый уровень предназначен для обучения студентов по направлению «Информатика и вычислительная техника» и смежным с ним направлениям на 3–4 курсе (до деления на специализации). Второй уровень может быть использован для профильной подготовки бакалавров по специализации «Информационно-управляющие системы» и смежным с ней специализациям. Третий уровень может быть использован в процессе подготовки магистров по специализации «Информационно-управляющие системы» и смежным с ней специализациям.

Цель – формирование целостного сквозного представления и глубокого понимания организации МПС.

Предлагаемая модель обучения имеет практический характер и представляет собой спектр тем, которые рассматриваются на каждом уровне и могут быть освоены с применением учебно-лабораторного комплекса SDK. Ее можно сравнить с моделью обучения какому-либо иностранному языку: первый уровень – алфавит, слоги, слова; второй уровень – предложения и правила их составления; третий уровень – рассказы, изложения на заданную тему. Как в случае с изучением иностранного языка необходимо знать родной, так и обучение по данной модели требует определенного уровня начальной подготовки (дискретная математика, основы схемотехники, программирование на языке Си, организация ЭВМ и др.).

Базовые элементы и основы программирования МПС

Основная тема – структурно-функциональная организация МПС. Сюда входят следующие разделы:

- знакомство с основными элементами современных МПС: вычислительное ядро (микроконтроллер), периферийные устройства и интерфейсы связи с ними;
- разработка драйверов периферийных устройств и программ с их использованием для решения простых прикладных задач;
- знакомство с инструментальными средствами и технологической цепочкой программирования МПС.

Принципы организации и программирования МПС

Основная тема – архитектурные особенности и технологии программирования автономных и распределенных МПС. Основным отличием данного уровня от первого является выделение двух направлений обучения: МПС на базе микроконтроллеров и МПС на базе ПЛИС. По сравнению с первым уровнем осуществляется более глубокое изучение понятий архитектуры и программирования МПС посредством увеличения количества аспектов представления таких систем (надежностный, энергетический и др.). Кроме того, уделяется внимание методам отладки и тестирования (верификации) МПС, коммуникационным протоколам, технологиям проводной и беспроводной связи. Характерным для данного уровня является знакомство со стандартными решениями, готовыми компонентами и формирование навыков их эффективного использования.

Основы проектирования МПС

Основная тема – разработка МПС с учетом совокупности аспектов проектирования, в которой стенды SDK являются технической базой или полигоном для апробации своей системы. Основным отличием данного уровня является использование стандартных решений, технологий программирования и готовых компонентов, знания о которых были получены на предыдущих уровнях, в комплексе с самостоятельно разработанными компонентами системы (вычислительными, инструментальными, коммуникационными).

Совершенствование навыков разработки микропроцессорных систем и повышение уровня абстракции их представления может продолжаться и после освоения программы перечисленных уровней, так же как нет предела совершенствованию знаний иностранного языка.

Семейство учебных контроллеров SDK

Учебные стенды SDK (контроллеры), разработанные ООО «ЛМТ» [4, 5], представляют собой функционально законченные устройства, в которых есть вычислительное ядро, набор устройств ввода-вывода и средств общения с оператором, учебно-инструментальное программное обеспечение. Контроллеры могут работать с инструментальной ЭВМ (персональный компьютер) или в автономном режиме. Стенды выполнены в едином конструктиве, допускающем настольное использование или монтаж на DIN-рейку. Стенды комплектуются сетевыми блоками питания, инструментальными кабелями, пользовательской, учебно-методической и технической документацией, демонстрационными и инструментальными программами [1]. В семейство входит 5 стендов. В статье рассматриваются 3 модели семейства SDK, состав и основные характеристики которых представлены в таблице.

	SDK–1.1	SDK–2.0	SDK–6.1
Тип МК/ПЛИС	ADuC812 (Analog Devices)	LPC2292 (Philips)	EP1C3 (Altera)
Внутренняя память	Flash 8Кб, EEPROM 640 байт, RAM 256 байт	Flash 256 Кб, RAM 16 Кб	–
Внешняя память	RAM до 512 Кб, EEPROM до 32 Кб	RAM 512 Кб	Flash 1 Мб, EEPROM 256 байт, RAM 512 Кб
Дискретные входы/выходы	20	20	24
АЦП	8	8	–
ЦАП	2	2	–
Дисплей	Симв. ЖКИ 16x2	Граф. ЖКИ 122x32	Симв. ЖКИ 16x2
Клавиатура	Матричная 4×4	Матричная 4×4	8 + 1
Светодиоды	8	8	9
Часы реального времени	PCF8583	В составе системного супервизора FM3104 (Ramtron)	–
Интерфейсы	RS232C, JTAG	RS232C, RS485, JTAG, CAN (2 шт.), Ethernet, IEEE 802.15.4/ZigBee	RS232C, JTAG

Таблица. Основные представители семейства стендов SDK

В основу архитектуры стендов легли разработки систем промышленной автоматики. Предусмотрены стабилизатор и супервизор питания, схема сброса, сторожевой таймер, энергонезависимая память на базе EEPROM/FRAM и CMOS (RTC). Интерфейс RS232C имеет гальваническую изоляцию, что позволяет подключать и отключать стенды «на ходу», не опасаясь повреждения приемопередатчиков.

Отличительные особенности стенда SDK–1.1:

- простая архитектура на базе широко распространенного процессорного ядра MCS51;
- широкий набор периферийных блоков, включая часы реального времени, энергонезависимую память EEPROM, двустрочный ЖКИ, матричную клавиатуру;

- удобная работа с аналоговыми сигналами благодаря встроенным в микроконтроллер блокам АЦП и ЦАП.

Отличительные особенности стенда SDK–2.0:

- высокая производительность процессорного ядра ARM7 [7];
- большое разнообразие коммуникационных интерфейсов, как проводных (RS232C, RS485, CAN 2.0, Ethernet), так и беспроводных (IEEE 802.15.4);
- графическая консоль.

На рисунке приведен внешний вид стенда SDK–2.0.

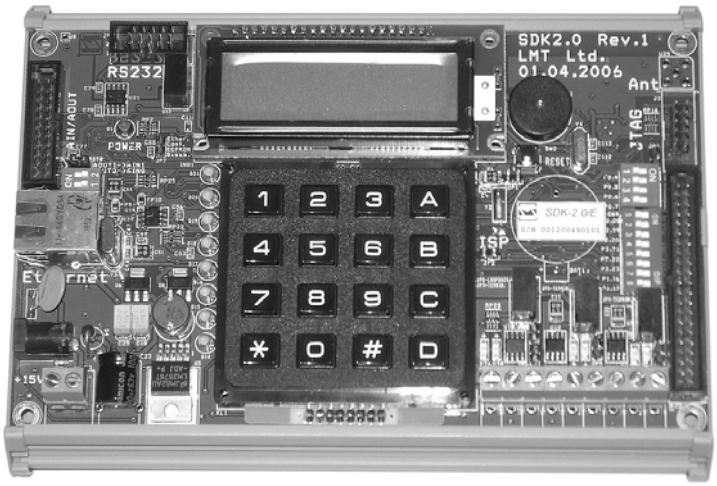


Рисунок. Учебный лабораторный стенд SDK–2.0

Отличительные особенности стенда SDK–6.1:

- в основе стенда – микросхема ПЛИС средней емкости;
- встроенная память (FLASH) для хранения нескольких конфигураций микросхемы ПЛИС;
- связь с ПК через интерфейс USB или RS–232 (с гальванической изоляцией).

В рамках предложенной модели обучения стенд SDK–1.1 может использоваться на первом уровне, стенд SDK–2.0 – на всех трех уровнях, стенд SDK–6.1 – на втором и третьем уровне. Далее приводятся конкретные примеры применения названных контроллеров семейства SDK в учебном процессе.

Учебный лабораторный стенд SDK–1.1

Для ознакомления с базовыми элементами и основами программирования микропроцессорных систем на первом уровне предлагаемой модели обучения с использованием учебного стенда SDK–1.1 могут проводиться следующие лабораторные работы.

1. Дискретные порты ввода–вывода.
2. Таймеры. Система прерываний.
3. Последовательный интерфейс.
4. Клавиатура.
5. АЦП. Прямой доступ к памяти.
6. Жидкокристаллический индикатор.
7. Последовательный интерфейс I2C.

Каждая работа включает в себя разработку драйвера периферийного устройства и выполнение с его использованием прикладной задачи, определенной вариантом задания. Подробное описание лабораторных работ представлено на сайте [8].

Перечисленные темы входят в комплекс лабораторных работ, проводимых на 4 курсе кафедры вычислительной техники СПбГУ ИТМО по дисциплинам «Интерфейсы

периферийных устройств» (лабораторные работы 1–4) и «Информационно-управляющие системы» (лабораторные работы 5–7) для студентов всех специализаций. В создании и ведении этих лабораторных работ принимали участие авторы статьи.

Далее приводятся краткие описания некоторых лабораторных работ.

1. Последовательный интерфейс.

Данная лабораторная работа посвящена изучению принципов работы с контроллером последовательного обмена – универсальным асинхронным приемопередатчиком (UART) микроконтроллера ADuC812. Модуль UART поддерживает протокол обмена для интерфейса RS232C. В работе требуется реализовать драйверы UART, работающие с использованием и без использования прерываний, с контролем ошибочных состояний. Драйвер UART, реализованный в режиме прерываний, использует программные буферы FIFO для организации взаимодействия процесса обработки прерывания и пользовательского процесса в виде API-функций.

В результате выполнения работы студент знакомится со следующими темами:

- организация работы с асинхронными процессами и устройствами, механизмы синхронизации (критические секции, взаимные исключения);
- организация последовательной передачи данных;
- стандарт RS232C;
- реализация гальванической изоляции последовательного интерфейса.

2. Клавиатура.

В работе требуется реализовать драйвер клавиатуры стенда SDK–1.1, работающего по таймеру микроконтроллера ADuC812, устраняющий эффект дребезга контактов. Драйвер клавиатуры должен определять факты нажатия и отпускания клавиш, идентифицировать скан-код нажатой клавиши и записывать его в программный буфер FIFO, иметь возможность настраивать задержку и скорость автоповтора передачи скан-кода нажатой клавиши в буфер.

В результате выполнения работы студент знакомится со следующими темами:

- организация работы с асинхронными процессами и устройствами, механизмы синхронизации (критические секции, взаимные исключения);
- принципы работы с матричной клавиатурой;
- методы сканирования клавиатуры, эффект дребезга контактов.

Учебный лабораторный стенд SDK–2.0

Темы лабораторных работ с использованием учебного стенда SDK–2.0 могут быть такими же, что и в лабораторном комплексе на базе SDK–1.1. Отличие заключается в более сложной архитектуре микропроцессорного ядра ARM7 в стенде SDK–2.0 по сравнению с архитектурой микроконтроллера MCS–51 стенда SDK–1.1, а также в наличии разнообразных коммуникационных интерфейсов. Предлагаемые лабораторные работы могут проводиться для ознакомления с принципами организации и программирования микропроцессорных систем на первых двух уровнях описанной модели обучения.

1. Порты ввода–вывода.
2. Таймер. Контроллер прерываний.
3. Последовательный канал.
4. АЦП и ШИМ.
5. Матричная клавиатура.
6. Графический ЖКИ.
7. Интерфейс I2C.
8. Системный супервизор. Часы реального времени.
9. Интерфейс RS485.

10. Интерфейс CAN–2.0.
11. Интерфейс Ethernet.
12. Беспроводной интерфейс IEEE 802.15.4.

Далее приводятся краткие описания некоторых лабораторных работ.

1. Таймер. Контроллер прерываний.

В работе требуется реализовать драйвер системного таймера, работающего по прерыванию. С помощью драйвера таймера предлагается реализовать динамическую индикацию линейки светодиодов и генерацию звука различных частот с помощью звукового пьезоизлучателя. В процессе выполнения лабораторной работы студент знакомится со следующими темами:

- режимы работы встроенных в ядро микропроцессора таймеров;
- организация векторного контроллера прерываний;
- особенности процедуры настройки прерываний от периферийных блоков микропроцессорного ядра ARM7.

2. Интерфейс Ethernet.

Задачей лабораторной работы является реализация драйвера контроллера Ethernet и осуществление обмена данными между двумя стендами SDK–2.0, объединенными в сеть типа «точка–точка» или «звезда» (в качестве центрального узла используется стандартный маршрутизатор). Целью лабораторной работы является приобретение навыков работы с интерфейсом Ethernet на канальном уровне стека протоколов TCP/IP.

Также стенд SDK–2.0 можно использовать как базу для выполнения курсовых проектов в рамках второго уровня описанной выше модели обучения, например, по следующим темам.

1. Построение контроллерной сети. Целью работы является построение сети контроллеров SDK–2.0, состоящей из 3–4 узлов, с гетерогенной средой передачи данных при помощи следующих интерфейсов: RS485, CAN–2.0, Ethernet, IEEE 802.15.4.
2. Создание Web-сервера. Целью работы является реализация Web–интерфейса, обеспечивающего отображение состояния периферийных блоков стенда SDK–2.0. Доступ к Web-серверу осуществляется по интерфейсу Ethernet.
3. Знакомство с технологией программирования в среде операционных систем реального времени (ОСРВ). Целью работы является адаптация (портирование) драйверов периферийных устройств стенда SDK–2.0 для работы под управлением ОСРВ и выполнение перечисленных выше лабораторных работ в среде ОСРВ.

На третьем уровне модели обучения могут выполняться следующие курсовые работы.

1. Реализация многофункционального загрузчика. Целью работы является реализация загрузчика программ/данных, обеспечивающего программирование всех доступных ресурсов памяти (FLASH, FRAM, внутреннее ОЗУ, внешнее ОЗУ) и передачу управления в любую указанную область памяти (в которой возможно исполнение программ). Прием данных должен осуществляться по следующим интерфейсам: RS232C, CAN и IEEE 802.15.4.
2. Построение беспроводной сети с ячеистой топологией. Целью работы является изучение принципов построения самоорганизующихся беспроводных сетей. Особое внимание в работе должно быть уделено вопросам надежности передачи данных, а также минимизации потребления энергии узлами сети. На практике требуется построить сеть, включающую 5–8 контроллеров SDK–2.0. В качестве среды передачи данных используется радиоканал стандарта IEEE 802.15.4 (ZigBee).

3. Проектирование элементов операционной системы реального времени. Целью работы является изучение и реализация основных компонентов операционной системы: переключатель задач, планировщик задач, средства межпроцессного взаимодействия и др.

Учебный лабораторный стенд SDK–6.1

Стенд SDK–6.1 предназначен для обучения основам проектирования современных электронных модулей на базе микросхем программируемой логики (ПЛИС) средней емкости.

Темы лабораторных работ с использованием стенда SDK–6.1, которые могут проводиться на втором уровне предлагаемой модели обучения.

1. Знакомство с архитектурой микросхем ПЛИС.
2. Устройства индикации. Кнопки.
3. Логические и арифметические структуры.
4. Внутрисхемная отладка через интерфейс JTAG.
5. Интерфейс RS232C.
6. Интерфейс I2C.
7. Память EEPROM и FLASH.
8. Алгоритмы и блоки сигнальной обработки данных.

Далее приводятся краткие описания некоторых лабораторных работ.

1. Логические и арифметические структуры.

В даннной работе предлагается реализовать один или несколько базовых блоков микропроцессорной техники, таких как триггеры, счетчики, сдвиговые регистры. Работу реализованных блоков необходимо проверить с помощью устройств ввода–вывода стенда SDK–6.1 (светодиодные индикаторы, кнопки, переключатели). В процессе выполнения лабораторной работы студент осваивает организацию базовых элементов микропроцессорной техники.

2. Интерфейс RS232C.

В работе нужно реализовать приемопередатчик последовательного асинхронного интерфейса RS232C с использованием библиотечных элементов САПР, и с помощью двух стендов SDK–6.1, соединенных между собой, убедиться в его работоспособности. В качестве линий передачи данных следует использовать дискретные порты ввода–вывода (в таком случае передача данных осуществляется на TTL–уровне сигналов). В ходе выполнения данной лабораторной работы студент должен ознакомиться со следующими темами:

- внутренняя организация асинхронного приемопередатчика последовательного интерфейса RS232C;
- построение самосинхронизирующихся интерфейсов.

Далее приведены возможные темы курсовых проектов, выполняющихся на базе стенда SDK–6.1 в рамках третьего уровня описанной модели обучения.

1. Проектирование простейшего процессора.
2. Проектирование блоков памяти.
3. Проектирование интерфейсов передачи данных.
4. Проектирование параллельных микропроцессорных систем.

Заключение

На сегодняшний день учебные лабораторные комплексы, разработанные в ООО «ЛМТ», используются в учебном процессе ряда вузов Российской Федерации и других государств, в том числе Санкт-Петербургского государственного университета инфор-

мационных технологий, механики и оптики, Санкт-Петербургского государственного политехнического университета, Гродненского государственного технического университета (Беларусь), Политехнического института Сибирского федерального университета (Красноярский государственный технический университет), Уральского государственного технического университета и др.

В статье предложена модель обучения принципам организации микропроцессорных систем на основе контроллеров семейства SDK, которая сейчас находится на стадии формализации и развития.

Литература

1. Ключев А.О., Кустарев П.В., Платунов А.Е. Инструментальные и учебные контроллеры семейства SDK // Компоненты и технологии. – 2002. – № 5. – С. 70–73.
2. Ключев А.О., Платунов А.Е. Встроенные инструментальные средства современных микроконтроллеров // Электронные компоненты. – 2002. – № 7. – С. 1–4.
3. Сайт компании «ЭФО». – Режим доступа: http://efo.ru
4. Сайт научно-производственной фирмы ООО «ЛМТ». – Режим доступа: http://lmt.ifmo.ru
5. Научно-производственная фирма ООО «ЛМТ» и ее продукция // Промышленные АСУ и контроллеры. – 2003. – № 2. – С. 24–26.
6. Сайт центра информационных систем кафедры вычислительной техники СПбГУ ИТМО. – Режим доступа: http://cis.ifmo.ru
7. ARM7TDMI–S. Technical Reference Manual // 2001 ARM Limited.
8. Сайт направления «Информационно-управляющие системы» кафедры вычислительной техники СПбГУ ИТМО. – Режим доступа: URL: http://emdedded.ifmo.ru

ВЫЧИСЛЕНИЕ ВРЕМЕННЫХ ХАРАКТЕРИСТИК ОБЪЕКТНО-СОБЫТИЙНЫХ МОДЕЛЕЙ ВСТРОЕННЫХ СИСТЕМ

А.Н. Лукичев
Научный руководитель – к.т.н., доцент А.Е. Платунов

Предложен способ вычисления временных характеристик объектно-событийных моделей встроенных систем на основе свойств полного частично упорядоченного множества атрибутов компонент модели и теоремы Кнастера-Тарского о неподвижной точке. Предложенный способ учитывает иерархичность моделей, допускает произвольные функции вычисления атрибутов отдельных компонент и использование обратной связи в модели. Результаты работы положены в основу реализации прототипа САПР системного уровня.

Введение

Встроенные системы (ВсС) представляют собой специализированные цифровые вычислительные системы, непосредственно взаимодействующие с объектом контроля или управления и объединенные с ним одной конструкцией. Отличительными характеристиками систем этого класса являются: работа в режиме реального времени, параллелизм и распределенность вычислительного процесса, специфическая аппаратура и программное обеспечение, жесткие конструктивные, вычислительные и другие ограничения, гетерогенность и т.д.

Объектно-событийная модель вычислений (ОСМВ) [1] предназначена для моделирования класса систем, в которых компоненты вычислительного процесса (функциональные блоки) связаны как информационными, так и управляющими потоками данных. Модель позволяет статический (без проведения имитационного моделирования) расчет временных характеристик системы, учитывает выбор вычислительной платформы и предоставляет средства для проверки выполнения требований к ее временным характеристикам. Тем не менее, модель является поведенческой, т.е. позволяет имитационное моделирование поведения проектируемой системы [2].

ОСМВ является достаточно молодой разработкой, и в ее теоретическом наполнении существуют серьезные недостатки, препятствующие широкому практическому применению для проектирования. В частности, математический аппарат ОСМВ [1] не позволяет вычислить временные характеристики моделей с обратной связью и не допускает альтернативный способ (функцию) вычисления характеристик отдельных компонентов модели. В статье представлен способ расчета на основе теоремы о неподвижной точке, позволяющий устранить эти ограничения.

Объектно-событийная модель вычислений

Объектно-событийная модель вычислений была предложена Постниковым [1] для моделирования поведенческого аспекта встроенных систем в рамках аспектной технологии проектирования. Вычислительный процесс в системе описывается с помощью функциональных блоков (ФБ), взаимодействующих через синхронные и асинхронные порты. События (изменения состояния сигнала) на синхронном входе активизируют ФБ, запуская или продолжая некий алгоритм, завершающийся генерацией события на выходе. События на асинхронном входе игнорируются в том смысле, что ФБ не реагирует на каждое событие по асинхронному входу, а лишь анализирует состояние на нем при возникновении события на одном из синхронных входов. Асинхронные входы можно назвать входами данных, синхронные – управляющими входами ФБ.

Функциональный блок представляет собой, таким образом, абстрактный вычислитель, выполняющий инструкции различного типа (события на синхронных входах) и обрабатывающий данные этих инструкций (состояние асинхронных входов). Конкрет-

ными примерами такого вычислителя могут быть: объект ООП, аппаратный вычислитель (микропроцессор), задача ОСРВ, драйвер устройства.

Сигнал (информационный или управляющий поток) моделируется как строго упорядоченная последовательность событий (данных или инструкций), снабженных временными метками.

Помимо функциональных свойств (алгоритмов преобразования сигналов), ФБ отражает временные свойства представляемой им подсистемы. Временные свойства задаются следующими атрибутами:

- *минимальное время обработки событий по синхронным входам*. Вектор $\Gamma = \|\gamma_j\|$, где γ_j есть параметр входа с номером *j*, задает эти величины (атрибуты) для всех входов ФБ;
- *сложность генерации выходного события*. Матрица $\mathrm{T} = \|\tau_{ij}\|$, задающая количество инструкций по входу *j*, необходимое для генерации одного события на выходе *i*;
- *минимальный интервал генерации выходных событий*. Вектор $\Omega = \|\omega_i\|$, задающий либо характеризующий данную величину (атрибут) по всем выходам ФБ.

При проектировании сверху вниз в процессе декомпозиции крупного функционального блока получается набор более мелких ФБ, реализующих поведение исходного и имеющих свои атрибуты. Каждый из полученных ФБ, в свою очередь, также подвергается декомпозиции, и так далее до тех пор, пока не будет получен набор примитивных ФБ, декомпозиция которых не имеет смысла, выражающихся в средствах выбранной вычислительной платформы. Такие ФБ называются атомарными функциональными блоками (АФБ). АФБ, очевидно, существуют только при наличии вычислительной платформы, на базе которой они реализованы. Конкретная реализация в рамках вычислительной платформы является неотъемлемым признаком АФБ. Временные атрибуты Γ и Ω для АФБ задаются характеристиками платформы, в рамках которой они реализованы.

Другим элементом ОСМВ является целевой узел (ЦУ), представляющий собой элемент вычислительной платформы, на которой реализована вся или часть системы. Примерами целевых узлов могут служить микропроцессор конкретного семейства, ПЛИС, набор микросхем жесткой логики, канал передачи данных на базе конкретного интерфейса, ОСРВ, виртуальная машина, набор программных библиотек. В процессе отображения ФБ на целевой узел его атрибуты Γ и Ω фиксируются, отражая производительность платформы, на которой реализована система.

ОСМВ предоставляет способ расчета вектора Ω, если известен вектор Γ, позволяя, таким образом, вычислить выходные характеристики ФБ по заданным входным. Если все инструкции интерпретируются функциональным блоком последовательно (например, при программной реализации), то

$$\Omega = \mathrm{T}\Gamma, \qquad (1)$$

то есть, обычное умножение матрицы T на вектор Γ. Если разнотипные (то есть, по разным синхронным входам) инструкции интерпретируются ФБ параллельно (например, при аппаратной реализации), то

$$\Omega = \mathrm{T} \otimes \Gamma, \qquad (2)$$

где $\otimes$ представляет собой операцию, результат которой есть вектор с элементами $\omega_i = \max\limits_{j=1..M} \tau_{ij}\gamma_j$.

Математический аппарат [1] также позволяет вычислить временные характеристики модели (композиции функциональных блоков), однако способ их вычисления требует, чтобы зависимость между атрибутами Γ и Ω всех ФБ в модели выражалась уравнениями (1) либо (2), использует нетривиальные преобразования матриц T функциональных блоков и не дает ответа при наличии обратной связи в модели.

Операции по нетривиальному (в зависимости от топологии конкретной модели и параллельной или последовательной интерпретации инструкций отдельными ФБ) объединению матриц T могут оказаться сложными в реализации и требовать больших затрат памяти при автоматизированном проектировании в рамках ОСМВ. Способ требует хранения в памяти «плоских» (без учета иерархии) матриц T всех ФБ и получения одной, учитывающей матрицы каждого атомарного ФБ, матрицы T для всей проектируемой системы. В то же время современные ВсС могут выражаться тысячами и десятками тысяч примитивных элементов платформы (АФБ). Это означает, что применение такого способа расчета временных характеристик значительно затрудняет практическое использование ОСМВ для проектирования ВсС.

Частично упорядоченное множество атрибутов ФБ

Атрибут синхронного входа (γ_j) или выхода (ω_i) представляет собой минимальный интервал времени между событиями в сигнале (на заданном входе или выходе ФБ). Если в качестве модели времени использовать множество $T = \mathbf{R}_+ \cup \{\infty\}$ неотрицательных действительных чисел плюс бесконечность (что является довольно простой и интуитивно понятной моделью времени), то, очевидно, любое значение этих атрибутов будет являться элементом этого множества (отрезок времени).

Обозначим β_j – некоторый атрибут (синхронного входа или выхода), равный минимальному интервалу между событиями в сигнале, $\beta_j \in T$. Между двумя значениями β_{j1} и β_{j2} можно установить отношение $\preceq$ такое, что

$$\beta_{j1} \preceq \beta_{j2} \Leftrightarrow \beta_{j2} \le \beta_{j1}, \qquad (3)$$

т.е. оно инвертирует обычное отношение чисел $\le$.

Обозначим $\beta_\perp = \infty$ – интервал между событиями в сигнале, который содержит менее двух событий (например, начальный сброс или однократное нажатие кнопки). Тогда множество всех значений β_j будет полностью упорядочено отношением $\preceq$ и будет иметь наименьший элемент $\beta_\perp$.

Возьмем вектор $\mathbf{B}$, содержащий *M* элементов β_j. Очевидно, $\mathbf{B} \in T^M$, где множество $T^M = \underbrace{T \times T \times \ldots \times T}_{M\ раз}$ есть множество всех наборов из *M* элементов множества T. Установим между любыми двумя векторами $\mathbf{B}_1, \mathbf{B}_2 \in T^M$ отношение $\preceq$ такое, что

$$\mathbf{B}_1 \preceq \mathbf{B}_2 \Leftrightarrow \forall j = 1 \ldots M : \beta_{j1} \preceq \beta_{j2}, \qquad (4)$$

где β_{j1} и β_{j2} есть элементы первого и второго векторов соответственно. То есть, два вектора связаны отношением $\preceq$ тогда и только тогда, когда все их соответствующие элементы связаны отношением (3). Отношение (4) рефлексивно ($\mathbf{B}_1 \preceq \mathbf{B}_1$), антисимметрично (если $\mathbf{B}_1 \preceq \mathbf{B}_2$ и $\mathbf{B}_2 \preceq \mathbf{B}_1$, то $\mathbf{B}_1 = \mathbf{B}_2$), транзитивно (если $\mathbf{B}_1 \preceq \mathbf{B}_2$ и $\mathbf{B}_2 \preceq \mathbf{B}_3$, то $\mathbf{B}_1 \preceq \mathbf{B}_3$), и делает множество векторов T^M частично упорядоченным.

Множество T^M содержит вектор $\mathbf{B}_\perp$, все элементы которого равны $\beta_\perp$. Легко показать, что $\mathbf{B}_\perp$ является наименьшим элементом T^M с отношением $\preceq$, что превращает это множество в полное частично упорядоченное множество. Каждое его направленное подмножество имеет точную верхнюю грань (так как любой элемент вектора β_j ограничен значением 0), а все множество имеет наименьший элемент $\mathbf{B}_\perp$.

Функция вычисления атрибутов модели

Вектор Ω может быть вычислен по вектору Γ с помощью соотношения (1) или (2). Абстрагируемся от их сути и заменим их функцией вычисления атрибутов выходов по атрибутам синхронных входов $F_\Omega : T^M \to T^N$ (M есть число синхронных входов, N число выходов ФБ). Единственное свойство (1) и (2), которое оставим, будет непрерывность F_Ω (в самом деле, легко показать, что преобразования (1) и (2) непрерывны).

Рассмотрим произвольную композицию конечного числа ФБ с функциями вычисления атрибутов $F_{\Omega 1}$, $F_{\Omega 2}$, …, $F_{\Omega n}$. Каждая из этих функций вычисляет атрибуты соответствующих ФБ Ω_1, Ω_2, …, Ω_n по их атрибутам Γ_1, Γ_2, …, Γ_n. Элементы этих векторов представляют собой атрибуты выходов или синхронных входов соответствующих ФБ. Если некоторое ω_i является атрибутом выхода ФБ с номером l, то оно вычисляется с помощью функции $F_{\Omega l}$ при наличии атрибутов всех его синхронных входов, т.е. если можно сформировать вектор Γ_l. Тогда, если есть атрибуты всех синхронных входов (т.е. можно сформировать все векторы Γ_1, Γ_2, …, Γ_n), то с помощью соответствующих функций можно вычислить атрибуты всех выходов.

Разделим все множество синхронных входов композиции на множество «внешних» и «внутренних». «Внутренний» вход отличается тем, что он имеет соединение с одним из выходов ФБ в композиции. Например, для композиции на рис. 1 (все входы в которой синхронные) «внутренними» являются входы, атрибуты которых обозначены γ_2, γ_3 и γ_5, а внешними являются входы с атрибутами γ_1 и γ_4.

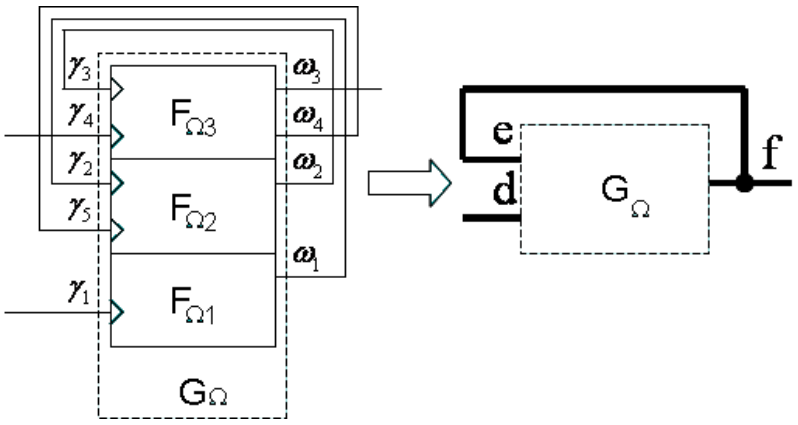


Рис. 1. Приведение произвольной композиции ФБ к виду с обратной связью

Сформируем из атрибутов «внешних» синхронных входов вектор **d**. Сформируем из атрибутов «внутренних» синхронных входов вектор **e** и дополним его при необходимости элементами со свободными значениями так, чтобы число элементов этого вектора совпадало с числом всех выходов композиции. Для примера на рис. 1 вектор **e** придется дополнить одним элементом, так как число выходов на единицу больше числа «внутренних» синхронных входов. Сформируем из атрибутов всех выходов композиции вектор **f** таким образом, чтобы его элемент с номером i был атрибутом выхода, соединенного с «внутренним» синхронным входом, атрибут которого является элементом с номером i вектора **e**. Для этого, возможно, потребуется продублировать некоторые элементы вектора **f**, если выход соединен с несколькими «внутренними» входами. Для примера на рис. 1 векторы будут иметь значения

$$\mathbf{d} = \begin{vmatrix} \gamma_1 \\ \gamma_4 \end{vmatrix},\ \mathbf{e} = \begin{vmatrix} \gamma_2 \\ \gamma_3 \\ \gamma_5 \\ x \end{vmatrix},\ \mathbf{f} = \begin{vmatrix} \omega_1 \\ \omega_2 \\ \omega_4 \\ \omega_3 \end{vmatrix}, \quad (5)$$

где x есть свободный элемент, который может принимать любое значение.

Выше было замечено, что если известны атрибуты всех синхронных входов, то с помощью соответствующих функций $F_{\Omega 1}$, $F_{\Omega 2}$, …, $F_{\Omega n}$ можно вычислить атрибуты всех выходов. Обозначим G_Ω функцию, вычисляющую вектор $\mathbf{f}$, содержащий атрибуты всех выходов композиции, по векторам $\mathbf{d}$ и $\mathbf{e}$:

$$\mathbf{f} = G_\Omega(\mathbf{d}, \mathbf{e}). \quad (6)$$

При этом, если некоторый элемент ω_i вектора $\mathbf{f}$ является атрибутом выхода ФБ с номером l, то он вычисляется с помощью функции $F_{\Omega l}$. G_Ω можно рассматривать как суперпозицию функций $F_{\Omega 1}$, $F_{\Omega 2}$, …, $F_{\Omega n}$. Очевидно, если все функции $F_{\Omega 1}$, $F_{\Omega 2}$, …, $F_{\Omega n}$ непрерывны, то G_Ω также непрерывна.

Обозначим $G_d(\mathbf{e}) = G_\Omega(\mathbf{d}, \mathbf{e})$ значение функции G_Ω при заданных атрибутах «внешних» синхронных входов $\mathbf{d}$.

Вследствие того, что все «внутренние» входы в композиции соединены с некоторыми выходами, а свободные элементы вектора $\mathbf{e}$ могут принимать любые значения, $\mathbf{e} = \mathbf{f}$ и

$$\mathbf{e} = G_d(\mathbf{e}). \quad (7)$$

Вычисление (а при наличии явной обратной связи в композиции – и *вычислимость*) атрибутов всех выходов в композиции сводится к нахождению (и существованию) неподвижной точки функции G_d, удовлетворяющей уравнению (7).

Множество E всех значений векторов $\mathbf{e}$ с отношением $\preceq$ является полным частично упорядоченным множеством. Теорема о неподвижной точке, которая считается одним из вариантов теоремы Кнастера-Тарского [3], формулирует условие существования и позволяет найти неподвижную точку функции (7).

Теорема 1. *Пусть дано полное частично-упорядоченное множество (E, $\preceq$). Тогда для любой непрерывной функции $G_d : E \to E$ существует неподвижная точка $\mathbf{e} = G_d(\mathbf{e})$. Причем, эта точка может быть найдена вычислением последовательности*

$$\mathbf{e}_\perp,\ G_d^1(\mathbf{e}_\perp),\ G_d^2(\mathbf{e}_\perp),\ \ldots,\ G_d^n(\mathbf{e}_\perp), \quad (8)$$

для некоторого n, в которой $G_d^i(\mathbf{e}_\perp) = G_d(G_d^{i-1}(\mathbf{e}_\perp))$, $G_d^0(\mathbf{e}_\perp) = \mathbf{e}_\perp$, а $\mathbf{e}_\perp$ есть минимальный элемент E.

Функция G_d непрерывна, так как непрерывна G_Ω [3], и ее неподвижная точка всегда может быть найдена за конечное число шагов. Чтобы вычислить атрибуты всех выходов композиции, достаточно взять вектор атрибутов «внешних» синхронных выходов $\mathbf{d}$, в качестве начального значения атрибутов «внутренних» входов взять вектор $\mathbf{e}_\perp$ (все элементы которого будут равны $\beta_\perp$) и итеративно вычислять значения функции G_d. На некотором шаге последовательность вычисленных значений сойдется к неподвижной точке.

Обозначим неподвижную точку (7) $fix(G_d)$. Если вспомнить, что $\mathbf{e} = \mathbf{f}$, то можно переписать (6):

$$\mathbf{f} = G_\Omega(\mathbf{d}, fix(G_d)) = F_\Omega(\mathbf{d}). \quad (9)$$

Функция $F_\Omega : D \to E$ (D есть множество всех значений векторов $\mathbf{d}$) представляет собой функцию вычисления атрибутов всех выходов композиции при заданном значении атрибутов «внешних» синхронных входов (см. рис. 2).

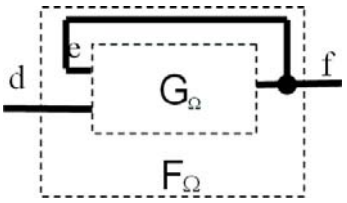


Рис. 2. Функция (9) вычисления атрибутов

F_Ω непрерывна, если непрерывна G_Ω [3].

Выводы

В работе применен подход и результаты, заимствованные из [3]. Однако в [3] с помощью вычисления неподвижной точки решается другая проблема (существование и поиск поведения системы в рамках различных моделей вычислений), не имеющая отношения к проблематике статьи.

Предложенный способ не требует конкретного вида функций $F_{\Omega 1}$, $F_{\Omega 2}$, …, $F_{\Omega n}$ вычисления атрибутов отдельных ФБ, а требует только, чтобы они были непрерывны. Зависимость между атрибутами выходов и входов отдельных ФБ может выражаться соотношением (1) или (2), а потенциально – любым другим непрерывным соотношением.

Способ не зависит от порядка интерпретации инструкций (последовательного или параллельного) функциональными блоками модели, в отличие от математического аппарата [1]. Это позволяет задавать способы расчета атрибутов для смешанных (параллельно-последовательных) вариантов порядка интерпретации [2] и использовать их для расчета атрибутов отдельных ФБ при вычислении атрибутов для всей модели [4].

Математический аппарат [1] не позволяет вычислить атрибуты выходов при наличии обратной связи в модели. Для функциональных блоков со строгой зависимостью выходов от входов (событие на выходе не может быть сгенерировано, пока не поступят инструкции по всем необходимым входам), которую задает матрица T, это и не требовалось. Легко показать, что любая композиция таких ФБ с обратной связью приведет к блокировке (deadlock). Тем не менее, обратная связь представляет собой базовый подход в проектировании управляющих систем, и ОСМВ в его отсутствие вряд ли получит широкое использование [2]. Актуальность этого вопроса подчеркивается тем, что аналогичная проблема существует и в синхронных языках (Statecharts, Esterel) [5]. В них она решается введением «нестрогих» компонентов, позволяющих генерацию событий на выходах, если известно состояние только части входов. Исследование в области формального обоснования «нестрогих» функциональных блоков могло бы решить эту проблему в ОСМВ. Способ вычисления атрибутов таких ФБ, очевидно, может отличаться от (1) и (2), однако этот вопрос выходит за рамки данной статьи. Предложенный способ вычисления временных характеристик композиции, тем не менее, делает возможным применение таких ФБ в моделях.

В отличие от математического аппарата [1], предложенный способ учитывает иерархию модели в том смысле, что при изменении одной из моделей на более низком уровне абстракции требуется пересчитать только ту часть высокоуровневой модели, которая непосредственно зависит от нее, а не всю модель целиком. Способ не требует

построения и хранения в памяти матриц T, что делает его более экономичным при вычислении характеристик крупных моделей, с точки зрения используемой памяти.

Результаты работы положены в основу реализации прототипа САПР системного уровня с поддержкой ОСМВ на основе комплекса моделирования Ptolemy II [6].

Литература

1. Постников Н.П. Поведенческий и инструментальный аспекты проектирования встроенных вычислительных систем: Дис. … канд. техн. наук / СПбГИТМО (ТУ), 2004.
2. Лукичев А.Н. Исследование моделей вычислений встроенных систем: Дис. … магистр. / СПбГУ ИТМО, 2005.
3. Xiaojun Liu. Semantic Foundation of the Tagged Signal Model. / Technical Report UCB/EECS-2005-31 – Electrical Engineering and Computer Sciences, University of California at Berkeley, 2005
4. Лукичев А.Н. Временные характеристики функциональных блоков при дискретно-событийном моделировании встроенных систем // Сборник докладов ИММОД-2007. Том 1. ISBN: 978-5-98361-048-4. СПб: ФГУП ЦНИИ технологии судостроения, 2007.
5. Gerard Berry. The Foundations of Esterel. / Proof, Language, and Interaction: Essays in Honour of Robin Milner. Editors: Gordon Plotkin, Colin Stirling, Mads Tofte. ISBN: 0262161885, MIT Press, 2000.
6. Cristopher Brooks, Edward A. Lee, Xiaojun Liu, Stephen Neuendorffer, Yang Zhao, Hayang Zheng. Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II). / Technical Report UCB/EECS-2007-7 - EECS Department, UC Berkeley, 2007.

СТОХАСТИЧЕСКИЕ ОСОБЕННОСТИ ВРЕМЕННЫХ И СОБЫТИЙНЫХ КОРРЕЛЯЦИЙ В ЭКВИ- И НЕЭКВИДИСТАНТНЫХ СЕРИЯХ АСТРОФИЗИЧЕСКИХ ДАННЫХ

В.М. Залялиева, С.А. Демин
(Татарский государственный гуманитарно-педагогический университет)
Научный руководитель – д.ф.-м.н., профессор Р.М. Юльметьев
(Татарский государственный гуманитарно-педагогический университет)

На основе техники проекционных операторов и формализма функций памяти выполнен анализ событийных и временных корреляций в дискретной стохастической эволюции сложных систем. Исследована экви- и неэквидистантная динамика полного потока рентгеновского излучения разнообразных астрофизических объектов: микроквазаров, квазаров, блазаров, галактик и пульсаров. Предложены информационные меры памяти, позволяющие классифицировать рентгеновское излучение астрофизических объектов. Выделено три класса астрофизических объектов в зависимости от соотношения масштабов статистической памяти и времен корреляций в дискретных временных сериях. Предложенный метод может оказаться полезным при анализе временных и событийных корреляций в экви- и неэквидистантных дискретных сериях регистрируемых сигналов сейсмических, астрофизических, социальных и живых систем.

1. Введение. Анализ экви- и неэквидистантных временных серий сложных систем

Исследование дискретной временной эволюции сложных систем на основе статистических методов анализа относится к одному из перспективных направлений в современной науке. Сложные системы состоят из большого числа взаимодействующих элементов. Такие системы имеют значительное (порой бесконечное) число степеней свободы. Извлечение информации о природе, характеристиках или свойствах таких систем, как правило, основывается на обработке экспериментальных временных серий. Регистрация показателей и параметров сложных систем осуществляется как с постоянным, так и переменным шагом дискретизации. С целью устранения неэквидистантности в регистрации временных серий часто прибегают к их усреднению. Эта процедура зачастую приводит к потере части информации о дискретной временной эволюции сложных систем.

В данной работе представлен оригинальный статистический метод анализа эффектов статистической памяти в неэквидистантных временных серий сложных систем. Суть метода состоит в переходе от анализа временных корреляций и флуктуаций, проявляющихся в усредненных временных сериях, к анализу событийных корреляций в неэквидистантных сигналах. В качестве примера приведем анализ неэквидистантных экспериментальных серий сейсмической активности локализованных участков земной коры, который проводится в работе [1]. Динамика потока рентгеновского излучения астрофизических объектов рассмотрена в работе [2], физиологических параметров живых систем, экономических и экологических показателей в [3], некоторых социальных, биологических и модельных систем – в работах [4, 5]. Термины *событие*, *время события* используются не только при анализе неэквидистантных временных серий. К анализу событий приводят также данные современных экспериментов в ядерной физике [6]. В последнее десятилетие для прогнозирования катастрофических явлений и процессов часто используют статистику экстремальных событий, т.е. набор максимальных значений показателей и параметров, фиксируемый на определенном временном сегменте. К настоящему моменту на основе анализа статистики экстремальных событий накоплено большое количество точных и эмпирических результатов [7, 8]. Следует отметить, что, в отличие от статистики экстремальных событий, где под событием понимается некоторое экстремальное значение сигнала, наш метод позволяет обрабатывать временную

серию с учетом всех значений, каждое из которых воспринимается как событие. В нашем методе выполняется анализ всего массива экспериментальных данных, регистрируемых со случайным шагом дискретизации в линейной шкале событий. Цель настоящей статьи состоит в выявлении стохастических особенностей динамики полного потока рентгеновского излучения различных астрофизических объектов.

2. Формализм функций памяти

Методы статистической физики весьма эффективны в анализе эффектов статистической памяти в дискретной стохастической динамике сложных систем. Для выявления эффектов статистической памяти в разнообразных системах, в частности в конденсированных средах, наиболее простой универсальной оказалась теория кинетических уравнений Цванцига–Мори. Важной особенностью многих сложных систем является большое число степеней свободы. Проекционный формализм Цванцига–Мори позволяет получить точные уравнения движения для связанных динамических переменных. Динамика всех остальных несвязанных степеней свободы скрыта в так называемых ядрах памяти. В этом разделе изложены основные понятия и положения статистической теории дискретных немарковских случайных процессов, которая представляет собой дискретное обобщение формализма функций памяти Цванцига и Мори на случай сложных систем негамильтоновой природы. Ранее в работах [9, 10] было предложено выражение для дискретной временной корреляционной функции (ВКФ): $a(t) = \left\langle \mathbf{A}_{\mathbf{k}}^{\mathbf{0}}(0) \cdot \mathbf{A}_{\mathbf{m+k}}^{\mathbf{m}}(t) \right\rangle \Big/ \left\langle \left| \mathbf{A}_{\mathbf{k}}^{\mathbf{0}}(0) \right|^2 \right\rangle$, представленной в виде скалярного произведения векторов начального и текущего состояний сложной системы:

$$\mathbf{A}_{\mathbf{k}}^{\mathbf{0}}(0) = \{\delta x_0, \delta x_1, \delta x_2, ..., \delta x_{k-1}\}, \mathbf{A}_{\mathbf{m+k}}^{\mathbf{m}}(\mathrm{t}) = \{\delta x_m, \delta x_{m+1}, \delta x_{m+2}, ..., \delta x_{m+k-1}\}.$$

Случайный дискретный процесс описывается временной серией $\{x_i\}$ случайной величины $X = \{x(\mathrm{T}), x(\mathrm{T}+\tau), ..., x(\mathrm{T}+(\mathrm{N}-1)\tau)\}$. Здесь T – начальный момент регистрации сигнала, $(\mathrm{N}-1)\tau$ – время регистрации сигнала, τ – временной шаг дискретизации сигнала, $x_j = x(T + j\tau)$ – значение случайной величины X на j-м шаге, флуктуация величины X на j-м шаге определяется в виде: $\delta x_j = x_j + \langle X \rangle$. Исходная ВКФ $a(t) = M_0(t)$ связана с функциями памяти более высокого порядка $M_n(t)$ $(n = 1,2...)$ посредством цепочки взаимосвязанных конечно-разностных уравнений:

$$\frac{\Delta M_{n-1}}{\Delta t} = \lambda_n M_{n-1}(t) - \tau \Lambda_n \sum_{j=0}^{m-1} M_n(j\tau) M_{n-1}(t - [j+1]\tau).$$

Здесь λ_n – кинетические (собственные значения квазиоператора Лиувилля $\mathcal{L}$), а Λ_n – релаксационные (размерностью квадрата частоты) параметры, соответственно.

3. Формализм функций памяти на случай событийных корреляций

Перейдем к случаю неэквидистантных временных серий. По аналогии с работой [11] воспользуемся техникой проекционных операторов для определения цепочки взаимосвязанных конечно-разностных уравнений для событийной корреляционной функции (СКФ) $a(n)$ и событийных функций памяти (СФП) $M_s(n)$.

Представим временные вариации полного потока рентгеновского излучения произвольного астрофизического объекта в виде последовательности случайных событий:

$$E = \{\xi_1, \xi_2, \xi_3, ..., \xi_k, ..., \xi_N\}, \quad (1)$$

где ξ_i – событие, зафиксированное в момент времени t_i, $i = 1,...,\mathrm{N}$ – номера событий.

Среднее значение $\langle E \rangle$, флуктуации $\delta\xi_i$ и абсолютную дисперсию σ^2 для последовательности из N событий можно представить в следующем виде:

$$\langle E \rangle = \frac{1}{N}\sum_{i=1}^{N}\xi_i, \delta\xi_i = \xi_i - \langle E \rangle,$$
$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\delta\xi_i^2 = \frac{1}{N}\sum_{i=1}^{N}\{\xi_i - \langle E \rangle\}^2. \quad (2)$$

По аналогии с работой [9] представим СКФ в виде

$$a(n) = \frac{1}{(N-m)\sigma^2}\sum_{i=1}^{N-m}\delta\xi_i\delta\xi_{i+m}.$$

Здесь $n = m\Delta n$, $\Delta n = 1$ – шаг дискретизации в событийной шкале. Нормированная СКФ удовлетворяет условиям нормировки и ослабления корреляций: $\lim_{n\to 1} a(n) = 1$, $\lim_{n\to\infty} a(n) = 0$.

Для описания эволюции дискретной последовательности событий (1) воспользуемся конечно-разностным уравнением движения Лиувилля:

$$\frac{\Delta\xi_i(n)}{\Delta n} = i\mathcal{L}(n,\Delta n)\xi_i(n). \quad (3)$$

Представим совокупность значений динамической переменной $\delta\xi_j = \delta\xi(j\Delta n), j = 1,...,\mathrm{N}$ в виде k-компонентного вектора в линейном евклидовом пространстве векторов состояния системы:

а) вектор начального состояния исследуемой сложной системы:

$$\mathbf{A}_{\mathbf{k}}^{\mathbf{1}} = \{\delta\xi_1, \delta\xi_2, \delta\xi_3,..., \delta\xi_k\}, \quad (4a)$$

b) вектор конечного состояния системы, который определяется сдвигом на расстояние m событий по дискретной шкале событий:

$$\mathbf{A}_{\mathbf{m+k}}^{\mathbf{m}} = \{\delta\xi_{m+1}, \delta\xi_{m+2}, \delta\xi_{m+3},..., \delta\xi_{m+k}\}, \quad (4b)$$

где $n = m\Delta n, 1 \le k \le N - m$.

С учетом (4) нормированную СКФ удобнее представить в следующем виде:

$$a(n) = \frac{\left\langle \mathbf{A}_{\mathbf{k}}^{\mathbf{1}}(1)\cdot\mathbf{A}_{\mathbf{k+m}}^{\mathbf{m}}(\mathrm{n})\right\rangle}{\left\langle\left|\mathbf{A}_{\mathbf{k}}^{\mathbf{1}}(1)\right|^2\right\rangle}. \quad (5)$$

С помощью техники проекционных операторов по аналогии с работой [11] для исходной СКФ можно получить замкнутое конечно-разностное уравнение немарковского типа:

$$\frac{\Delta a(n)}{\Delta n} = \lambda_1 a(n) - \Delta n\Lambda_1\sum_{j=1}^{m}M_1(j\Delta n)a\left(\left[m-j+1\right]\Delta n\right).$$

Здесь λ_1 – собственное значение квазиоператора Лиувилля $\mathcal{L}$, Λ_1 – релаксационный параметр размерности квадрата частоты, $M_1(j\Delta n)$ – нормированная функция памяти первого порядка:

$$\lambda_1 = i\frac{\left\langle \mathbf{A}_{\mathbf{k}}^{\mathbf{1}}(1)\mathcal{L}\mathbf{A}_{\mathbf{k}}^{\mathbf{1}}(1)\right\rangle}{\left\langle\left|\mathbf{A}_{\mathbf{k}}^{\mathbf{1}}(1)\right|^2\right\rangle}, \Lambda_1 = \frac{\left\langle \mathbf{A}_{\mathbf{k}}^{\mathbf{1}}(1)\mathcal{L}_{12}\mathcal{L}_{21}\mathbf{A}_{\mathbf{k}}^{\mathbf{1}}(1)\right\rangle}{\left(\left|\mathbf{A}_{\mathbf{k}}^{\mathbf{1}}(1)\right|^2\right)},$$

$$M_1(j\Delta n) = \frac{\left\langle \mathbf{A_k^1}(1)\mathcal{L}_{12}\left(1 + i\Delta n\mathcal{L}_{22}\right)\mathcal{L}_{21}\mathbf{A_k^1}(1)\right\rangle}{\left\langle \mathbf{A_k^1}(1)\mathcal{L}_{12}\mathcal{L}_{21}\mathbf{A_k^1}(1)\right\rangle}.$$

Пошагово повторяя вышеуказанную процедуру, можно записать последующее конечно-разностное кинетическое уравнение для нормированной событийной функции памяти (СФП) первого порядка $M_1(n)$, а далее – для функций памяти более высокого порядка.

Для упрощения данного алгоритма можно также воспользоваться процедурой ортогонализации Грама–Шмидта [12]:

$$\left\langle \mathbf{W_s}, \mathbf{W_p} \right\rangle = \delta_{sp}\left\langle \left| \mathbf{W_s}^2 \right| \right\rangle, \delta_{sp} = \begin{cases} 1, \text{в случае } s = p, \\ 0, \text{в случае } s \neq p, \end{cases}$$

где δ_{sp} – символ Кронекера. Процедура ортогонализации Грама-Шмидта позволяет превратить линейно зависимую систему в ортонормированную. Теперь мы можем получить рекуррентную формулу $\mathbf{W_s} = \mathbf{W_s(n)}$ для определения множества ортогональных динамических переменных:

$$\mathbf{W_0} = \mathbf{A_k^1}, \mathbf{W_1} = \{i\mathcal{L} - \lambda_1\}\mathbf{W_0}, \mathbf{W_2} = \{i\mathcal{L} - \lambda_2\}\mathbf{W_1} - \Lambda_1\mathbf{W}_0, \ldots$$

С помощью новых проекционных операторов для ортогональных переменных $\mathbf{W_s}$, приходим к цепочке связанных конечно-разностных уравнений немарковского типа для нормированных СФП (s-1)-го порядка:

$$\frac{\Delta M_{s-1}(n)}{\Delta n} = \lambda_s M_{s-1}(n) - \Delta n\Lambda_s \sum_{j=1}^{m} M_{s-1}(j\Delta n)M_{s-1}\left([m - j + 1]\Delta n\right). \qquad (6)$$

Представленные выше конечно-разностные кинетические уравнения (6) представляют собой дискретное обобщение статистической теории дискретных немарковских стохастических процессов на случай событийных корреляций в дискретной стохастической эволюции сложных систем негамильтоновой природы.

4. Информационные меры статистической памяти

В последнее время заметно возрос интерес к исследованию статистических эффектов памяти, связанных с природой реальных объектов. При этом усилия физиков направлены, прежде всего, на поиск фундаментальных статистических индикаторов и численных показателей и на оптимизацию таких мер для описания дискретной временной эволюции сложных систем разнообразной природы.

Для количественного описания эффектов статистической памяти в стохастической динамике сложных систем негамильтоновой природы воспользуемся статистическими кванторами памяти $\varepsilon_i(\nu)$, $\delta_i(\nu)$. Первая информационная мера для случая эквидистантных временных серий впервые был предложена в работах [9, 10]. Этот статистический квантор памяти позволяет выполнить сравнение и сопоставление времен релаксации исходной корреляционной функций и функций памяти. Позднее в работе [11] было представлено обобщение данной информационной меры памяти на случай неэквидистантных временных серий:

$$\varepsilon_i(\nu) = \left\{ \frac{\mu_{i-1}(\nu)}{\mu_i(\nu)} \right\}^{\frac{1}{2}}.$$

Здесь $\mu_i(\nu)$ – частотный спектр мощности СФП i-го порядка $M_i(n)$:

$$\mu_i(\nu) = \left| \Delta n \sum_{n=1}^{N} M_i(n) \cos(2\pi n \nu) \right|^2 .$$

Вторая информационная мера была введена первоначально в работе [13] на случай эквидистантных временных серий и обобщена в работе [11] для случая неэквидистантных временных серий:

$$\delta_i(\nu) = \left| \frac{\widetilde{M}_i'(\nu)}{\widetilde{M}_{i+1}'(\nu)} \right| .$$

Здесь $\widetilde{M}_i'(\nu) = d\widetilde{M}_i(\nu)/d\nu$, где $\widetilde{M}_i(\nu)$ – Фурье-образ соответствующей событийной функции памяти $M_i(n)$. Данная информационная мера позволяет выявить эффекты марковизации (или демарковизации) в произвольном релаксационном процессе.

С помощью информационных мер памяти релаксационные процессы условно можно разделить на три класса:

(1) немарковские процессы, в динамике которых проявляется долговременная (сильная) статистическая память, $\varepsilon, \delta \approx 1$;

(2) квазимарковские процессы с умеренной статистической памятью, $\varepsilon, \delta > 1$;

(3) марковские процессы, отличающиеся кратковременной (слабой) статистической памятью, $\varepsilon, \delta >> 1$.

Значения $\varepsilon_1(0)$ и $\delta_1(0)$ указанных информационных мер памяти отражают дальнодействующие особенности исходных экви- и неэквидистантных временных серий. Данные показатели являются уникальными информационными мерами памяти и количественно описывают эффекты статистической памяти в стохастической эволюции сложных систем негамильтоновой природы.

5. Анализ экви- и неэквидистантных серий астрофизических данных

5.1. Экспериментальные данные

Используемые нами экспериментальные данные представляют собой показатели динамики полного потока рентгеновского излучения различных астрофизических объектов. Регистрация рентгеновского излучения осуществлялось обзорной камерой ASM (All-Sky Monitor), находящейся на околоземном космическом спутнике Rossi X-Ray Timing Explorer (RXTE) [14]. Космическая станция RXTE была запущена 30 декабря 1995 г. в рамках проекта NASA с крупнейшего космодрома США им. Кеннеди. Высота круговой орбиты спутника – 580 км, что соответствует периоду обращения 90 мин. Контролирует спутник-обсерваторию Центр космических полетов им. Годдарда [15]. Наблюдения за объектами ведется при помощи трех камер, чувствительных в различных энергетических диапазонах: A (1,5–3 *кэВ*), B (3–5 *кэВ*), C (5–12 *кэВ*). База данных содержит также экспонированные значения 90-секундных наблюдений полного потока рентгеновского излучения S (1,5–12 *кэВ*). Суммарный поток рентгеновского излучения представлен в виде собственных единиц ASM ($ASM\ counts/\sec$). $1\,Crab = 75\ ASM\ counts/\sec$, где один *Crab* есть полный поток Крабовидной туманности в диапазоне $2 \div 10\ кэВ$ [14].

В настоящей работе проведен анализ экви- и неэквидистантных временных серий полного потока рентгеновского излучения 48 астрофизических объектов: 19 микроквазаров, 8 активных ядер галактик (1 квазар, 7 блазаров), 12 нормальных галактик (3 радиогалактики, 9 галактик Сейферта) и 9 пульсаров (3 аномальных пульсара и 6 миллисекундных рентгеновских пульсара). Регистрация полного потока рентгеновского излучения данных объектов осуществлялась в период с 1 января 1996 г. по 1 января 2006 г. Динамика полного рентгеновского потока для каждого из объектов представлена в виде

дискретных временных рядов двух типов: временной ряд, усредненный по дням (эквидистантная серия – I тип данных), исходная запись сигнала с переменным шагом дискретизации (неэквидистантная серия – II тип данных).

На примере микроквазара XTE J1550-564 далее будут продемонстрированы спектры мощности исходной ВКФ и СКФ, а также частотные зависимости информационных мер памяти для двух типов экспериментальных данных. Рентгеновская новая XTE J1550-564, расстояние до которой $5.3\,кпк$ [16], открыта 7 сентября 1998 г. при помощи обзорной камеры ASM спутника RXTE. Наиболее вероятным оптическим компонентом источника считается звезда блеска $V = 16^m.7 \pm 0^m.1$ с широкими и яркими эмиссионными линиями (Ha, Hb, He II) [16]. Масса этой звезды приблизительно равна массе Солнца, а масса компактного объекта оценивается в 9.4 масс Солнца [17]. Микроквазар XTE j1550-564 является мягким рентгеновским транзиентом с интенсивностью $70\,mCrab$ $(1.5 \div 12\,кэВ)$ и координатами $\alpha = 15^h 50^m 41^s, \delta = -56^0 27' 6''$ [18]. В сентябре 1998 г.а у источника впервые были обнаружены высокочастотные квазипериодические осцилляции (QPO) с центральной частотой 271 ± 2 мГц и шириной на полувысоте 30 ± 5 мГц [19]. В период с 4 по 5 марта 1999 г. наблюдался переход объекта в высокое состояние с увеличением потока в мягкой области (2–20 *кэВ*), а также ужесточением степенной компоненты спектра и очередным появлением высокочастотных квазипериодических осцилляций [20]. Весной 2003 г. во время программы обзора галактической плоскости, проводимой спутником INTEGRAL, была зарегистрирована еще одна вспышка рентгеновской активности, но на этот раз в жесткой части спектра (> 20 *кэВ*) [21]. В течение этой вспышки объект находился в низком жестком состоянии [22].

5.2. Классификация интенсивности рентгеновского излучения астрофизических объектов

Ранее было установлено, что статистические кванторы ε, δ позволяют выделить три возможных сценария проявления статистической памяти в произвольных релаксационных процессах. Вычисление данных информационных мер памяти для полного потока рентгеновского излучения для двух типов данных свидетельствует о разной степени проявления статистической памяти. Это послужило основанием для классификации астрофизических объектов по значениям информационных мер памяти $\varepsilon_1(0)$ и $\delta_1(0)$. Отметим, что значения, полученные для эквидистантных временных серии, не совпадают ни в одном из рассмотренных случаев с результатами для неэквидистантных серий. Расхождение этих значений для некоторых объектов несущественно. Однако для некоторых объектов, к примеру, XTE j1550-564, эти два типа экспериментальных данных приводят к качественно различимым сценариям. В случае эквидистантной временной серии значения статистических кванторов составляют $\varepsilon_1(0) = 45.87$ и $\delta_1(0) = 2298.2$, что соответствует квазимарковскому процессу. В случае анализа II типа данных обнаруживается яркий марковский сценарий: $\varepsilon_1(0) = 156.91$ и $\delta_1(0) = 22944$.

Для численного сопоставления мер памяти $\varepsilon_1(0)$, $\delta_1(0)$ для двух типов астрофизических данных воспользуемся следующими критериями:

$$k_1 = \frac{\varepsilon_1^I(0)}{\varepsilon_1^{II}(0)}, k_2 = \frac{\delta_1^I(0)}{\delta_1^{II}(0)}.$$

В зависимости от значений данных критериев выделим три случая:

(1) $k_1, k_2 > 1$ – усреднение сигнала приводит к усилению статистической памяти и усилению робастности в хаотической динамике рентгеновской активности астрофизического объекта;

(2) $k_1, k_2 < 1$ – усреднение экспериментальных данных приводит к ослаблению статистической памяти и усилению случайного характера флуктуаций рентгеновского потока. Подобная информационная неравноценность двух типов данных объясняется тем, что усреднение экспериментальных данных неизбежно приводит к потере части качественно отличающейся информации о природе сложной системы. В представленных случаях анализ неэквидистантной временной серии оказывается более информативным и достоверным;

(3) $k_1, k_2 \approx 1$ – оба типа экспериментальных данных оказываются примерно равноценными. В этом случае анализ как экви-, так и неэкидистантной временной серий приводит к аналогичным результатам.

Значения k_1 и k_2 с учетом проведенной классификации представлены соответственно в табл. 1 и 2.

Микроквазары		квазары, лацертиды		галактики		пульсары	
I группа: $k_1 \approx 1$							
Gro j0422+32	1.07	1es0033+595	1.02	3c390.3	1.10	1e1048.1-5937	0.99
Grs 1758-258	1.15	3c273	1.05	Cen A	1.13	Igr j00291+5934	1.04
Rx j1826_2-1450	0.98	3c279	1.09	Cir Galaxy	1.04	Sax j1808.4-3658	1.12
Ss433	1.13	3c66a	1.09	Ic4329a	1.19	Xte j0929-314	1.07
		Oj287	1.03	Ic5063	1.01	Xte j1751-305	1.01
		V709cas	0.99	M87	1.10	Xte j1807-294	0.99
				Mcg-6-30-15	1.13	Xte j1810-197	0.99
				Mkn279	1.04	Xte j1814-338	0.99
				Mkn348	1.08		
				Ngc3516	1.06		
				Ngc3783	1.06		
				Ngc7674	1.05		
II группа: $k_1 > 1$							
1e1740_7-2942	1.29	1es1959+650	1.32			1e2259.0+5836	1.26
Cyg X-3	1.42	Mkn 421	2.36				
Gx 339-4	1.33						
Lsi+61303	1.24						
H1743-322	3.40						
Xte j1118+480	1.83						
Xte	2.00						

j1859+226							
III группа: $k_1 < 1$							
Cir X-1	0.27						
Cyg X-1	0.53						
Gro j1655-40	0.45						
Grs 1915+105	0.42						
Sco X-1	0.24						
V4641sgr	0.70						
Xte j0421+560	0.54						
Xte j1550-564	0.30						

Таблица 1. Классификация полного потока рентгеновского излучения астрофизических объектов согласно значениям критерия k_1

Микроквазары		квазары, лацертиды		галактики		Пульсары	
I группа: $k_2 \approx 1$							
Gro j0422+32	1.17	1es0033+595	1.19	3c390.3	1.11	1e1048.1-5937	0.96
Rx j1826_2-1450	0.99	3c273	1.06	Cir Galaxy	1.08	Igr j00291+5934	1.05
		3c66a	0.80	Ic5063	1.00	Xte j0929-314	1.09
		Oj287	1.18	Mkn279	1.08	Xte j1751-305	1.03
		V709cas	0.99	Mkn348	1.18	Xte j1807-294	0.99
				Ngc3516	1.17	Xte j1810-197	0.97
				Ngc3783	1.14	Xte j1814-338	1.16
				Ngc7674	1.19		
II группа: $k_2 > 1$							
1e1740_7-2942	1.67	1es1959+650	1.74	Ic4329a	1.32	1e2259.0+5836	1.58
Cyg X-3	2.39	Mkn 421	10.1	M87	1.35	Sax j1808.4-3658	1.60
Grs 1758-258	1.33			Mcg-6-30-15	1.29		
Gx 339-4	5.59						
H1743-322	10.9						
Lsi+61303	1.54						
Ss433	1.29						
Xte j1118+480	3.32						

Xte j1859+226	2.60						
III группа: $k_2 < 1$							
Cir X-1	0.07			Cen A	0.64		
Cyg X-1	0.27						
Gro j1655-40	0.31						
Grs 1915+105	0.55						
Sco X-1	0.06						
V4641sgr	0.56						
Xte j0421+560	0.30						
Xte j1550-564	0.10						

Таблица 2. Классификация интенсивности рентгеновского потока астрофизических объектов в зависимости от значений критерия k_2

Как видно из представленных таблиц, к первой группе можно отнести микроквазары, квазары, лацертиды, галактики и пульсары. Значения информационных мер памяти $\varepsilon_1(0)$, $\delta_1(0)$ для двух типов астрофизических данных приблизительно равны. Первая группа представлена наибольшим числом астрофизических объектов. Интерес вызывает вторая и третья группы классификации. В этих случаях усреднение экспериментальных данных приводит к искажению информации о некоторых характеристиках дискретной эволюции астрофизических объектов. В частности, усреднение неэквидистантных временных серий приводит к усилению (или ослаблению) статистической памяти. Фактически такое усреднение приводит к иному сценарию релаксационных процессов. В этом случае анализ всего набора данных позволяет извлечь более достоверную и детальную информацию о динамике рентгеновского излучения астрофизических объектов. Отметим, что для рентгеновской активности микроквазаров характерны как немарковские, так и марковские (квазимарковские) особенности. Хаотической динамике интенсивности рентгеновского излучения квазаров, лацертид, галактик и пульсаров свойственны немарковский и квазимарковский сценарии релаксационных процессов.

6. Временные и спектральные особенности рентгеновского излучения микроквазара XTE j1550-564

На рис. 1 представлены исходные временные серии (1 января 1996 г. – 1 января 2006 г.) полного потока рентгеновского излучения в мягкой области рентгеновского спектра 1,5–12 *кэВ* микроквазара XTE j1550-564. На рис. 1, a, представлена эквидистантная (усредненная) временная серия (шаг дискретизации 1 день). На рис. 1, b, неэквидистантная временная запись полного потока рентгеновского излучения астрофизического объекта представлена в линейной шкале событий (нумерация ведется по индексам). Шкалу событий можно связать с обычной временной шкалой при помощи среднего интервала между событиями. Увеличение потока в мягкой области спектра как для экви-, так и неэквидистантной временной серии связано с появлением высокочастотных квазипериодических осцилляций, о которых говорилось ранее.

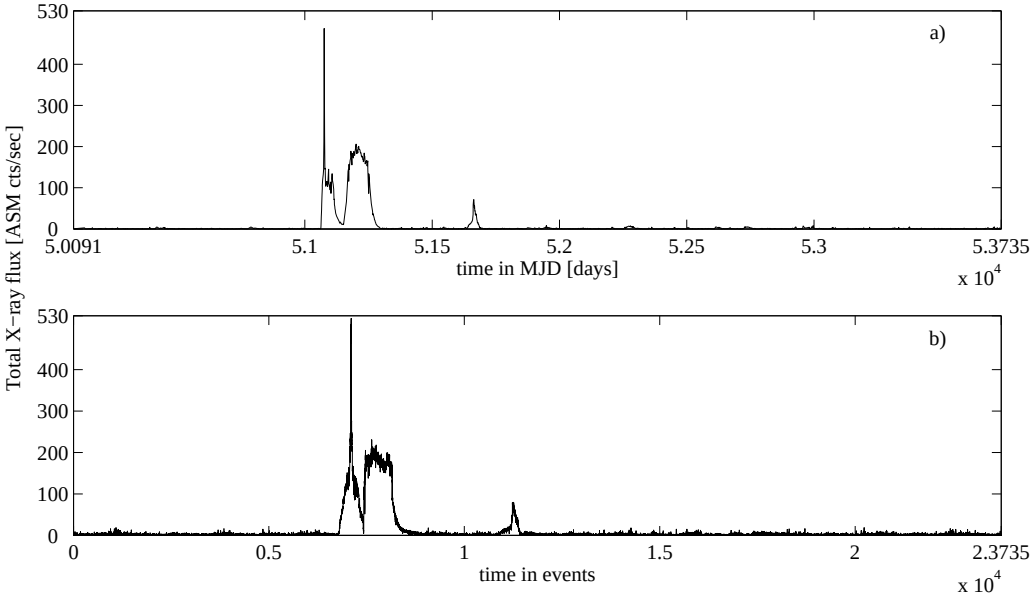


Рис. 1. Суммарный поток рентгеновского излучения микрокваза Xte j1550-564 (энергетический диапазон 1,5–12 кэВ). Период регистрации – 1 янв. 1996 г. – 1 янв. 2006 г. a) представлена эквидистантная временная серия полного потока рентгеновского излучения в модернизированной юлианской шкале. b) представлена неэквидистантная временная серия полного потока рентгеновского излучения в дискретной событийной шкале. Средний временной интервал между двумя событиями для Xte j1550-564 составляет 88,6 минут. Длина выборки в случае равноинтервальной серии – 3644 значений, неэквидистантной временной серии – 23735

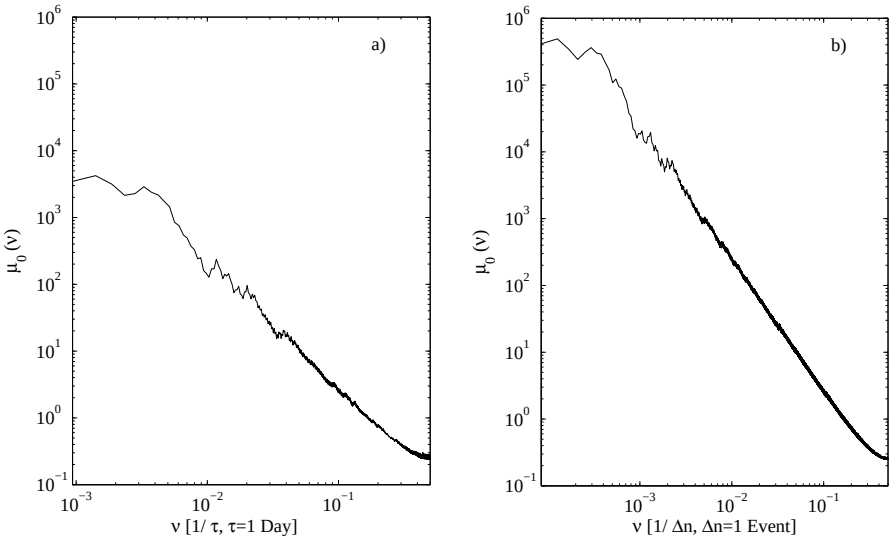


Рис. 2. Спектр мощности исходной корреляционной функции $\mu_0(\nu)$ для динамики полного потока рентгеновского излучения микрокваза Xte j1550-564. a) – частотная зависимость ВКФ, b) – спектр мощности СКФ. Графики представлены в дважды логарифмической шкале. В обоих случаях в спектрах мощности обнаруживается степенная зависимость $\mu_0(\nu) \approx \nu^{-\alpha}$

На рис. 2 представлены спектры мощности исходных ВКФ и СКФ $\mu_0(\nu)$ для экви- и неэквидистантной динамики полного потока рентгеновского излучения XTE j1550-564. На спектрах наблюдаются степенные зависимости: $\mu_0(\nu) \approx \nu^{-\alpha}$, где α – показатель фрактальности. В случае эквидистантной серии данный показатель принимает значение 1.75, в случае неэквидистантной временной серии $\alpha = 1.79$.

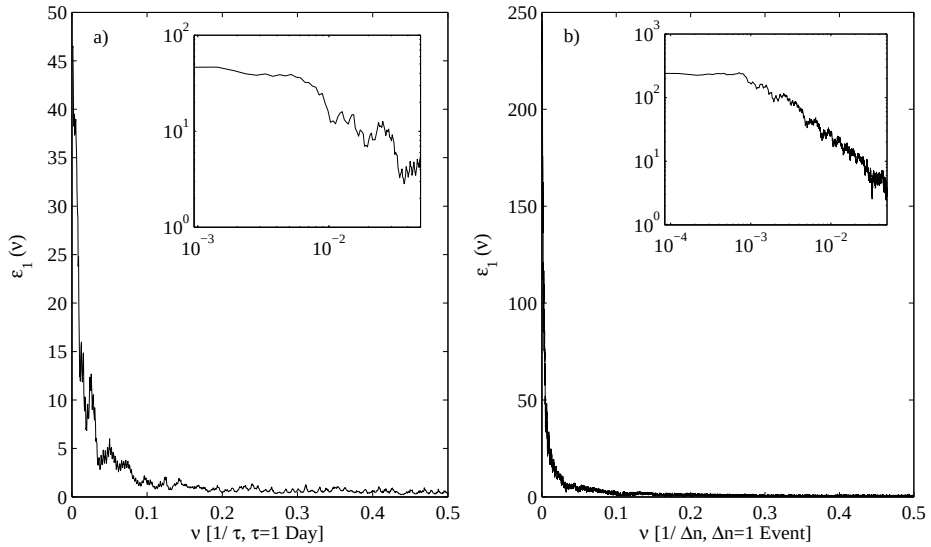


Рис. 3. Частотная зависимость первой информационной меры памяти $\varepsilon_1(\nu)$ для динамики полного потока рентгеновского излучения микроквазара XTE j1550-564. a) – частотная зависимость $\varepsilon_1(\nu)$ для случая эквидистантной временной серии, b) – частотная зависимость на случай событийных корреляций. Вставки к рисункам представляют области низких частот зависимости $\varepsilon_1(\nu)$ в дважды логарифмической шкале

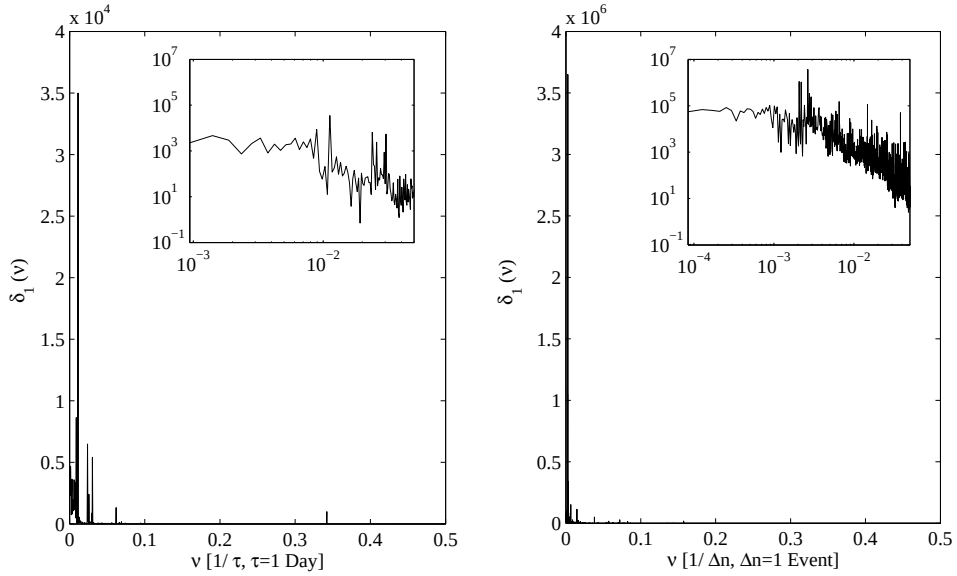


Рис. 4. Частотная зависимость второй информационной меры памяти $\delta_1(\nu)$ для динамики полного потока рентгеновского излучения микроквазара XTE j1550-564. a) – случай эквидистантной (усредненной по дням) серии, b) – случай неэквидистантной записи. На вставках к рисункам представлены низкочастотные области зависимости $\delta_1(\nu)$ в дважды логарифмической шкале

На рис. 3 представлена частотная зависимость первой информационной меры памяти $\varepsilon_1(\nu)$ для эквидистантной и неэквидистантной временных серий полного потока рентгеновского излучения XTE J1550-564. На рис. 3, a, в области низких частот обнаруживаются квазипериодические осцилляции, характеризующие дальнодействующие особенности исходного временного сигнала. Критерий $k_1 = 0.3$ означает ослабление статистической памяти при переходе от эквидистантной серии к неэквидистантной. Фактически обнаруживается отчетливый переход от квазимарковского сценария к марковскому при переходе от данных первого типа ко второму. Таким образом, усреднение

экспериментальных данных приводит к усилению статистической памяти и робастных компонент.

На рис. 4 представлена частотная зависимость второй информационной меры памяти $\delta_1(0)$ для эквидистантной и неэквидистантной временных серий полного потока рентгеновского излучения XTE J1550-564. Усреднение сигнала также приводит к усилению эффектов статистической памяти и регулярности в динамике полного потока рентгеновского излучения (значение критерия $k_2 = 0.1$).

7. Заключение

В работе представлен новый статистический метод анализа временных и событийных корреляций в дискретной временной эволюции сложных систем. Развитый метод позволяет количественно оценивать такие особенности экспериментальных серий, как время существования статистической памяти, эффекты регулярности и случайности, скорость релаксационных процессов. Преимущество нового метода заключается в том, что он позволяет выполнить анализ временных серий как с фиксированным временным шагом, так и с переменным шагом.

На основе информационных мер памяти нами разработана классификация стохастической динамики полного потока рентгеновского излучения различных астрофизических объектов в зависимости от времени существования статистической памяти. В хаотической динамике рентгеновского излучения таких объектов, как квазары, лацертиды, активные галактики и пульсары, проявляются преимущественно эффекты долговременной статистической памяти. Для релаксационных процессов рентгеновской активности таких объектов характерно более регулярное поведение. Интерес представляют результаты, связанные с анализом временных и событийных корреляций в интенсивности рентгеновского излучения микроквазаров. Среди этих объектов можно выделить объекты с немарковской, квазимарковской и марковской динамикой полного потока рентгеновского излучения. Подобное разнообразие в проявлении эффектов статистической памяти в рентгеновской активности микроквазаров, по-видимому, обусловлено более непредсказуемым механизмом аккреции вещества. Наши результаты в целом свидетельствуют о довольно сложном стохастическом характере и взаимосвязи физических процессов, ответственных за механизмы рентгеновского излучения. *A priori* нельзя говорить об определенном марковском или немарковском характере излучения, которое связано с различными физическими процессами, в том числе и аккрецией вещества. Конкретная природа рентгеновского излучения обусловлена большой совокупностью различных факторов, обладающих рядом специфических особенностей. Выявление этих особенностей представляет собой отдельную и достаточно трудную проблему, требующую проведения дополнительного анализа статистических свойств и характеристик астрофизических объектов.

Полученные здесь результаты можно рассматривать как первый шаг к пониманию стохастических особенностей релаксационных процессов рентгеновской активности астрофизических объектов. В качестве перспектив можно выделить исследование эффектов статистической памяти и эффектов регулярности и хаотичности в различных диапазонах энергетического спектра астрофизических объектов, анализ кросс- (перекрестных) корреляционных эффектов между различными энергетическими диапазонами астрофизических объектов, выявление взаимосвязи между временными и событийными флуктуациями активности астрофизических объектов и их физической природой.

Работа поддержана фондами: грант Федерального агентства по образованию Министерства образования и науки РФ № РНП.2.1.1.741, грант РФФИ 08-02-00123-а.

Литература

1. Abe S., Sarles N.V., Scordas E.S. et al. // Physics Review Letters. – 2005. – 94. – 170601-1.
2. Greenhough J., Chapman S.C., Chaty et al. // A&A. –2002. –385. – 693.
3. Varotsos P.A., Sarles N.V., Scordas E.S. et al. // Physics Review E. – 2005. – 71. – 011110-1.
4. Allegrini P., Barbi F., Grigolini P. et. al. // Physics Review E. –2006. – 73. – 046136-1.
5. Varotsos P.A., Sarles N.V., Scordas E.S. et al. // Physics Review E. – 2006. – 73. – 046136-1.
6. Baym G. and Heiselberg H. // Physics Letters B. –1999. – 469. – 7.
7. Eichner J.F, Kantelhardt J.W., Bande A., et al. // Physics Review E.–2006.–73.–016130-1.
8. Nicolis C., Balacrishnan V., Nicolis G. // Physics Review Letters.–2006.–97.–210602-1.
9. Yulmetyev R.M., Hдnggi P., Gafarov F.M. // Physics Review E.–2000.–62.–6178-6194.
10. Yulmetyev R.M., Hдnggi P., Gafarov F. // Physics Review E.–2002.–65.–046107-1-15.
11. Yulmetyev R.M., Demin S.A., Panischev O.Yu, et al. // Nonlinear Phenomena in Complex Systems.–2006.–9.–313-330.
12. Рид М., Саймон Б. Методы современной математической физики.– М.: Мир, 1997.–т. 1.–357 с.
13. Mokshin A.V., Yulmetyev R.M., Hдnggi P. // Physics Review Letters.–2005.–95.–200601-1-4.
14. The Rossi X-Ray Timing Explorer Project [electron resources] / Kavli Institute for Astrophysics and Space Research, Massachusetts Institute of Technology. – Access: http://xte.mit.edu/main.html, free. – Language english.
15. RXTE Quest Observer Facility [electron resources] / Goddard Space Flight Center. – Access: http://heasarc.gsfc.nasa.gov/docs/xte/xhp_geninfo.html, free. – Language english.
16. Orosz J.A., et al. // The Astrophysical Journal.–2002.–568.–845-861.
17. Castro-Tirado A. J., Duerbeck H. W., Hook I. // IAU Circ.–1999.–7013.
18. Smith D.A. // IAU Circ.–1998.–7008.
19. Finger M.H., Dieters S.W., Wilson R.B. // IAU Circ.–1998.–7010.
20. Homan J., Wijnands R., van der Klis M. // IAU Circ.–1999.–7121.
21. Dubath P., et al. // IAU Circ. –2003.–8100.
22. Sturner S.J., Shrader C. R. // IAU Circ. –2003.–8100.

МОБИЛЬНЫЙ МУЗЫКАЛЬНЫЙ СЕРВИС ДЛЯ ШИРОКОПОЛОСНЫХ СЕТЕЙ

С.Е. Антонов, Н.С. Токалов
Научный руководитель – к.п.н. А.В. Маятин

Неупорядоченность, ограничения использования, неуниверсальность, пиратство – этими словами можно описать отношение многих людей к аудиоданным. Развитие широкополосных беспроводных сетей и мобильных устройств открывает безграничные возможности для развития мира, в частности мира музыкальных сервисов. Появляется возможность создания универсального музыкального сервиса, доступного всем и ориентированного под конкретного пользователя, концепцию которого в рамках данной статьи мы и попытаемся рассмотреть.

Введение

Концепция современного мира трудно разделима с технологиями. Большое количество людей каждый день слушают музыку, используя радио, портативный плеер, сотовый телефон, Интернет. Это стало неотъемлемой частью жизни современного общества. Однако использование этих достижений цивилизации влечет за собой ряд неудобств. Радио нельзя слушать в метро, музыку на плеере приходится постоянно менять. Поиск медиаинформации, актуальный в данный момент, ведется постоянно и, зачастую, весьма непрактичными способами. Процесс обмена большими объемами музыкального контента – это очень актуальная проблема для широкого круга пользователей. При этом следует учесть тот факт, что нередко музыкальный контент распространяется незаконно и без соблюдения авторских прав, что требует внимания при разработке систем распространения медиаконтента.

Разрабатываемая система призвана разрешить некоторые эти проблемы, используя развивающиеся сейчас мобильные беспроводные технологии доступа к локальным сетям и Интернету. С помощью нее можно будет слушать радио в метро, быстро легко и легально обмениваться большими объемами медиаданных с друзьями, в любой момент находить и слушать то, что хочется.

Современные тенденции к минимизации техники, совмещающей в себе всевозможные функции, тенденции к «избавлению от проводов» и строительству беспроводных широкополосных сетей с использованием технологий Wi-Fi, WiMAX, 3G, CDMA отлично сочетаются с возможностью создания и развития данного проекта. Эти тенденции отмечают аналитические агентства по всему миру, в том числе и в России [1–4]. Развитие клиентских частей проекта на различных платформах и устройствах, интеграция разнообразных сервисов делает проектируемую систему очень перспективной.

Постановка задачи

Основная цель проекта – создание сервиса для прослушивания музыки (прогнозов погоды, анекдотов, аудиокниг, радио и др.) на мобильных устройствах, подключенных к беспроводным широкополосным сетям. В дальнейшем планируется портирование сервиса на обычные компьютеры, расширения функционала. Примерная схема функционирования проекта представлена на рис. 1.

Базовый функционал предполагает прослушивание музыки на базе объемного упорядоченного каталога, общение пользователей, обмен списками воспроизведения, прослушивания музыки в условиях непостоянного приема беспроводной сети. Первой задачей является создание кроссплатформенной серверной части и программы-клиента для устройств на базе Windows Mobile.

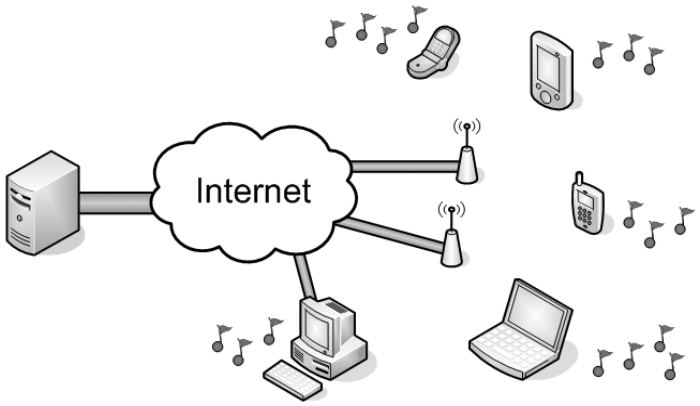


Рис. 1. Схема работы проекта

Разрабатываемый сервис должен обладать максимально гибкой для портирования между платформами архитектурой, с возможностью работы в условиях нехватки памяти и процессорных ресурсов.

Программирование в условиях ограниченности ресурсов

Программирование для мобильных устройств – это всегда «борьба» между неограниченным функционалом программы и ресурсами мобильного устройства. Показатели ресурсов мобильного устройства на сегодняшний день уже достигли немалых высот, однако им все же еще очень далеко до настольных компьютеров. И, как отмечают разработчики мобильных устройств, дальнейшее увеличение производительности ведет к существенному снижению времени автономной работы. Но мобильные устройства имеют возможность выхода в Интернет, и это дает им практически неограниченные возможности. Разработчикам остается лишь аккуратно расходовать имеющиеся ресурсы устройства и перекладывать все наиболее сложные вычисления и обработку на сторону сервера.

Использование WinAPI и C++ в Windows Mobile как способ разработки высокопроизводительных программ. На данный момент поддержка Java [8] в Windows Mobile осуществляется, только если к этому прикладывают усилия фирмы-разработчики мобильных устройств. Однако осуществляется поддержка языка C#, который, в свою очередь, не поддерживается на большинстве Windows-несовместимых устройств. Плюс, стоит отметить, что транслирование программ из исходного кода в байт-код и дальнейшее его исполнение с использованием виртуальной машины снижает производительность и требует от пользователя поддержки самой виртуальной машины. При этом, конечно, появляется возможность использования кроссплатформенности, однако, как показывает практика, при этом снова возникают проблемы совместимости функциональных возможностей различных устройств из-за различий непосредственно в hardware и ОС устройства. В связи с этим, для разработки клиентской части программного продукта был выбран язык программирования C++ для разработки логики работы программы и WinApi для обеспечения быстрой и корректной работы интерфейса. Разработка кода с использованием объектного подхода и с соблюдением стандартов позволяет в дальнейшем провести относительно простое портирование кода на платформу Symbian [9].

Потоковая буферизация, кэширование. Емкость flash-памяти, доступной для использования мобильными устройствами, неуклонно растет, но никогда не найдется мобильного устройства, которое сможет всегда хранить в себе все, что нужно в данный момент. Но даже если случится, что появится устройство достаточной емкости, то возникнут проблемы по обслуживанию информации, ее сортировке, упорядочиванию, обновлению, и эти проблемы будет сложно решать как в ручном режиме каждым пользо-

вателем, так и используя автоматизированные средства обработки информации на мобильном устройстве. В связи с этим решено организовать хранение данных на стороне сервера. Клиентская же часть программы будет подключаться к серверу, чтобы скачивать оттуда информацию, необходимую для воспроизведения в данный момент.

В отличие от настольных компьютеров, которые могут безболезненно сохранить в оперативной памяти много мегабайт данных, в мобильных устройствах не все так легко. Так как мы хотим обеспечить бесперебойное воспроизведение музыки даже при условии периодических обрывов связи, при буферизации нам придется по максимуму использовать зарезервированное место в оперативной памяти. Для этого можно использовать принцип конвейерной работы с буфером. При этом изначально буфер пытается заполниться целиком от начала. Так как воспроизведение музыки идет из буфера, то постепенно информация в буфере становится неактуальной. Чтобы этого не было, место в буфере, которое уже было прослушано, также считается неактуальным, пустым и подлежит заполнению. Воспроизведение данных из буфера идет по кругу. Заранее установлены минимальный размер загружаемого с сервера блока и минимальный размер загруженных данных, при котором идет воспроизведение. При окончании одного трека, идет буферизация следующего по списку трека. Это дает возможность смены треков даже при отсутствии связи. На данный момент эти принципы буферизации реализованы в клиентской части программы для Windows Mobile.

При разрыве связи программа ожидает подключения, и, когда соединение вновь устанавливается, она пытается по максимуму заполнить доступный буфер.

Для обеспечения большего времени автономной работы можно организовать кэширование данных, получаемых с сервера еще и на flash-память устройства. Это актуально для мобильных устройств с очень маленьким объемом оперативной памяти, но имеющих несколько (или много) большую flash-память в своем арсенале. Обычно у мобильных устройств все так и есть. Чтобы не вмешиваться в уже налаженное циклическое конвейерное воспроизведение музыки, можно организовать кэширование непосредственно в методах объектов, которые отвечают за загрузку данных с сервера. При этом функции работы с циклическим буфером останутся теми же, а вот функции загрузки данных с сервера будут работать с отдельным объектом, который будет сам загружать данные в flash-память (до определенного размера) и затем по мере надобности выдавать их функции. Таким образом, можно будет обеспечить очень продолжительное время работы без подключения к сети, что актуально при работе в условиях метро.

Клиентская часть программы

Клиентская часть программы представляет из себя исполняемый файл, написанный на языке программирования C++ с применением WinAPI [6].

Унификация функций работы со средой хранения и передачи данных, объектный подход. При разработке клиентской части программы использовались принципы объектно-ориентированного программирования и, по возможности, стандарты программирования на C++. Для повышения гибкости программы выделялись из объектов типовые функции взаимодействия со средой, например, функции получения данных с сервера, функции воспроизведения полученных данных. Это позволит в дальнейшем более просто вносить критические изменения в программу. Также такой подход позволит при портировании, например, на платформу Symbian оставить код части классов неизменным и переписать только взаимодействие программы со средой передачи информации, с графическим интерфейсом пользователя, аудио-интерфейсом воспроизведения и взаимодействие с flash-накопителем. Также объектно-ориентированный подход позволит в дальнейшем использовать созданный код в репозитории классов.

Особенности работы интерфейса клиентской части программы, типовые возможности. Для удобства и повышения скорости работы программы интерфейс разрабатывается с применением технологии WinAPI [7]. При этом логическая часть программы максимально абстрагирована от интерфейса, что дает возможности легкой переносимости из проекта в проект, или, что немаловажно, между различными Windows-подобными платформами.

Вся функциональная часть интерфейса по максимуму перенесена для выполнения на сторону сервера. Клиент после успешной авторизации начинает подгружать данные об интерфейсе с сервера в виде тегированного html кода с небольшими расширениями. На программную часть интерфейса, расположенную на клиентской машине возлагается несколько типовых задач:

- авторизация;
- взаимодействие с сервером данных, получение новых страниц интерфейса;
- отображение страниц интерфейса;
- обеспечение удобства и простоты в использовании;
- буферизация и воспроизведения медиаданных;
- воспроизведение плей-листов из списка недавно прослушанных.

Таким образом, получаем, что разработка интерфейса программы для клиентских приложений сводится к решению довольно типовых задач. Всем остальным будет заниматься единая для всех клиентов серверная часть сервиса.

Многопоточность. Клиентская часть программы написана с использованием многопоточности. В разные потоки разведены такие функции клиентской части:

- работа интерфейса (непосредственно взаимодействие с пользователем);
- буферизация и кэширование данных с сервера;
- воспроизведение звука.

Многопоточность позволяет избавиться от «подвисаний» клиентского приложения в моменты разрыва связи, при попытках соединения с сервером.

Взаимодействие между потоками осуществляется с помощью специально написанного класса «семафоров», от которого наследуются классы, которым необходимо кросспроцессорное взаимодействие.

Серверная часть программы, взаимодействие с клиентскими приложениями

Клиент-серверные взаимодействия в подобных сервисах – это очень критичный момент разработки, так как следует учесть аспекты безопасности, соблюдения авторских прав. Сервер должен быть спроектирован с учетом возможности работы в условиях больших нагрузок. Хотя данный момент очень сильно зависит и от «hardware» составляющей сервера, не стоит забывать, что грамотно спроектированная архитектура может сэкономить очень много ресурсов. На данном этапе разработки мы попытаемся определить и описать базовые возможности серверной части сервиса и описать некоторые особенности, не вдаваясь в проектирование всех особенностей, так как это выйдет за рамки данной статьи.

Разделение «обязанностей», типовые возможности. Для облегчения написания версий клиентских программ для различных платформ максимум функциональности предполагается разместить на стороне сервера и сделать работу с сервером такой же простой, как работа с обычными html-страницами, добавив к этому интерактивные возможности клиентских приложений. В связи с этим для программирования серверной части был выбран язык программирования PHP [5] – не самый быстрый из возможных, но достаточно удобный для разработок на начальном этапе. В дальнейшем, при появлении перегрузок сервера, возможно, придется переписать ядро, отвечающее за передачу

медиаданных клиентам на более производительном языке программирования, но это не вызовет особых сложностей. Интерфейсную же часть сервера, интегрированную с базой данных музыки, можно будет оставить на PHP, возможно, с некоторыми бинарными вставками.

В базовый функционал сервера следует включить следующие возможности:

- авторизация;
- возможность создания списков воспроизведения каждому клиенту в отдельности;
- возможность обмена списками воспроизведения между пользователями;
- возможность обмена сообщениями между пользователями;
- возможность для администратора группировки и занесения музыки с учетом основной информации о треках;
- возможность поиска информации по исполнителю, названию, жанру;
- «раздача» данных клиентам;
- сбор статистики.

Некоторые особенности разработки сервера. Главным отличием серверной части сервиса от обычной базы данных музыки является возможность передачи данных клиентским программам сервиса. Для решения этой задачи разработан специальный серверный модуль. Он по запросу клиента, «id» плей-листа и трека начинает ему передачу данных. В дальнейшем, для поддержания возможности соблюдения авторских прав и усложнения задачи считывания музыки, можно реализовать потоковое шифрование данных на стороне сервера и стороне клиента. Планируется очень тщательно протестировать повышение нагрузок на сервер и мобильное устройство и выбрать оптимальный алгоритм шифрования, который будет иметь оптимальное соотношение производительность/стойкость.

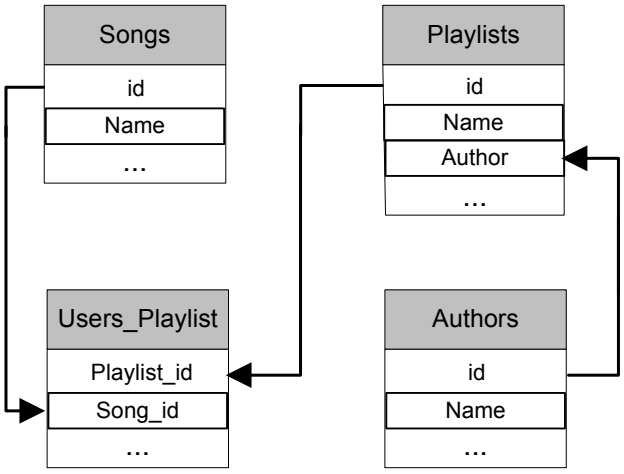


Рис. 2. Схема минимальной базы данных сервера

Для начала база данных сервера предполагает наличие таблицы с треками, таблицы с пользовательскими списками воспроизведения, и таблицы множественной связки между списками воспроизведения и треками (ее примерная схема представлена на рис. 2). Это позволит обеспечить минимальные возможности по созданию, обмену списков воспроизведения. Также это сделает возможным масштабирование системы без вмешательства в уже организованную систему таблиц в базе данных. В списки воспроизведения в любом случае будут добавляться непосредственно идентификаторы треков. Например, для того чтобы добавить к такой системе возможности поиска, сортировки по исполнителю, жанру и т.п., надо будет создать таблицы, описывающие жанр песни, исполнителей (и др., если требуется), которые будут описывать новые сущности системы, и таблицы множественной связки между треками и новыми таблицами. Необходимы множест-

венные связки, так как разные треки могут исполнять разные исполнители (или объединения исполнителей), также одну и ту же песню можно классифицировать в разных жанрах, все это надо учитывать. Следовательно, мы получаем расширение функционала без вмешательства в уже реализованные возможности сервера и клиента. Таким образом, можно очень сильно масштабировать систему, использовать модульную структуру разработок. При этом разработка нового функционала будет сводиться к программированию его на PHP на стороне сервера и отсутствию необходимости вносить изменения в клиентские программы.

В итоге получаем блоки сервиса «клиент» и «сервер», которые могут развиваться параллельно, не нарушая основной парадигмы взаимодействия, принятой при проектировании сервиса, и удачно дополнять друг друга.

Интеллектуальная система псевдослучайного воспроизведения

Данная интеллектуальная система призвана удовлетворить потребности пользователей по подбору музыки на основе анализа взаимодействия и вкусов других пользователей.

Статистический анализ данных о пользователе. В процессе взаимодействия пользователей с сервисом собирается статистика прослушивания треков. Эта статистика образует типичный топ-лист популярности. От него можно отталкиваться при построении псевдослучайного списка воспроизведения. Или же можно дать возможность пользователю самому задать «точку отсчета», указав группу, трек или жанр (и т.п.).

Одновременно с этим собирается статистика о последовательностях прослушивания треков, о том, какие треки слушает пользователь в списке с какими. Можно дать возможность пользователю оценить правильность псевдослучайного выбора системы, что тоже будет учитываться в статистических таблицах базы данных.

В плане реализации в виде базы данных это может быть таблица, состоящая из трех полей:

- идентификатор трека-объекта;
- идентификатор трека-субъекта;
- степень связи трека объекта и субъекта в процентах.

Собрав достаточно обширную статистику по трекам, жанрам, исполнителям сервиса, можно получить достаточно точную картину о вкусах людей, о том, что бы стал слушать человек, если он слушает трек «икс». Также эта статистика может быть крайне полезна для реализации модуля целевой рекламы на сервисе.

Особенности реализации интеллектуального подбора «любимой» музыки. В общем случае работа системы интеллектуального подбора сводится к следующим действиям. Сначала строится взвешенный граф, вершинами которого являются определенные треки, а вес ребер задан степенью связи трека-объекта и трека-субъекта. Далее рассчитываются вероятности перехода на соседние вершины графа в соответствии с весами соответствующих ребер, и строится распределение вероятности перехода. Наконец, генератор случайных чисел генерирует случайное число от 0 до 1, и в зависимости от него и распределения вероятности осуществляется переход на одну из соседних вершин графа.

На практике это подход связан с возникновением ряда проблем. Например, связей между некоторыми треками может быть очень много, что приведет к перегрузке сервера при подборе списка воспроизведения. Некоторые не очень популярные треки могут образовывать небольшие циклы, что будет не очень интересно пользователю системы. В этих случаях можно произвести следующие модификации системы: брать, например, не все связи, а верхние 10–15 связей; отслеживать отсутствие повторений в течение некоторого количества шагов или в течение всего воспроизведения. В случае обнаружения замкнутых циклов проиграть их единожды и далее направить пользователя по самой по-

пулярной музыке данного жанра или исполнителя, т.е. подключить еще один уровень логических связей. При разработке такой системы будет необходимо найти некий оптимум между функциональностью системы и нагрузкой на сервер, объемом базы данных.

Стоит отдельным пунктом отметить, что для более равномерного распределения нагрузки на сервер стоит не вычислять весь список псевдослучайного воспроизведения сразу, а рассчитывать воспроизведение на один–два шага вперед. Однако в этом случае остается открытым вопрос об оптимальной, с точки зрения производительности, функции исключения повторений и циклов.

Возможно, в дальнейшем отдельным проектом будет разработана система для построения многоуровневых стохастических логических цепочек и связей на основе статистического анализа данных и обучения системы. Такая система в процессе работы будет постоянно обучаться, и подстраиваться под текущую схему связей в графе предпочтений пользователей. Область применимости такой системы не будет ограничена возможностями данного медиапроекта, однако, в условиях реально функционирующей системы обслуживания клиентов можно будет отлично протестировать систему, выявить слабые места и логические противоречия или недоработки.

Заключение

Тенденции современного общества к минимизации и мобилизации электроники уже трудно отрицать, это объективный факт. Разработанная система может органично вписаться в необходимую функциональность мобильных устройств нового поколения; также она может помочь решить некоторые проблемы, сопутствующие использованию и распространению аудиоданных.

Для разработки системы, отвечающей первоначальным требованиям, целесообразно использовать клиент-серверную архитектуру с распределением «обязанностей» между клиентом и сервером. Функционал стоит максимально перенести на сторону сервера, а на стороне клиента организовать решение типовых задач. Такой подход к разработке сделает систему масштабируемой и легко портируемой на различные клиентские платформы. Целесообразность и жизнеспособность такого подхода подтверждается реализованным прототипом для Windows Mobile и сервера, поддерживающего PHP.

Литература

1. Портал новостей InfoArticles [Электронный ресурс] / Новости телекоммуникаций портала InfoArticles; ред. Кривенко С.Е. – Яз. рус. – Режим доступа http://infoarticles.ru/telecom/news_2007-10-18-22-25-02-817.html, свободный.
2. Деловая газета Взгляд [Электронный ресурс] / Интернет версия деловой газеты Взгляд. – Режим доступа: http://www.vz.ru/news/2007/4/3/75556.html, свободный. – Яз. рус.
3. ИКС Навигатор [Электронный ресурс] / Каталог услуг и продуктов отрасли ИКТ; авт. Евдокименко Е.Н. – Режим доступа http://www.iksnavigator.ru/vision/454805.html, свободный. – Яз. рус., англ.
4. Новостной портал Mobile-Review [Электронный ресурс] / Агрегатор новостей Mobile Review; ред. Тюрина И.В. – Яз. рус., англ. – Режим доступа http://www.mobile-review.com/fullnews/main/2007/May/15.shtml, свободный.
5. Сообщество программистов PHP [Электронный ресурс] / Официальный сайт языка программирования PHP – Режим доступа: http://php.net, свободный. – Яз. рус., англ.

6. Microsoft Developer Network [Электронный ресурс] / Официальный сайт подразделения компании Microsoft по взаимодействию с разработчиками – Режим доступа: http://msdn.ru, свободный. – Яз. рус., англ., нем.
7. Russian Software Developer Network [Электронный ресурс] / Сообщество русскоговорящих разработчиков программного обеспечения; ред. RSDN Magazine – Режим доступа: http://rsdn.ru, свободный. – Яз. рус., англ.
8. Язык программирования Java [Электронный ресурс] / Официальный сайт языка программирования Java – Режим доступа: http://java.com, свободный. – Яз. рус., англ., нем.
9. Операционная система Symbian OS [Электронный ресурс] / Официальный сайт Symbian OS – Режим доступа: http://www.symbian.com/, свободный. – Яз. англ.

РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА «ЗДОРОВЫЙ РЕБЕНОК» ДЛЯ АВТОМАТИЗАЦИИ РАБОТЫ ВРАЧА-ПЕДИАТРА

Д.Н. Труфанов, М.С. Фролова
(Тамбовский государственный технический университет)
Научный руководитель – д.т.н., профессор С.В. Фролов
(Тамбовский государственный технический университет)

Для повышения эффективности работы врача-педиатра, сокращения времени на выполнение обязательных рутинных операций и снижения вероятности врачебных ошибок разработан программно-аппаратный комплекс «Здоровый ребенок», который позволяет работать с детьми всех возрастов, и является первым в России автоматизированным рабочим местом врача-педиатра.

Введение

В реализации национального проекта «Здоровье» основное внимание уделяется укреплению первичного медицинского звена муниципальных поликлиник, основу которого составляют участковые врачи. Высокая нагрузка и ответственность ложится на врачей-педиатров.

За рабочий день, составляющий 7 часов, врач-терапевт должен принять 25–30 детей. Специалист также обязан заполнять большое количество различной документации, проводить измерения антропометрических данных (рост, вес, силу кисти и др.), вести поиск по различным таблицам и справочникам, позволяющим определять уровень развития ребенка, давать рекомендации. Все действия проводятся врачом вручную. Это ведет к утомлению специалиста, а также к повышению вероятности ошибок. Кроме того, за время приема (7–8 минут) у врача-педиатра не остается времени для полноценного общения с ребенком и его родителями.

Целью нашей работы стало создание программно-аппаратного комплекса для автоматизации работы участкового врача-педиатра, который предназначен для работы с детьми всех возрастов – от рождения до 18,5 лет.

К ближайшим аналогам предлагаемой разработки относятся «Автоматизированный комплекс диспансерных обследований» (АКДО), разработанный ООО «Интеллектуальные программные системы» (С.-Петербург). АКДО предназначен для обследований детей и подростков в возрасте от 3 лет до 20 лет. Одним из недостатков АКДО является то, что комплекс не рассчитан для работы с детьми до 3-х лет. К тому же все антропометрические данные (рост, вес, сила кисти) по показаниям измерительных приборов вводятся медицинским работником в программу вручную. Другим аналогом описываемого комплекса является программно-аппаратный комплекс «Оценка уровня здоровья человека и его образа жизни (ValeoTest)», разработанный Breath Technologies (г. Рязань). Компьютерная программа ValeoTest предназначена для комплексной диагностики и мониторинга основных функциональных, физических и психологических параметров здоровья человека.

Итак, общими недостатками аналогов является то, что они не охватывают все возрастные группы детей, с которыми работает педиатр, не предполагают автоматизированное измерение антропометрических показателей, не включают ряд методик оценки состояния развития ребенка, а также медицинские документы, входящие в историю развития ребенка. Таким образом, аналоги не могут быть использованы для автоматизации работы врача-педиатра.

Предлагаемый программно-аппаратный комплекс является первым в России автоматизированным рабочим местом врача-педиатра, позволяющим работать с детьми всех возрастов и обеспечивающим автоматизацию процесса измерения антропометрических данных [1, 2].

Научно-техническая часть проекта

Врач-педиатр в ходе своей работы сталкивается с множеством проблем, а именно:

(1) неточности и неудобства измерений антропометрических данных;

(2) рутинные операции по ведению медицинской документации с большими затратами времени;

(3) сложность проведения оценок:

(а) потери времени и неудобства работы с бумажными носителями (сложность навигации и поиска);

(б) неудобство работы с данными, имеющими смешанный формат представления, в основном табличный и текстовый;

(в) необходимость переключаться между различными типами работ (опрос, поиск данных, арифметические операции, оформление документов) и хранить в памяти исходную, промежуточную информацию и результат оценки – потери фокуса внимания при оценке и опросе, высокая вероятность врачебных ошибок.

(4) нехватка времени на полноценное общение с пациентом;

(5) отсутствие оперативного обмена информацией между врачами и доступа к мировым ресурсам медицинской информации.

Для изучения существующих проблем, их причин и выработки способов их решения были поставлены следующие задачи исследования:

(1) используя методики сбора фактов, изучить работу врача-педиатра;

(2) на основе собранных фактов и анализа предметной области создать информационную модель предметной области;

(3) разработать модель информационной системы (ИС);

(4) выбрать приемлемые инструментальные средства и реализовать на основе модели автоматизированную информационную систему;

(5) подобрать оборудование и разработать подсистему связи с оборудованием, обеспечивающую автоматизированный ввод получаемых с него данных.

Для решения поставленных задач были использованы следующие методы:

(а) методы сбора фактов – изучение научно-методической литературы, изучение ведущейся документации, наблюдение за работой педиатрического отделения, собеседование и анкетирование, направленные на выявлении и обобщение мнений специалистов-экспертов, использование их опыта и нетрадиционных подходов к анализу деятельности;

(б) методы моделирования: моделирование бизнес-процессов (IDEF0), потоков работ (IDEF3/Work Flow Diagram), потоков данных (Data Flow Diagram), моделирование данных (IDEF1x/Entity-Relation Diagram), объектно-ориентированное моделирование (Object Oriented Modeling).

Для разработки информационной системы выбраны следующие подходы: RUP (Rational Unified Process), RAD (Rapidly Application Development) и компонентный подход. Выбор таких подходов определяется средой разработки, имеющей научно-образовательный характер, где неизбежно происходит изменение числа и состава участников проекта. Такой выбор оправдывается также с позиции использования накопленных ресурсов для решения в дальнейшем задач автоматизации деятельности широкого круга медицинских специалистов.

Предметная область изучалась на основе длительного наблюдения за работой педиатрического отделения МЛПУ «Городская детская поликлиника имени В. Коваля» г. Тамбова, анализа справочных материалов руководства участкового врача [2], медицинских документов, входящих в «Историю развития ребенка» (учетн. ф. №112), а также систематических консультаций ведущих врачей-педиатров Тамбовской области.

На основе собранных фактов была построена информационная модель предметной области, включающая описание бизнес-процессов, потоков работ, потоков данных, концептуальную модель данных и объектно-ориентированную модель, созданных при помощи CASE-средств: Sybase PowerDesigner 12.0, CA BPwin 4.0.

Информационная модель предметной области легла в основу моделирования информационной системы, для описания которой использовался унифицированный язык моделирования (Unified Modeling Language, UML), предлагающий широкий спектр диаграмм, позволивших зафиксировать специфику и аспекты информационной системы (ИС) для данной предметной области.

Основными средствами, отвечающим описанным подходам разработки ИС, явились платформа Microsoft .Net и Microsoft Visual Studio 2005. Здесь важно отметить доступность и возможность бесплатного использования в научно-образовательной среде. Также используются CASE-средства, MS SQL Server Express 2005 и ряд компонентов для платформы .NET: IdeaBlade DevForce Express, Developer Express .Net.

Основой пользовательского интерфейса комплекса являются специальные формы, созданные на базе широко распространенных медицинских документов, входящих в карточку «История развития ребенка». Вместе с тем был учтен опыт врачебной деятельности: в систему добавлены элементы, которые не были предусмотрены в стандартных документах. Выбранный набор инструментов позволяет без труда создавать интуитивно простые в работе и настройке пользовательские формы. Их особенностью является возможность осуществлять настройку расположения и группировки элементов формы в период выполнения программы с последующим сохранением вариантов схемы размещения. Такая возможность выносит за рамки программирования процесс улучшения эргономических показателей программного обеспечения для конечного пользователя – врача, оставляя за ним право изменения отдельных элементов интерфейса в соответствии с индивидуальными потребностями.

В интерфейсе была реализована попытка свести к минимуму ошибки при работе пользователя с ИС: при вводе и сохранении проходят необходимые варианты верификации данных. Для передачи сообщений пользователю со стороны ИС во многих случаях использованы всплывающие подсказки, которые наилучшим образом подходят для этих целей. Они привлекают внимание, однако не забирают при этом фокус ввода.

Описание ряда характеристик, являющихся базовыми для многих оценок состояния здоровья и развития ребенка, проводилось ранее с помощью указания чисел из некоторого диапазона. Числа представляют степень выраженности описываемой характеристики, при этом их абсолютное значение не несет в себе значимой информации. В предлагаемой ИС использован способ описания с помощью естественного языка («выраженный», «невыраженный», «нормальный», «в большей степени», «в меньшей степени» и т.д.), что будет являться более информативным для врача. У него появится возможность субъективно установить для описания нечеткие понятия и использовать их впоследствии.

Вследствие того, что оценочные методики основаны на системе баллов, в ИС предусмотрена возможность изменения врачом веса того или иного фактора, участвующего в оценке, на основе личного опыта и с учетом реально наблюдаемой картины.

В разработанную ИС входит реляционная база данных в формате MS SQL Server, содержащая информацию обо всех детях, закрепленных за участковым врачом. Благодаря выбранным подходам создания ИС, не затрагивая ее бизнес-логику, можно перейти на другой технически приемлемый формат БД.

Схема программно-аппаратного комплекса врача-педиатра представлена на рис. 1.

Согласно схеме, в разработанной ИС функционально можно выделить следующие основные подсистемы.

(1) Подсистема ведения электронной истории развития ребенка. Она предназначена для ведения электронного варианта карточки истории развития ребенка, и является базой для работы остальных подсистем.

(2) Подсистема расчета и выдачи рекомендаций позволяет на основе введенных данных оценить состояние здоровья и развития ребенка, помогает врачу составить рекомендации и сформировать заключение.

(3) Подсистема связи с оборудованием предназначена для обеспечения взаимодействия ИС с измерительными приборами. Она включает внешний модуль с пользовательским интерфейсом (виртуальный прибор), связанный с базой данных, и пользовательский интерфейс, встроенный в формы.

(4) Подсистема телемедицинских консультаций позволяет педиатру проводить удаленные консультации на рабочем месте.

(5) Подсистема подготовки печатной документации дает возможность быстро сформировать медицинский документ, который может быть распечатан или сохранен в электронном варианте.

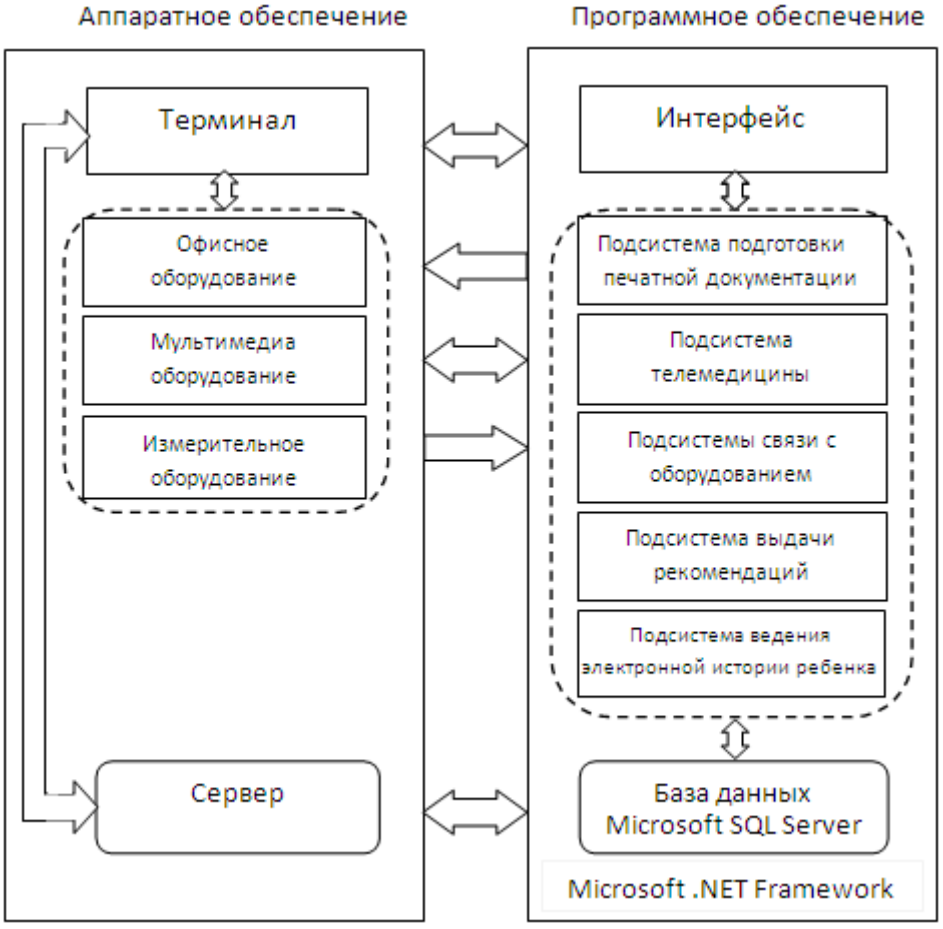


Рис. 1. Схема программно-аппаратного комплекса «Здоровый ребенок»

Основным элементом аппаратного обеспечения комплекса «Здоровый ребенок» является персональный компьютер, к которому подключаются периферийные устройства: ввода-вывода документов (сканер, принтер), электронные измерительные приборы (ростомер, весы, силомер), устройства мультимедиа (микрофон, колонки, вебкамера, проектор).

Устройства ввода-вывода документов используются подсистемой подготовки печатной документации.

Электронные измерительные приборы служат для определения антропометрических данных ребенка и взаимодействуют с ИС посредством подсистемы связи с оборудованием. Приборы являются продукцией Тамбовского приборостроительного завода ОАО «ТВЕС».

Устройства мультимедиа требуются для функционирования подсистемы телемедицинских консультаций.

В информационной системе оценку состояния здоровья и развития ребенка можно классифицировать по следующим группам.

(1) Здоровье новорожденных детей. При первичном врачебно-сестринском патронаже врач составляет генеалогическое дерево, выбирает факторы перинатального риска и факторы, влияющие на развитие патологий у детей. На основе этих данных программа оценивает генеалогический анамнез, определяет степень перинатального риска и направленность риска по развитию патологий. В дальнейшем предполагается ведение учета и реализация факторов риска.

(2) Физическое развитие. Ввод данных производится напрямую с измерительных приборов и отображается непосредственно в формах. Программа сопоставляет антропометрические данные, начиная с момента рождения, с нормами, соответствующими возрасту ребенка; проводит оценку уровня биологической зрелости; выявляет степень гипотрофии ребенка; определяет уровень стигматизации. В результате выводится форма с заключением о физическом развитии ребенка, осуществляется статистика изменения массы и длины тела ребенка с возрастом.

(3) Нервно-психическое развитие (НПР). Оценка НПР проводится по специально разработанным стандартам развития в установленные сроки. Врач регистрирует ответные реакции ребенка на раздражители. Программа осуществляет качественную и количественную оценку НПР.

(4) Резистентность. Комплекс производит оценку степени резистентности по кратности острых заболеваний, перенесенных ребенком в течение определенного периода времени, с выдачей предварительного заключения.

(5) Функциональное состояние органов и систем (ФСО). Уровень ФСО оценивается по результатам клинического осмотра, лабораторных и инструментальных исследований, на основании анализа поведения, а также адаптационных возможностей организма ребенка. Собранные таким образом данные врач заносит в базу данных. Затем в программе осуществляется сравнение показателей основных функциональных систем с нормальными параметрами. Результатом всех перечисленных действий является вывод о развитии ребенка – соответствие всем показателям или отклонения от норм, которые необходимо выявить на ранних стадиях развития.

(6) Группы здоровья. На основе всех перечисленных критериев программа определяет группу здоровья ребенка.

(7) Профилактика состояния здоровья. Комплекс ведет учет и следит за сроками осмотра детей врачами-специалистами и сроками лабораторных исследований, специфической профилактикой управляемых инфекционных заболеваний, сроками инкубационного периода.

(8) Готовность ребенка при поступлении в детские учреждения. Эта оценка включает прогноз адаптационных способностей ребенка в зависимости от возраста, проведение психофизиологического обследования, выявление дефектов звукопроизношения, необходимые при поступлении ребенка в детские учреждения.

Представленные группы формируются на основе информативных модулей, обобщенная схема которых представлена на рис. 2.

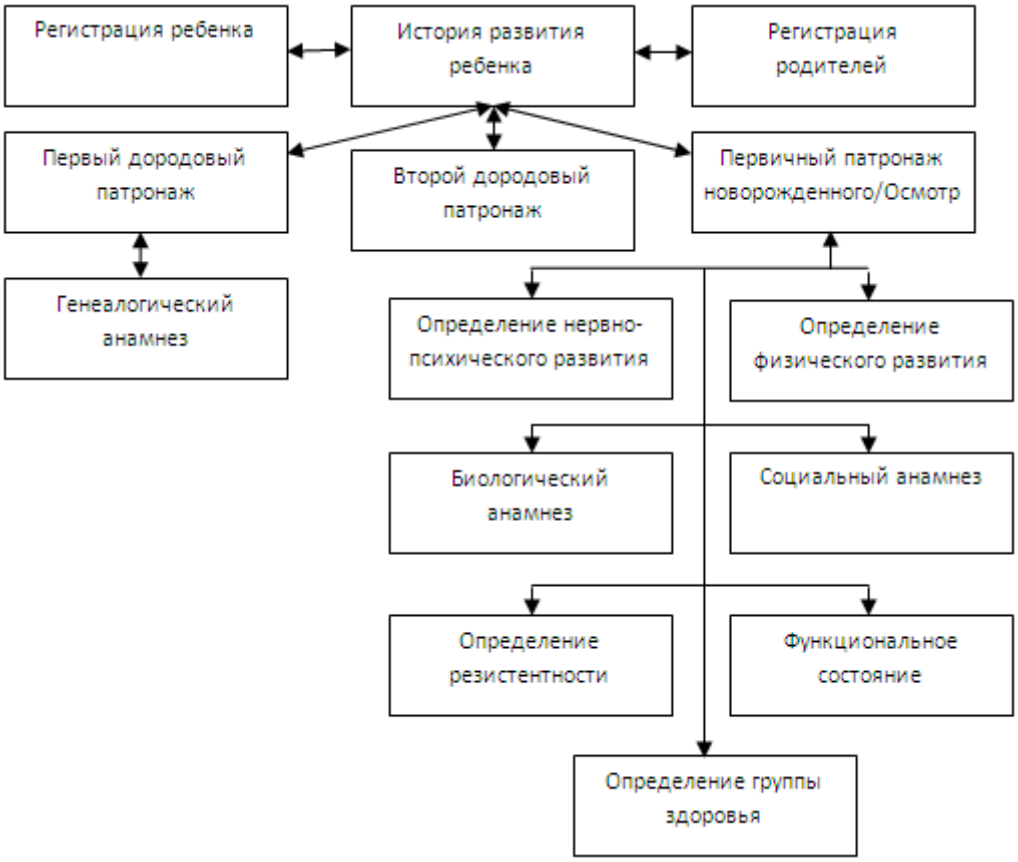


Рис. 2. Информативные модули программно-аппаратного комплекса «Здоровый ребенок»

Практическая значимость

Экспериментальный образец программно-аппаратного комплекса «Здоровый ребенок» внедрен в МЛПУ «Городская детская поликлиника имени В. Коваля» г. Тамбова. Врачи отмечают положительный эффект от внедрения, в первую очередь, в плане экономии времени, соблюдения медицинских стандартов, улучшения качества ведения медицинской документации, точности и актуальности отчетной документации.

Проведем сравнение определения физического развития ребенка врачом «вручную» и при использовании комплекса.

Пример: девочка 5 лет. Длина тела 112 см, масса тела 20,5 кг, окружность грудной клетки 53 см, мышечная сила (МС) правой кисти 6 кг, МС левой кисти 6 кг, жизненная емкость легких 700 мл, за последний год выросла на 7 см, коэффициент ОГ/ДТ 47%, у девочки 2 постоянных зуба, «Филиппинский тест» положительный.

Наиболее информативным методом оценки физического развития является комплексная оценка, позволяющая определить уровень биологического развития индивидуума и степень гармоничности его морфофункционального статуса.

Рассмотрим действия врача по оценке физического развития ребенка «вручную».

По средним значениям показателей биологического развития девочек врач-педиатр находит в нескольких таблицах, что по длине тела, по годовой прибавке длины тела, коэффициенту ОГ/ДТ, числу постоянных зубов и «Филиппинскому тесту» биологическое развитие девочки соответствует возрасту.

Далее, по соотношению основных показателей физического развития у детей, на основании шкал регрессии по длине тела, представленных в справочном руководстве в виде таблицы, врач устанавливает, что при длине тела 112 см масса тела должна находиться в пределах 18,6–22,0 кг, а окружность грудной клетки – 52,1–56,6 см.

Показатели обследуемого ребенка находятся в диапазоне указанных величин. Врачу-педиатру необходим также документ, по которому оцениваются функциональные показатели. Врачом устанавливается, что они находятся в пределах средних (Р25–Р75) центильных величин. Таким образом, получен результат: биологическое развитие девочки соответствует календарному возрасту, ее морфофункциональное состояние гармоничное.

При использовании для оценки программно-аппаратного комплекса удается существенно ускорить и упростить представленный выше процесс оценки. Для определения физического развития врачу-педиатру для выбранного ребенка в форме «Осмотр» необходимо выбрать пункт «Определить физразвитие». На экран выводится форма «Физическое развитие ребенка». После получения антропометрических данных ребенка подпрограмма автоматически вычисляет оценку его развития. В описанной ситуации сэкономленное время, в зависимости от степени подготовки врача и его профессиональных навыков, может составлять от одной до трех минут.

Заключение

Итак, такие трудности в работе врача-педиатра, как неточности и неудобства измерений антропометрических данных, рутинные операции по ведению медицинской документации, затраты времени при работе с табличными данными, затраты времени на поиск и доступ к архивным документам, отсутствие оперативного обмена информацией между врачами могут быть решены с помощью программно-аппаратного комплекса «Здоровый ребенок».

В комплексе «Здоровый ребенок» впервые в России одновременно обеспечивается:

(1) автоматизированное измерение антропометрических данных ребенка с помощью подключенных к компьютеру медицинских приборов;

(2) автоматизация подготовки медицинских документов и отчетов;

(3) автоматизация оценки и прогнозирования состояния здоровья и дальнейшего развития детей всех возрастов (от 0 до 18,5 лет);

(4) ведение электронной истории развития ребенка и быстрый, гибкий поиск информации о детях в базе данных;

(5) использование в работе ИС доступной справочной медицинской информации в электронном виде.

Таким образом, благодаря программно-аппаратному комплексу «Здоровый ребенок» автоматизируется и облегчается работа врача-педиатра, снижается вероятность врачебных ошибок, и у врача-педиатра появляется время для полноценного общения с ребенком и его родителями.

Литература

1. Фролов С.В., Фролова М.С. Автоматизированное рабочее место врача-педиатра // 3-й Международный форум MedSoft – 2007. Медицинские информационные технологии. – М., 2007. – С. 66–67.
2. Фролов С.В., Фролова М.С. Создание автоматизированного рабочего места врача-педиатра // Сердечно-сосудистые заболевания. Бюллетень НЦССХ им. Бакулева РАМН. – 2007. – Т.5. – №3. Приложение. – С. 153.
3. Участковый педиатр: Справочное руководство / Под ред. М.Ф. Рзянкиной, В.Г. Молочного. – Ростов на Дону: Феникс, 2005. – 313 с.

АДАПТИРОВАНИЕ И ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ НОВОГО МЕТОДА ХРАНЕНИЯ И СИСТЕМАТИЗАЦИИ ИНФОРМАЦИИ В УФО-АНАЛИЗЕ ПОД СРЕДУ ПРОЕКТИРОВАНИЯ БИЗНЕСА UFO-TOOLKIT

А.Г. Жихарев (Белгородский государственный университет)
Научный руководитель – д.т.н. профессор С.И. Маторин
(Белгородский государственный университет)

В работе рассмотрены адаптация и реализация нового метода хранения и систематизации информации в среде проектирования бизнеса UFO-toolkit. Решается задача адаптации и логического проектирования базы знаний под CASE-инструментарий UFO-toolkit. В результате получена логическая модель базы знаний для хранения УФО-элементов.

Введение

Моделирование бизнеса – одна из наиболее развивающихся сфер информационных технологий, это обусловлено тем, что любое предприятие, фирма и т.д., выходя на рынок, сталкивается с очень серьезной для нее проблемой – конкуренцией. Чтобы преодолеть данную проблему, фирме необходимо непрерывно улучшать свой бизнес, развивать новые отрасли своей деятельности, т.е. проводить непрерывную реорганизацию своего бизнеса. На данном этапе руководителю приходят на помощь так называемые CASE-инструментарии моделирования бизнеса. Одним из таких инструментариев является среда проектирования бизнеса UFO-toolkit [6].

Данный инструмент в своей основе имеет оригинальный системный подход к проектирований бизнеса «узел–функция–объект» [7]. Одним из наиболее перспективных отличий данного инструментария от ему подобных является частичная автоматизация построения УФО-диаграмм за счет хранения ранее созданных элементов такой диаграммы (УФО-элементов) в специальных библиотеках – репозитариях. Но существуют некоторые проблемы, связанные с данной библиотекой [1]. Для их решения разработан метод хранения и систематизации информации, основанный на классификации УФО-элементов по типу и объекту их деятельности [2], которая представлена на рис. 1.

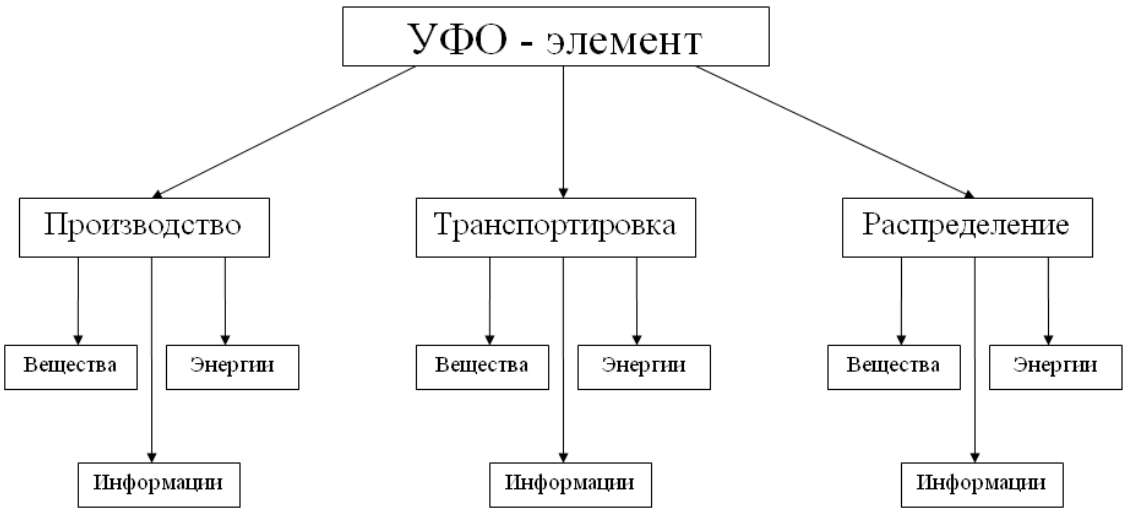


Рис. 1. Классификация УФО-элементов

На основе данной классификации (см. рис.1) была разработана база знаний (БЗ) [4], которая позволяет хранить УФО-элементы, осуществлять их редактирование и вы-

борку. Но БЗ была ориентированна на сам подход, т.е. она позволяет хранить лишь неотъемлемую информацию об узле, функции и объекте, соответственно. Требуется адаптация БЗ непосредственно под инструментарий UFO-toolkit. Прежде чем приступить к проектированию и реализации, определимся с требованиями к БЗ.

Требования к базе знаний UFO-toolkit

Как было сказано выше, на данный момент БЗ позволяет хранить лишь неотъемлемую информацию УФО-элемента. Инфологическая схема базы знаний показана на рис. 2.

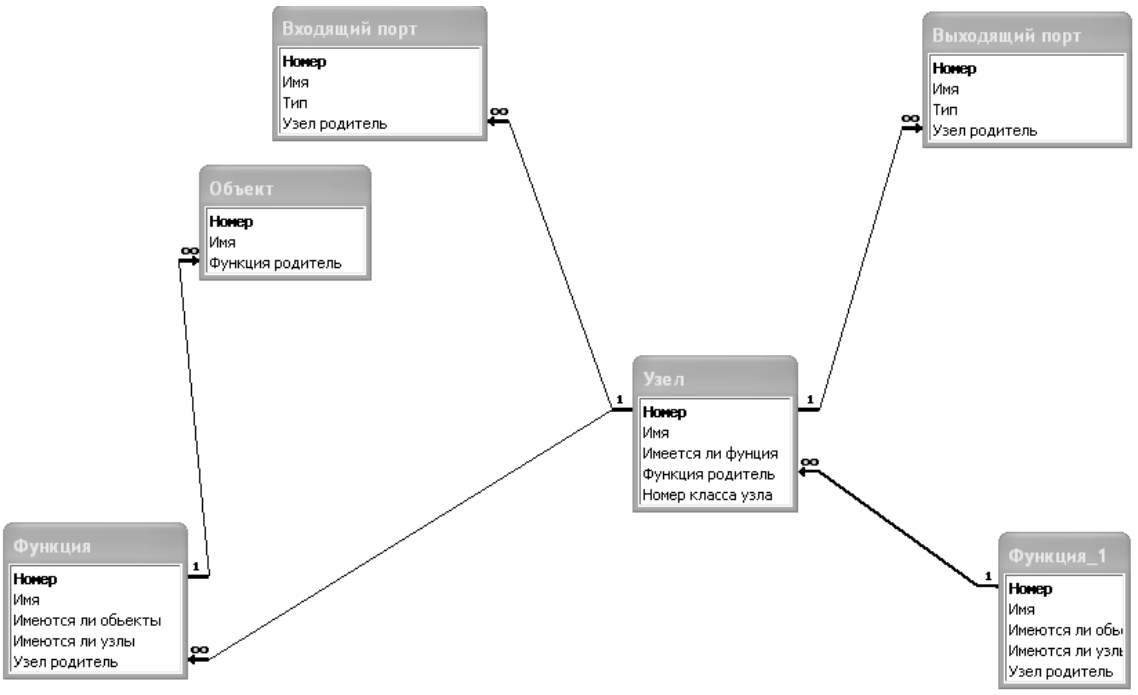


Рис. 2. Инфологическая модель базы знаний UFO-toolkit

Для внедрения такой БЗ в инструмент необходима дополнительная информация об узлах, функциях, объектах и портах. Так как в среде UFO-toolkit соответствующие УФО-элементы отображаются на экране в виде диаграмм, то необходимо также хранить информацию о расположении портов на УФО-элементе. С первого взгляда можно подумать, что нужно хранить информацию о расположении самого УФО-элемента на диаграмме, однако при его внедрении в проект заранее неизвестно, куда он будет помещен. Эти параметры удобнее вычислить при его вставке в зависимости от расположения свободных портов [6], именно поэтому данные характеристики хранить в БЗ нет смысла. При выборке УФО-элемента из БЗ он анализируется на процентное соответствие свободным портам. Этот анализ проводится с использованием таких характеристик, как количество входящих и выходящих портов, поэтому, чтобы не вычислять эти переменные при анализе, имеет смысл хранить соответствующие характеристики в БЗ.

Проанализировав требования к базе знаний УФО-элементов, можно приступить к ее проектированию и разработке инфологической модели.

Порты УФО-элементов

Каждый УФО-элемент имеет входящие и выходящие порты, которые хранятся в таблицах. Для порта (как входящего, так и выходящего) необходимо хранить следующую информацию:

1. идентификатор порта;
2. имя порта;
3. тип порта;
4. идентификатор узла, которому принадлежит данный порт;
5. идентификатор визуального стиля;
6. тип связи;
7. позиция.

В данном случае существует проблема, связанная с возможностью повторения идентификатора порта, так как таблица будет хранить входящие и выходящие порты всех УФО-элементов. Вполне возможно, что в один УФО-элемент, например, будут входить несколько одинаковых связей, и идентификаторы у них будут одинаковы, так как они берутся из таблицы иерархии связей проекта. В данном случае можно добавить в рассматриваемую таблицу поле целого типа, которое будет увеличиваться на единицу при попытке занесения в таблицу записи с существующим идентификатором, т.е. будут учитываться все связи, даже с одинаковым идентификатором. Итоговое изображение логической модели входящего/выходящего порта представлено на рис. 3.

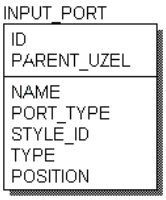


Рис. 3. Входящий порт

Что же касается таблицы «Выходящий порт», то она абсолютно идентична вышеприведенной, изменяется только название таблицы.

Иерархия связей

В БЗ имеется таблица, которая хранит все возможные связи, созданные в рамках одного проекта (рис. 4).

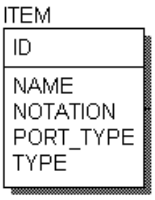


Рис. 4. Иерархия связей

Таблица, представленная на рис. 4, является родительской по отношению к таблицам «Входящий/выходящий порт». Так как каждый входящий/выходящий порт является «экземпляром» какой-либо связи из общей их иерархии, то между ними будет проходить идентифицирующая связь, вследствие чего идентификатор связи мигрирует в первичный ключ входящего/выходящего порта.

Узел

Каждый узел может иметь неопределенное количество входящих/выходящих портов, функций и объектов, поэтому таблицы, хранящие информацию об этих состав-

ляющих УФО-элемента, будут дочерними по отношению к таблице «Узел». Данная таблица будет логической моделью, как показано на рис. 5.

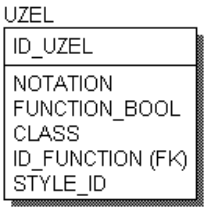


Рис. 5. Таблица Узел

Функция и объект

Функция по отношению к таблице «Узел» является одновременно и дочерней, и как бы родительской, так как любой узел содержит хотя бы одну функцию, в свою очередь, функцию могут реализовывать не только объекты, но и сами узлы могут выступать в роли объектов. Общая инфологическая модель преобразованной БЗ и модели таблиц «Функция» и «Объект» представлены на рис. 6.

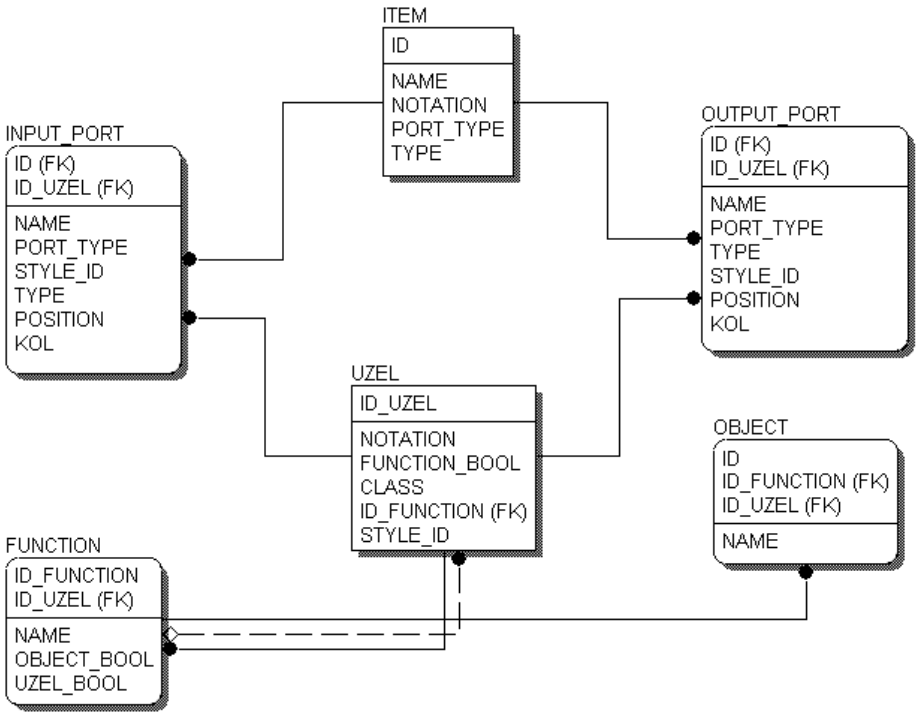


Рис. 6. Инфологическая модель базы знаний UFO-toolkit

Таким образом, из рис. 6 видно, что каждый узел может содержать в себе функции, и функция, в свою очередь, не имеет смысла без узла (идентифицирующая связь на модели), но узел вполне может существовать и без функции.

Заключение

В рамках написания статьи была адаптирована БЗ для хранения информации об УФО-элементе под инструментарий моделирования бизнеса UFO-toolkit. В дальнейшем

планируется разработка программного модуля для работы инструмента с рассмотренной БЗ с возможностью сетевого доступа к ней.

Литература

1. Маторин С.И., Жихарев А.Г. Организация библиотек в CASE – инструментарии моделирования бизнеса «UFO-toolkit» / Материалы VII Международной научно-практической конференции «Компьютерные технологии в науке, производстве, социальных и экономических процессах»: Сб. научных трудов. Ч.3. – Новочеркасск: ЮРГТУ(НПИ). 2006. – С. 23–28.
2. Жихарев А.Г. Проектирование средств хранения и систематизации информации в среде «UFO-toolkit» / Материалы VIII Международной конференции «Кибернетика и высокие технологии XXI века (C&T 2007)»: Том 2. – Воронеж. 2007. – С. 982–989.
3. Великая Я.Г., Жихарев А.Г., Зимовец О.А., Маторин С.И., Тубольцев М.Ф. О перспективах развития технологии моделирования бизнеса «Узел-Функция-Объект» // Научные ведомости БелГУ. – Сер. Информатика и прикладная математика. – 2007. – №3. – Вып.4.
4. Маторин С.И., Зимовец О.А., Жихарев А.Г. О развитии технологии графоаналитического моделирования бизнеса с использованием системного подхода «Узел-Функция-Объект» // НТИ. Сер. 2. – 2007. – № 11. – С. 10–17.
5. Маторин С.И., Зимовец О.А., Жихарев А.Г. Технология информационного обеспечения управления на основе системного подхода «Узел-Функция-Объект» // Вестник ХГПУ. Новые решения в современных технологиях. 2007 (в печати).
6. Маторин С.И., Попов А.С. «UFO-toolkit» – BI-инструментарий нового поколения [Электронный ресурс]. – Электрон. дан.(1 файл). – Режим доступа: http://citforum.ru/consulting/BI/UFO/. – Последнее обращение: 25.10.2005. – Загл. с экрана.
7. Маторин С.И., Ельчанинов Д.Б., Зиньков С.В., Маторин В.С. Синтез и анализ систем в свете подхода «Узел-Функция-Объект» // НТИ. – Сер. 2. – №8. – М.: ВИНИТИ, 2006. – С. 10–16.

ИСПОЛЬЗОВАНИЕ МЕТОДА КОРРЕЛЯЦИОННЫХ ПЛЕЯД ДЛЯ ИЗУЧЕНИЯ ИНФОРМАЦИОННЫХ ХАРАКТЕРИСТИК УЧЕБНЫХ ТЕКСТОВ

М.М. Невдах (Белорусский государственный технологический университет)
Научный руководитель – д.х.н., профессор М.А. Зильберглейт
(Белорусский государственный технологический университет)

В статье проанализированы информационные характеристики учебных текстов с использованием эвристического метода корреляционных плеяд. В частности, изучена связь между 49 параметрами текстов по экономической теории для высшей школы на основе выделения корреляционных плеяд с помощью различных формальных приемов. В дальнейшем полученные результаты будут учитываться при построении решающего правила разбиения.

Введение

Проблемам, связанным с исследованием влияния информационных характеристик текста на его читабельность[1], посвящен ряд работ [2–14], в которых основное внимание уделяется небольшому числу параметров текста, включающих обычно среднюю длину предложения в словах, среднюю длину слов в слогах, процент конкретных/абстрактных существительных, процент незнакомых слов, среднюю частоту повторения слова. Только в отдельных работах [15, 16] число исследуемых характеристик текста, влияющих на его читабельность, превышает 100 признаков. Систематических исследований, посвященных изучению влияния значительного числа параметров текста на его усвоение, до настоящего времени не проводилось.

В принципе, текст можно представить как объект, характеризующийся многомерным вектором, состоящим из различного рода переменных. В связи с этим он может быть исследован методами многомерного статистического анализа.

В данной работе методами кластерного анализа изучены характеристики 16-ти учебных текстов по экономической теории для высшей школы [17–20]. Объем выборки составил 1800–2000 печатных знаков. Выбор данной величины обусловлен тем, что в [21] показано, что, начиная с объема в 1800 печатных знаков, статистические характеристики текста становятся относительно постоянными.

В качестве переменных были выбраны 49 признаков текста. Очевидно, что использование такого большого числа характеристик для практических целей невозможно. В первую очередь это связано с тем, что данные параметры могут быть сильно коррелированны. С другой стороны, ничем не оправданное уменьшение числа переменных может привести к потере точности экспериментов.

В прикладной статистике рассматриваются различные методы сокращения размерности признакового пространства. В проведенных ранее исследованиях с использованием методов кластерного и факторного анализа все характеристики исследуемых текстов были разбиты на девять и восемь групп соответственно, в которые объединились близкие параметры. Сделать выводы о природе связей между явлениями позволяет и метод корреляционных плеяд, впервые предложенный П.В. Терентьевым [22, 23]. Таким образом, цель работы – использование данного метода для анализа информационных характеристик текста.

[1] Под читабельностью понимают некоторую характеристику текста, зависящую от всех элементов внутри данного материала, которые влияют на успешность его усвоения определенной группой читателей. Мерой такого успешного усвоения является то, насколько средний читатель интересующей группы понимает исследуемый материал, в какой мере скорость, с которой он его читает, приближается к оптимальной, и какой интерес представляет данный материал для читателя [1].

Выделение корреляционных плеяд

Как известно, метод корреляционных плеяд предназначен для нахождения таких групп признаков (плеяд), в которых корреляционная связь между параметрами одной группы (внутриплеядная связь) велика, а связь между параметрами из разных групп (межплеядная связь) мала [24]. По определенному правилу по корреляционной матрице признаков образуют граф, который затем с помощью различных приемов разбивают на подграфы. Элементы, соответствующие каждому из подграфов, и образуют плеяду.

Для вычисления корреляционной матрицы были использованы следующие параметры учебных текстов:

(1) длина текста в абзацах;
(2) длина текста в словах;
(3) длина текста в буквах;
(4) средняя длина абзаца в фразах;
(5) средняя длина абзаца в словах;
(6) средняя длина абзаца в буквах;
(7) средняя длина абзаца в печатных знаках;
(8) средняя длина предложения в фразах;
(9) средняя длина предложения в словах;
(10) средняя длина предложения в слогах;
(11) средняя длина предложения в буквах;
(12) средняя длина предложения в печатных знаках;
(13) средняя длина самостоятельного предложения в фразах;
(14) средняя длина самостоятельного предложения в словах;
(15) средняя длина самостоятельного предложения в слогах;
(16) средняя длина самостоятельного предложения в буквах;
(17) средняя длина самостоятельного предложения в печатных знаках;
(18) средняя длина фразы в словах;
(19) средняя длина фразы в слогах;
(20) средняя длина фразы в буквах;
(21) средняя длина фразы в печатных знаках;
(22) средняя длина слов в слогах;
(23) средняя длина слов в буквах;
(24) средняя длина слов в печатных знаках;
(25) средняя длина слов по Деверу;
(26) процент слов длиной в 5 букв и больше;
(27) процент слов длиной в 6 букв и больше;
(28) процент слов длиной в 7 букв и больше;
(29) процент слов длиной в 8 букв и больше;
(30) процент слов длиной в 9 букв и больше;
(31) процент слов длиной в 10 букв и больше;
(32) процент слов длиной в 11 букв и больше;
(33) процент слов длиной в 12 букв и больше;
(34) процент слов длиной в 13 букв и больше;
(35) процент слов в 3 слога и больше;
(36) процент слов в 4 слога и больше;
(37) процент слов в 5 слогов и больше;
(38) процент слов в 6 слогов и больше;
(39) процент неповторяющихся слов;
(40) средняя частота повторения слова;
(41) процент неповторяющихся существительных;

(42) процент повторяющихся существительных;
(43) процент конкретных существительных;
(44) процент абстрактных существительных;
(45) процент прилагательных;
(46) процент глаголов;
(47) процент сложных предложений;
(48) процент простых предложений;
(49) процент придаточных предложений среди фраз.

Следует сделать несколько уточнений. Под термином «фраза» в данной статье понимается отрезок текста, в котором содержится одна предикативная связь. Таким образом, к фразе относятся простое предложение, части сложносочиненного предложения, главное и придаточное предложения в сложноподчиненном предложении. Самостоятельным предложением считаются простые предложения, части сложносочиненного предложения и сложноподчиненное в целом. Впервые такую единицу текста использовал Р. Флеш в [9]. Средняя длина слов по Деверу рассчитывалась делением общего количества знаков с пробелами на число знаков без пробелов.

Фрагмент корреляционной матрицы для исследуемых характеристик текста представлен в табл. 1.

№ п/п	1	2	3	4	5	6	7	8	9	…	49
1	1,000	-0,442	-0,476	-0,740	-0,901	-0,967	-0,967	-0,488	-0,613	…	-0,341
2	-0,442	1,000	0,504	0,638	0,661	0,496	0,502	0,700	0,654	…	0,412
3	-0,476	0,504	1,000	0,165	0,339	0,491	0,477	0,226	0,410	…	0,400
4	-0,740	0,638	0,165	1,000	0,859	0,738	0,753	0,766	0,569	…	0,157
5	-0,901	0,661	0,339	0,859	1,000	0,945	0,951	0,702	0,766	…	0,390
6	-0,967	0,496	0,491	0,738	0,945	1,000	0,999	0,555	0,707	…	0,394
7	-0,967	0,502	0,477	0,753	0,951	0,999	1,000	0,572	0,716	…	0,393
8	-0,488	0,700	0,226	0,766	0,702	0,555	0,572	1,000	0,870	…	0,391
9	-0,613	0,654	0,410	0,569	0,766	0,707	0,716	0,870	1,000	…	0,583
…	…	…	…	…	…	…	…	…	…	…	…
49	-0,341	0,412	0,400	0,157	0,390	0,394	0,393	0,391	0,583	…	1,000

Таблица 1. Корреляционная матрица исходных признаков

Выделение корреляционных плеяд осуществляется с помощью двух формальных методов. Суть первого метода заключается в том, что граф, представляющий собой изучаемые признаки, связанные ребрами с соответствующим значением коэффициента корреляции, разбивается таким образом, что при выбранном некотором пороговом значении коэффициента корреляции r_0 из него исключаются ребра, которые соответствуют коэффициентам корреляции, по модулю меньшим r_0.

В связи с этим существенным для техники выделения корреляционных плеяд является выбор r_0. Используя прямое (z) и обратное (z^{-1}) преобразования Фишера, можно определить r_0 для заданного объема выборки: $| r_0 | = | z_{\alpha, \nu} |^{-1}$, где $| z_{\alpha, \nu} | = t_{\alpha, \nu} / (\sqrt{n-3})$; $\nu = n - 1$; $\alpha \le 0{,}01$.

Для заданного объема выборки ($n = 16$) получили $r_0 = 0{,}5$. При данном пороговом значении коэффициента корреляции все параметры сохранились, что не привело к снижению размерности признакового пространства. Очевидно, что в этом случае следует задать такие значения r_1, r_2, … r_n, при которых произойдет заметное сокращение информационных характеристик текста. Например, если задать шаг «расслоения» 0,1, то при $r_1 \ge 0{,}6$ граф распадается на три подграфа с исключением признаков 45 и 46, при $r_2 \ge 0{,}7$ – на четыре подграфа с исключением признаков 41, 45 и 46, при $r_3 \ge 0{,}8$ – на

пять подграфов с исключением признаков 2, 3, 41, 43, 45, 46 и 49, при $r_4 \geq 0{,}9$ – на восемь подграфов с исключением признаков 1–4, 8–10, 13, 14, 24, 33, 34, 38, 41–46 и 49. Из-за большого числа связей изображать плеяды для низких коэффициентов корреляции представляется нецелесообразным. Для наглядности выделим плеяды для значений $r \geq 0{,}85$ (рис. 1).

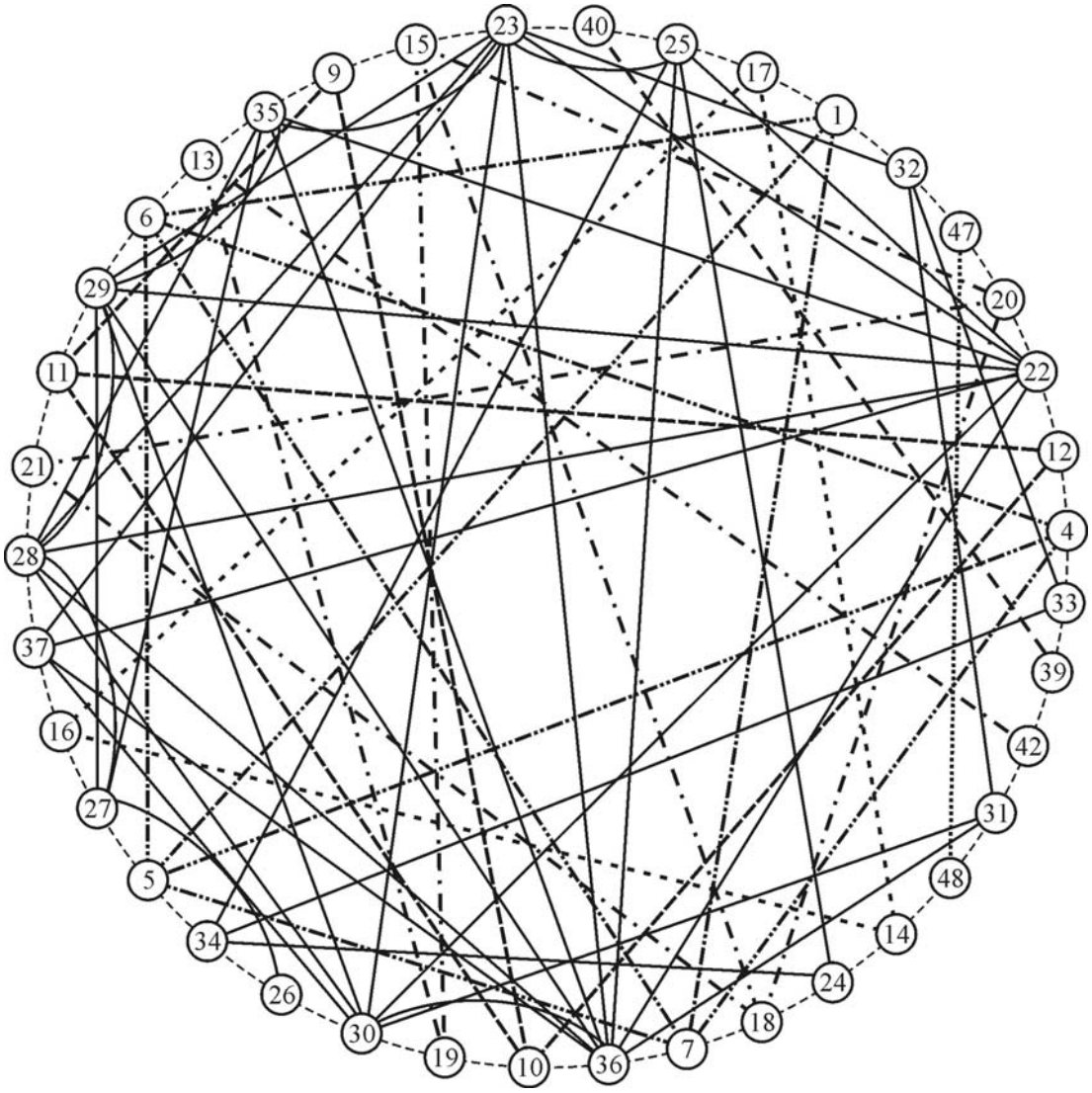


Рис. 1. Корреляционные плеяды при $r \geq 0{,}85$

Использование первого метода выделения корреляционных плеяд позволило обнаружить наиболее связанные признаки, которые образуют несколько групп.

Первая группа. Признаки 1, 4–7 – длина текста в абзацах, средняя длина абзаца во фразах, средняя длина абзаца в словах, средняя длина абзаца в буквах и средняя длина абзаца в печатных знаках.

Вторая группа. Признаки 9–12 – средняя длина предложения в словах, средняя длина предложения в слогах, средняя длина предложения в буквах и средняя длина предложения в печатных знаках.

Третья группа. Признаки 14, 16, 17 – средняя длина самостоятельного предложения в словах, средняя длина самостоятельного предложения в буквах и средняя длина самостоятельного предложения в печатных знаках.

Четвертая группа. Признаки 13, 15, 18–21, 42 – средняя длина самостоятельного предложения во фразах, средняя длина самостоятельного предложения в слогах, сред-

няя длина фразы в словах, средняя длина фразы в слогах, средняя длина фразы в буквах, средняя длина фразы в печатных знаках, процент повторяющихся существительных.

Пятая группа. Признаки 22–37 – средняя длина слов в слогах, средняя длина слов в буквах, средняя длина слов в печатных знаках, средняя длина слов по Деверу, процент слов длиной в 5 букв и больше, процент слов длиной в 6 букв и больше, процент слов длиной в 7 букв и больше, процент слов длиной в 8 букв и больше, процент слов длиной в 9 букв и больше, процент слов длиной в 10 букв и больше, процент слов длиной в 11 букв и больше, процент слов длиной в 12 букв и больше, процент слов длиной в 13 букв и больше, процент слов в 3 слога и больше, процент слов в 4 слога и больше, процент слов в 5 слогов и больше.

Шестая группа. Признаки 39 и 40 – процент неповторяющихся слов и средняя частота повторения слова.

Седьмая группа. Признаки 47 и 48 – процент сложных предложений и процент простых предложений.

Сравнение данных, полученных ранее с помощью методов кластерного и факторного анализа, показало, что наиболее устойчивыми являются первая, вторая, третья и четвертая группы. При этом следует отметить одну особенность: при использовании кластерного анализа очень часто признаки 11 и 12 (средняя длина предложения в буквах и средняя длина предложения в печатных знаках), 16 и 17 (средняя длина самостоятельного предложения в буквах и средняя длина самостоятельного предложения в печатных знаках), 20 и 21 (средняя длина фразы в буквах и средняя длина фразы в печатных знаках) выделяются отдельно. Признаки 10 (средняя длина предложения в слогах), 15 (средняя длина самостоятельного предложения в слогах) и 19 (средняя длина фразы в слогах) входят в другую группу. Но, с точки зрения здравого смысла, увеличение длины предложения предполагает одновременное увеличение в нем количества слогов, букв и печатных знаков, что и зафиксировано в методе корреляционных плеяд.

В другом варианте корреляционных плеяд признаки упорядочиваются и рассматриваются только те коэффициенты корреляции, которые соответствуют связям между элементами в упорядоченной системе.

№ п/п	47	48	46	49	18	20	21	15	19	…	3
47	1,000	-1,000	0,578	0,540	0,048	-0,100	-0,114	0,011	0,020	…	0,184
48	-1,000	1,000	-0,578	-0,540	-0,048	0,100	0,114	-0,011	-0,020	…	-0,184
46	0,578	-0,578	1,000	0,080	-0,422	-0,451	-0,477	-0,251	-0,215	…	-0,120
49	0,540	-0,540	0,080	1,000	0,726	0,499	0,493	0,528	0,536	…	0,121
18	0,048	-0,048	-0,422	0,726	1,000	0,936	0,932	0,852	0,814	…	0,148
20	-0,100	0,100	-0,451	0,499	0,936	1,000	0,995	0,873	0,826	…	0,197
21	-0,114	0,114	-0,477	0,493	0,932	0,995	1,000	0,846	0,799	…	0,236
15	0,011	-0,011	-0,251	0,528	0,852	0,873	0,846	1,000	0,988	…	0,141
19	0,020	-0,020	-0,215	0,536	0,814	0,826	0,799	0,988	1,000	…	0,110
…	…	…	…	…	…	…	…	…	…	…	…
3	0,184	-0,184	-0,120	0,121	0,148	0,197	0,236	0,141	0,110	…	1,000

Таблица 2. Упорядоченная корреляционная матрица исходных признаков

Упорядочение производится на основании принципа максимального корреляционного пути: все p признаков связываются при помощи $(p - 1)$ линий так, чтобы сумма модулей коэффициентов корреляции была максимальной. Это достигается следующим образом: выбираются два наиболее сопряженных признака, для чего в матрице коэф-

фициентов корреляции определяется максимальное значение r_{ij}, $i \neq j$; признаки i и j образуют две первые вершины графа (в нашем случае признаки 47 и 48, $r_{47,\,48} = 1$). Далее в строках i и j находится следующий наиболее сопряженный признак, образующий новую третью вершину, соединенную с вершиной j. Эта процедура повторяется до тех пор, пока не будут задействованы все признаки.

После проведения описанной процедуры для удобства построения графа была составлена упорядоченная корреляционная матрица, фрагмент которой представлен в табл. 2. На основании упорядоченной корреляционной матрицы был построен граф, представленный на рис. 2.

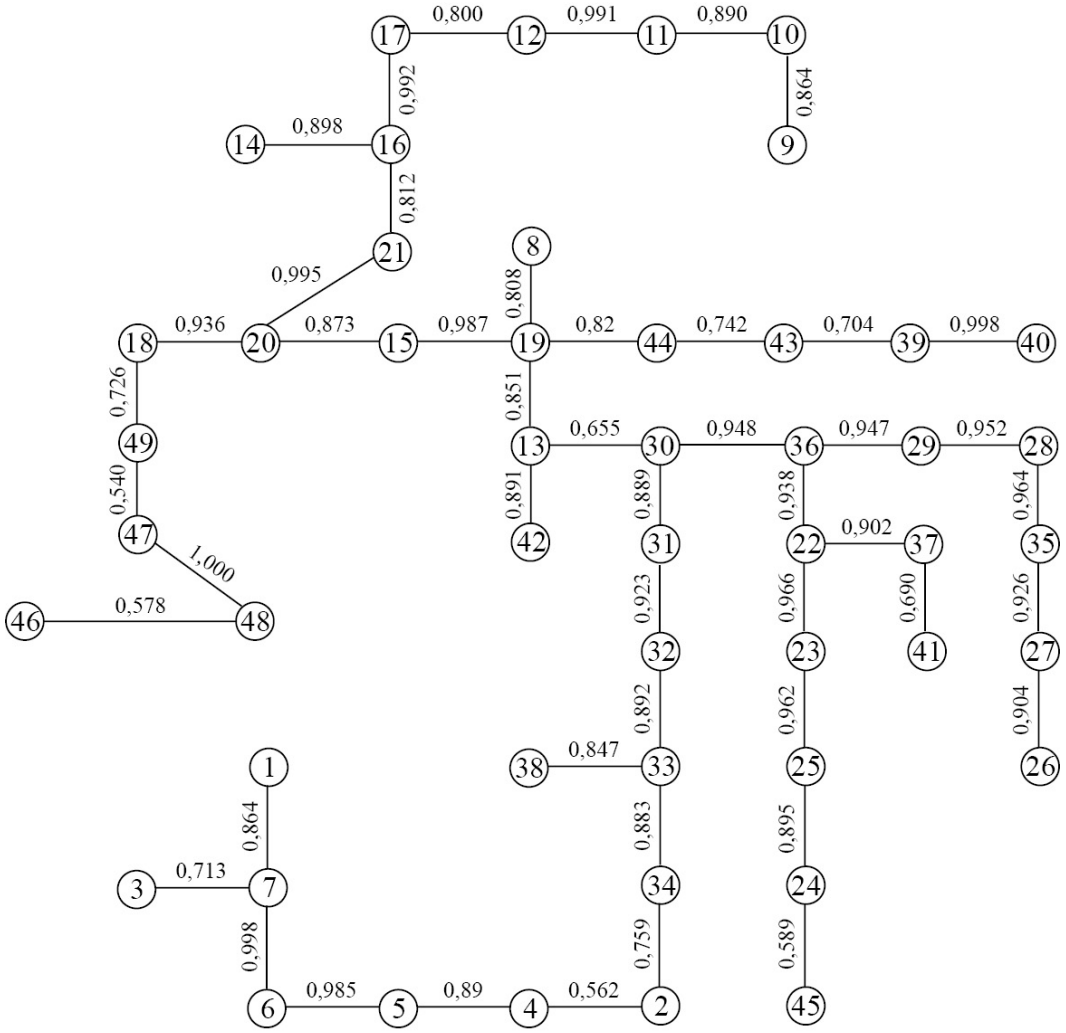


Рис. 2. Граф максимального корреляционного пути

Данный граф представляет собой кратчайший незамкнутый путь, в котором соединены все исследуемые параметры текстов. Если задать определенное пороговое значение коэффициента корреляции (r_0), то полученный граф максимального корреляционного пути можно разбить на подграфы (плеяды), проводя разрыв между теми признаками со значением сопряженности, меньшим r_0.

Исходя из поставленной цели и анализа корреляционной матрицы исследуемых характеристик текста, был задан пороговый коэффициент корреляции $r \geq 0{,}85$. Исходный граф распался на семь подграфов (рис. 3), что позволило выявить наиболее связанные друг с другом признаки.

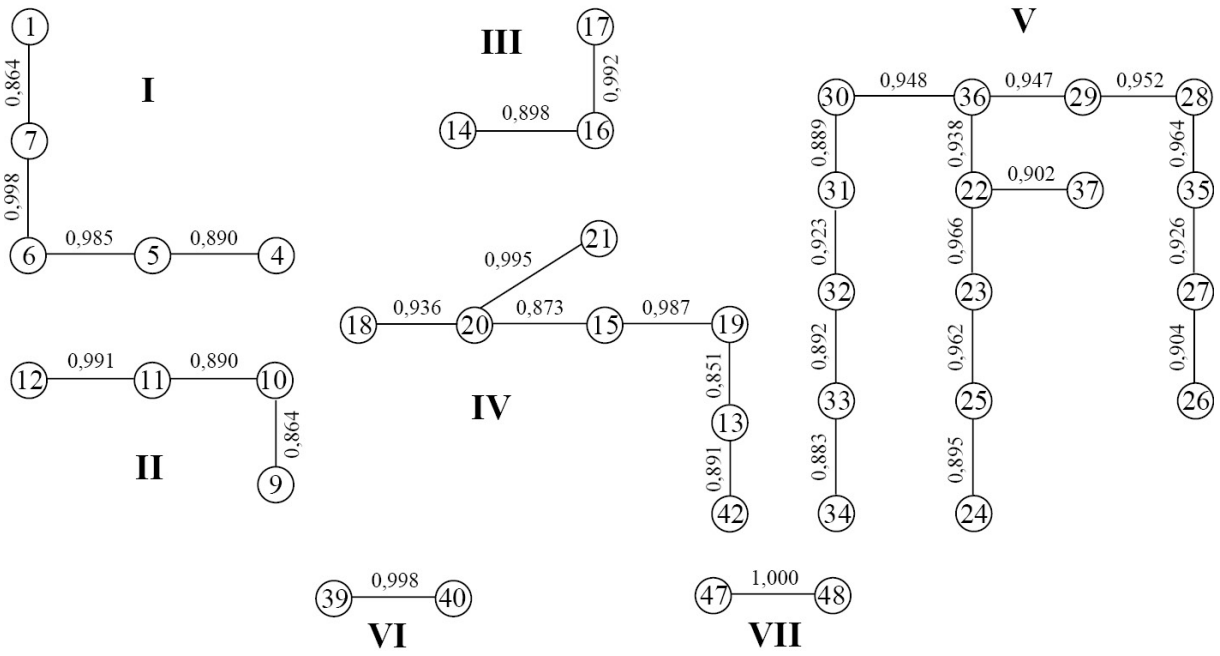


Рис. 3. Корреляционные плеяды при пороговом значении коэффициента корреляции $r \geq 0{,}85$

Сравнение корреляционных плеяд, выделенных на основе двух формальных методик, показало их идентичность.

Заключение

Не претендуя на содержательную интерпретацию полученных результатов, следует отметить, что в выделенных группах сумма модулей коэффициентов корреляции между параметрами достаточно велика, что с определенной долей уверенности дает нам право сделать следующий вывод: в последующую обработку достаточно включить лишь один признак из каждой группы. Например, можно использовать признак «длина текста в абзацах» – из первой группы; признак «средняя длина предложения в буквах» – из второй; признак «средняя длина самостоятельного предложения в буквах» – из третьей; признак «средняя длина фразы в буквах» – из четвертой; признак «средняя длина слов в слогах» – из пятой; признак «процент неповторяющихся слов» – из шестой и признак «процент сложных предложений» – из седьмой. Остальные признаки, не вошедшие в выделенные семь групп, требуют дальнейшего исследования.

Литература

1. Dale E. The concept of readability / E. Dale, J. S. Chall // Elementary English. – 1949. – № 26. – P. 23.
2. Крсманович М. Методы исследования удобочитаемости учебных текстов / М. Крсманович // Проблемы школьного учебника». – М., 1974 – Вып. 2. – С. 108–120.
3. Матеева А.Д. О задаче прогнозирования трудности понимания текста по объективным языковым характеристикам / А.Д. Матеева // Сборник научных трудов МГПИИЯ. – Вып. 124. – М., 1977. – С. 60–69.

4. Мацковский М.С. Проблемы читабельности печатного материала. / М.С. Мацковский // Смысловое восприятие речевого сообщения (в условиях массовой коммуникации). – М., 1976. – С. 126–142.
5. Микк Я.А. Применение формул читабельности к русскому тексту. / Я.А. Микк // Ученые записки Тартуского государственного университета. – Тарту, 1977. – Вып. 415. – С. 94–102.
6. Тулдава Ю. Об измерении трудности текстов / Ю. Тулдава // Ученые записки Тартуского университета. – 1975. – Вып. 345, IV. Труды по методике преподавания иностранных языков. – С. 102–120.
7. Bormuth J.R. Development of readability analysis / J.R. Bormuth. – Washington D.C.: U.S. Office of Education, Bureau of Research, U.S. Department of Health, Education, and Welfare, 1969.
8. Chall, J.S. Readability: an appraisal of research and application / J.S. Chall. – Columbus, OH: Ohio State University Press, 1958.
9. Flesch R. The art of readable writing / R. Flesch. – New York: Harper, 1949.
10. Fry, E.B. The readability graph validated at primary levels / E.B. Fry // The reading teacher. – 1969. № 22. – P. 534–538.
11. Klare, G.R. The measurement of readability / G.R. Klare. – Ames, Iowa: Iowa State University Press, 1963.
12. Paul, T. Guided Independent Reading / T. Paul. – Madison, WI: School Renaissance Institute, 2003.
13. Stenner, A.J., The objective measurement of reading comprehension in response to technical questions raised by the California department of education technical study group / A.J Stenner, D.S. Burdick. – Durham, NC: MetaMetrics, Inc., 1997.
14. Weaver, B.M. Leveling books K-8: Matching readers to text / B.M. Weaver. – Newark, DE: International Reading Association, 2000.
15. Микк Я.А. Методика разработки формул читабельности / Я.А. Микк // Советская педагогика и школа. – Тарту, 1974. – Вып. 9. – С. 78–163.
16. Gray, W.S. What makes a book readable / W.S. Gray, B. Leary. – Chicago: Chicago University Press, 1935.
17. Экономическая теория: учебное пособие / Л.Н. Давыденко, А.И. Базылева, А.А. Дичковский и др.; под общ. ред. Л.Н. Давыденко. – Мн.: Вышэйшая школа, 2002.
18. Экономическая теория: учебник / Н.И. Базылев, А.В. Бондарь, С.П. Гурко и др.; под общ. ред. Н.И. Базылева, С.П. Гурко. – Мн.: Экоперспектива, 1997.
19. Экономическая теория: учебник для студентов вузов / Под ред. В.Д. Камаева. – 6-е изд., перераб. и доп. – М.: ВЛАДОС, 2001.
20. Сажина М.А., Чибриков Г.Г. Основы экономической теории: учебное пособие для неэкономических специальностей вузов / Отв. ред. и руководитель авт. коллектива П.В. Савченко. – М.: Экономика, 1995.
21. Косова М.М. Описательная статистика учебных текстов по физике / М.М. Косо–ва, М.А. Зильберглейт // Труды БГТУ. Сер. VI. Издат. дело и полиграфия. – 2006. – Вып. XIV. – С. 167–170.
22. Терентьев П.В. Метод корреляционных плеяд / П.В. Терентьев // Вестник ЛГУ. – 1959. – № 9.
23. Терентьев П.В. Дальнейшее развитие метода корреляционных плеяд / П.В. Терентьев // Применение математических методов в биологии. – ЛГУ, Ленинград, 1960.
24. Айвазян С.А. Прикладная статистика и основы эконометрики: учебник для вузов / С.А. Айвазян, В.С. Мхитарян. – М.: ЮНИТИ, 1998. – 572 с.

РАЗРАБОТКА КОЛИЧЕСТВЕННОЙ МЕТОДИКИ ОЦЕНКИ ТРУДНОСТИ ВОСПРИЯТИЯ УЧЕБНЫХ ТЕКСТОВ ДЛЯ ВЫСШЕЙ ШКОЛЫ

Ю.Ф. Шпаковский (Белорусский государственный технологический университет)
Научный руководитель – к.филол.н., доцент Л.И. Петрова
(Белорусский государственный технологический университет)

В статье рассмотрены основные этапы разработки количественной методики для оценки трудности восприятия учебного текста для высшей школы (на материале текстов по химии). С этой целью была определена однозначная функциональная зависимость в виде статистической формулы, связывающей величины текстовых параметров и успешность понимания текста. Создание объективной методики позволит дать практические рекомендации авторам учебных текстов и редакторам таких текстов по их оптимизации в целях более успешного усвоения материала.

Введение

В настоящее время в сфере образования происходят существенные изменения, которые не могли не коснуться учебной литературы. За последние годы выросло количество учебных изданий для высшей школы, однако их качество, по мнению специалистов, не всегда соответствует требованиям, предъявляемых к изданиям подобного типа.

Одним из основных требований к учебным текстам является их простота и доступность изложения в них новой информации. В редакционно-издательской практике вопрос доступности материала автор и редактор решают в настоящее время, опираясь лишь на свою интуицию и профессиональный опыт. Отсутствие научной методики при оценке трудности восприятия учебного текста для высшей школы определило цель работы: разработать количественную методику оценки трудности восприятия учебного текста для высшей школы.

Разработка уравнения регрессии для оценки трудности восприятия учебного текста для высшей школы

Попытки объяснить, почему один текст для читателей сложнее (или легче), чем другой, и какие элементы внутри материала влияют на его усвоение, исследователи принимали уже давно. Например, в начале 30-х гг. прошлого столетия два американских ученых, У. Грей и Б. Лири, провели обширное исследование, опрашивая журналистов, редакторов, библиотекарей, педагогов с целью выяснить, чем обусловлено понимание или непонимание материала читателями [1]. Все факторы были сведены в четыре группы:

(1) содержание материала;

(2) язык и стиль произведения;

(3) организация материала (последовательность основных и второстепенных положений, правильное разбиение на параграфы и т.д.);

(4) внешнее оформление издания (формат издания, шрифт, иллюстрации, композиционные выделения и т.д.).

Исследования подобного рода получили бурное развитие в США в начале XX столетия и известны под словом «читабельность». Э. Дейл и Дж. Чолл под данным словом понимают «…некоторую характеристику печатного материала, зависящую от всех элементов внутри данного материала, которые влияют на успешность его усвоения определенной группой читателей. Мерой такого успешного усвоения является то, насколько средний читатель интересующей нас группы понимает исследуемый материал, в какой мере скорость, с которой он его читает, приближается к оптимальной, и, наконец, какой интерес представляет данный материал для этого среднего читателя» [2].

Следует отметить, что к проблеме исследования читабельности возможны два подхода – качественный и количественный [3].

При качественном подходе с помощью массовых опросов компетентных лиц пытаются выяснить, чем, по их мнению, обусловлена читабельность печатного материала. В вышеупомянутой работе У. Грея и Б. Лири используется именно этот подход. Данные, полученные в результате таких опросов, позволяют получить общее представление о факторах, влияющих на успешность усвоения печатного материала лицами с различной степенью подготовленности. Такой подход используется редко, так как полученные данные недостаточно надежны, а проведение таких опросов связано с большими организационными трудностями. Эти опросы помогают получить ряд гипотез о существовании факторов, от которых зависит трудность текста, но не могут помочь определить, насколько труден тот или иной материал для человека с определенным запасом знаний и навыков.

В связи с этим был разработан количественный метод для определения читабельности различных печатных изданий. При количественном подходе к изучению трудности восприятия текста сначала выбирается определенная группа лиц, а затем исследуется трудность понимания различного печатного материала для среднего читателя этой группы. Таким образом, утверждение, что текст *А* более труден, чем текст *Б*, значит, что текст *А* менее понятен случайной выборке из исследуемой группы лиц, чем текст *Б*, или текст *А* более труден, чем текст *Б*, если для адекватного понимания текста *А* и *Б* требуется, чтобы средний читатель имел более высокий образовательный уровень.

Выяснив путем опроса или тестирования читателей относительную трудность восприятия всех исследуемых текстов, экспериментатор стремится выявить все свойства, присущие текстам, которые дали бы возможность с достаточной степенью точности прогнозировать трудность их понимания. Очевидно, что совокупность факторов, предсказывающих степень понимания того или иного текста, даст объективный метод прогноза лишь в том случае, если можно будет надежно измерить эти факторы.

Таким образом, при количественном изучении читабельности встает проблема выяснения зависимости между оценкой трудности восприятия текста для определенной группы читателей, получаемой в результате тестирования, и объективными характеристиками этого текста. Для решения этой проблемы М. Вогель и К. Уошберн предложили использовать следующую схему [4].

1. На первом этапе экспериментатор производит репрезентативную выборку текстов из общей совокупности, трудность которых он собирается в дальнейшем определять при помощи данной формулы читабельности, и ранжирует выбранные тексты по их трудности для понимания средним представителем определенной группы лиц.

2. На втором этапе работы находится совокупность надежно измеряемых характеристик, общих для всех текстов, которые с максимальной степенью точности определяли бы степень их трудности, т.е. коэффициент множественной корреляции которых с оценками трудности текстов был бы максимальным.

3. На последнем этапе находится функция, которая бы однозначно описывала зависимость между успешностью понимания данного материала случайной выборкой интересующей группы лиц и определенной комбинацией упомянутых выше характеристик. Полученное уравнение регрессии, связывающее оценку трудности восприятия текста с рядом формальных элементов текста, называется формулой читабельности.

Практическое применение формул читабельности заключается в том, чтобы прогнозировать трудность восприятия материала для определенной группы читателей.

В нашем исследовании экспериментальным материалом послужили тексты из учебных изданий по химии для высших учебных заведений [5–9]. В качестве испытуемых выступали 150 студентов 3-го курса Белорусского государственного технологического университета (специальности химического профиля).

Исходя из предложенной схемы, **на первом этапе** были найдены объективные критерии, определяющие трудность восприятия текста отмеченной категорией читателей. С этой целью были проведены психолингвистические эксперименты по восприятию текстов с помощью двух методик: методики дополнения и экспертных оценок трудности восприятия текста [10, 11].

Методика дополнения (предложена в 1953 году Тэйлором) – это заполнение пропусков в тексте, в котором слова через определенный интервал заменены точками. Плюсы данной методики состоят в том, что пропускается всегда только одно слово, и слова пропускаются не по собственному усмотрению исследователя, а по определенным строгим правилам. Правильными считались не только слова, которые употребил автор, но и все словарные и контекстуальные синонимы.

Суть 2-го метода заключалась в следующем: после прочтения отрывка испытуемым предлагалось оценить его трудность по шестибалльной шкале: 1 – очень легкий текст; 2 – легкий текст; 3 – текст со средней трудностью; 4 – трудный текст; 5 – очень трудный текст; 6 – сверхтрудный текст.

Для того чтобы исключить поверхностное знакомство испытуемых с текстом, что исказило бы результаты при оценке трудности его восприятия, студентам перед суждением о трудности восприятия текста по шкале предлагалось выписать несколько ключевых слов и выразить основное содержание отрывка одним предложением. Эти меры заставили испытуемых тщательно изучить тексты. При тестировании фиксировалось также время работы с текстом.

По данным этих экспериментов были найдены объективные психолингвистические критерии, определяющие трудность восприятия текста. Среди них: процент неправильно заполненных пропусков и время работы с текстом (с использованием методики дополнения), средняя оценка трудности восприятия текста и время работы с ним (с использованием экспертных оценок трудности восприятия текста).

На втором этапе были выявлены текстовые параметры, величины которых позволили оценить сложность текста [11]. Всего было выделено 83 признака (например, средняя длина предложения, процент числа неповторяющихся слов, процент числа конкретных существительных). Анализ учебного текста по химии для высшей школы позволил сделать следующий вывод: трудность восприятия и понимания учебных текстов по химии, помимо общепризнанных параметров (длина слова, длина предложения, абстрактность существительных и др.), зависит также во многом от наличия и числа в таком тексте химических и математических формул, количества таблиц и иллюстраций, числа терминов, ключевых слов и типа распределения таких слов по тексту.

В связи с этим в нашем исследовании впервые выделено 22 параметра, связанных со спецификой учебного материала. К их числу относятся, например, количество таблиц и иллюстраций, процент числа всех терминов, процент числа всех ключевых слов, количество информации, процент числа условных обозначений в химических реакциях и математических формулах.

На третьем этапе были разработаны способы измерения величин текстовых параметров. Для этого одни признаки измерялись вручную (например, процент числа сложных и простых предложений, процент числа ключевых слов и терминов, процент числа существительных и т.д.), другие – с помощью компьютерных программ (например, средняя длина абзаца и предложения и др.), третьи – с помощью экспериментальных подсчетов, полученных с привлечением студентов (например, процент числа незнакомых слов, процент числа слов с различной трудностью межсловных связей).

Как известно, длина предложения является существенным показателем его сложности, что связано с кратковременной памятью человека. В связи с этим на данном этапе была определена допустимая трудность межсловных связей в предложении и максимальное количество слов, которое студент 3-го курса может запомнить при первом

чтении. Под трудностью связи слова следует понимать количество слов, которое следует запомнить, чтобы образовалась эта связь. Так как слова в предложении связаны, и это влияет на запоминание, необходимо было узнать, насколько грамматические и семантические связи увеличивают количество запоминаемых слов. В результате экспериментов мы выяснили, что допустимая трудность межсловных связей в предложении не должна превышать 4,61 несвязанных слова, каждая грамматическая связь позволяет запомнить на 0,81 слова больше, каждая семантическая связь – на 1,18 слова больше. Семантические связи между отдельными парами слов для разных людей различны, что зависит от подготовленности читателя, его уровня образования и т. д. В нашей работе сила семантической связи определялась экспериментально. В связи с этим в уравнении целесообразно учитывать только грамматические связи.

В результате было составлено следующее уравнение: $Y = 4{,}61 + 0{,}81X$, где Y – количество слов, которые необходимо запомнить для образования самой длинной связи этого слова; X – количество уже образовавшихся грамматических связей между словами.

На основе этой модели было вычислено максимальное количество слов, которое студент 3-го курса может запомнить при первом чтении, равное 16-ти словам. Для определения трудности связей слова уравнение имеет вид: $R = Y - 0{,}81X$, где R – трудность связей слова; Y – количество слов, необходимых запомнить для образования самой длинной связи этого слова; X – количество уже образовавшихся грамматических связей между теми же словами.

Таким образом, цель автора и редактора – ограничить трудность связи слова. Это позволит читателю осознавать связи с первого раза чтения и не возвращаться к уже прочитанному, теряя время.

Последняя важнейшая задача исследования – анализ взаимосвязи между степенью понимания учебного текста и его сложностью, который станет основанием для разработки уравнения регрессии для определения трудности восприятия учебного текста по химии для высшей школы.

Проблема установления характера взаимосвязей и выявления степени воздействия различных факторов на трудность восприятия учебного материала решалась с помощью шагового регрессионного анализа в программе «Статистика».

Регрессионный анализ проводился по каждой зависимой переменной, и все выделенные существенные факторы заносились в табл. 1.

Показатели трудности текстов	**Факторы**
Y_1 (84) — методика дополнения (ответы испытуемых)	53, 69, 27, 65, 46, 55, 30, 52, 6, 79, 9, 51, 64, 38, 36, 74, 13, 80, 32, 62, 26, 59, 60, 58, 76, 22
Y_2 (85) — методика дополнения (время работы с текстом)	53, 39, 42, 63, 52, 83, 54, 73, 9, 44, 65, 68, 29, 50, 19, 16, 74, 40, 64, 57, 61, 30, 46, 77, 70, 48
Y_3 (86) — шкалирование (оценка трудности текста по шкале)	53, 2, 69, 42, 65, 75, 55, 74, 26, 27, 28, 76, 43, 77, 8, 9, 50, 41, 51, 49, 30, 54, 23, 67, 70, 79
Y_4 (87) — шкалирование (время работы с текстом)	9, 38, 57, 52, 56, 41, 81, 72, 8, 27, 4, 77, 65, 53, 55, 61, 28, 69, 75, 1, 82, 32, 34, 51, 68

Таблица 1. Распределение факторов по показателям трудности текста

Для наглядности факторы можно распределить по частоте (табл. 2). Очевидно, что чем больше встречается определенный фактор трудности текста, тем более существенным является он в оценке его трудности.

Таким образом, регрессионный анализ позволил сделать вывод о наиболее существенных факторах трудности текста. Ими оказались следующие факторы: «средняя

длина предложения в словах» (9); «процент числа незнакомых слов» (53) и «процент числа сложносочиненных предложений» (65). На их основе был проведен шаговый регрессионный анализ.

Встречаемость факторов			
1 раз	**2 раза**	**3 раза**	**4 раза**
1, 2, 4, 6, 13, 16, 19, 22, 23, 29, 34, 36, 39, 40, 43, 44, 48, 49, 56, 58, 59, 60, 62, 63, 72, 73, 80, 81, 82, 83	8, 26, 28, 32, 38, 41, 42, 46, 50, 54, 57, 61, 64, 68, 70, 76, 79	27, 30, 51, 52, 55, 69, 74, 77	9, 53, 65

Таблица 2. Распределение факторов трудности текста по частоте

Значимым оказался только один фактор – «процент числа незнакомых слов» (53). Линейное регрессионное уравнение на основе данного фактора имеет следующий вид:

$$Y_1 = 38{,}65 + 0{,}652X_{53}. \quad (1)$$

Коэффициент множественной корреляции равен 0,65. Несмотря на то, что модель является достоверной (p-уровень значимости критерия Фишера меньше 0,05), уравнение с таким низким коэффициентом корреляции не является надежным. Кроме того, его применение будет затруднено по той причине, что в формуле используется фактор «процент числа незнакомых слов» (53). Данный признак текста определялся экспериментально, т.е. испытуемые подчеркивали незнакомые слова в тексте. Провести такой предварительный анализ не всегда возможно, поэтому следует разработать другое уравнение регрессии.

Далее было принято решение провести новый регрессионный анализ и включить в него все признаки текстов, которые связаны с фактором 53. Корреляционная матрица выявила связь со следующими признаками: длина текста в словах (2); средняя длина слов в буквах (23); средняя длина слов в печатных знаках (24); средняя длина слов по Деверу (25); процент числа слов длиной в 13 букв и больше (34); процент числа всех терминов (в единицах) (42); процент числа неповторяющихся терминов (в словах) (43); процент числа всех терминов (в словах) (44); процент числа абстрактных (чувственных и непредметных) существительных (48); процент числа прилагательных (50); процент числа пар слов с трудностью межсловных связей выше 4 (56); минимальная трудность межсловных связей (62); количество таблиц (75). Трудно понять, как с фактором «процент числа незнакомых слов» (53) связаны признаки 56 – процент числа пар слов с трудностью межсловных связей выше 4, 62 – минимальная трудность межсловных связей, 75 – количество таблиц. Но оставим их для регрессионного анализа. Кроме того, опираясь на здравый смысл, было добавлено еще несколько признаков, таких, как процент числа неповторяющихся слов (39), процент числа неповторяющихся словоформ (40), процент числа неповторяющихся терминов (в единицах) (41), средняя частота повторения слова (64), процент числа неповторяющихся ключевых слов (78), процент числа всех ключевых слов (79).

Пошаговый регрессионный анализ с данной группой признаков для каждого показателя трудности восприятия текста выявил следующие существенные факторы: «процент числа слов длиной в 13 букв и больше» (34), «процент числа неповторяющихся слов» (39), «процент числа всех терминов (в словах)» (44), «процент числа абстрактных (чувственных и непредметных) существительных» (48), «процент числа прилагательных» (50), «минимальная трудность межсловных связей» (62) и «процент числа всех ключевых слов» (79). Следовательно, на трудность восприятия учебного текста влияют длина слов, их абстрактность, количество разных слов, процент числа терминов и клю-

чевых слов, а также минимальная трудность связи. Влияние большинства из названных факторов на трудность восприятия текста было доказано в предыдущих исследованиях по читабельности научно-популярных текстов, материалов из средств массовой информации. Но появились относительно новые факторы, что связано с особенностями учебного текста.

Как показывает корреляционно-регрессионный анализ, для учебного материала, помимо известных факторов трудности текста (длина предложения, количество разных слов, их абстрактность), немаловажную роль играет число терминов и ключевых слов. Этот вывод должны принять во внимание не только составители вузовских учебников, но и лица, ответственные за их выпуск (в частности, редакторы). Удовлетворительное уравнение регрессии, учитывающее фактор «средняя длина предложения в словах» (9) и один из вышеперечисленных факторов, получить не удалось. Наиболее оптимальное полученное уравнение имеет вид:

$$Y_2 = 59{,}54 + 0{,}349X_{39} + 0{,}554X_{44} - 0{,}55X_{62}. \quad (2)$$

Коэффициент множественной корреляции равен 0,83. Однако применение и этой формулы затруднено из-за фактора «минимальная трудность межсловных связей» (62). Хотя сегодня статистической обработкой текста занимается компьютер, но даже он не в силах подсчитать трудность межсловных связей по Ингве. Поэтому лучше вычислить формулу без учета этого фактора:

$$Y_2 = 3{,}27 + 0{,}713X_{39} + 0{,}528X_{44}. \quad (3)$$

Коэффициент множественной корреляции равен 0,72. Частная корреляция факторов «процент числа неповторяющихся слов» (39) и «процент числа всех терминов (в словах)» (44) показала, что оба фактора являются самостоятельными. Очевидно, что качество анализа текста в данном случае будет зависеть от качества подключаемых словарей к программе анализа читабельности текста. Поэтому, несмотря на хорошие показатели полученной формулы, попробуем найти ей альтернативу.

Для этой цели в шаговый регрессионный анализ были включены признаки, которые встречались 2 раза. И этот анализ не дал положительных результатов, выявив дополнительно такие наиболее существенные факторы, как «процент числа слов длиной в 6 букв и больше» (27), «процент числа глаголов» (51) и «процент числа придаточных предложений среди общего числа фраз» (69).

Далее было решено провести анализ признаков, которые встречались 3 раза. Были выделены значимые факторы «процент числа слов в 6 слогов и больше» (38); «процент числа неповторяющихся терминов (в единицах)» (41); «процент числа простых предложений» (68) и получено следующее уравнение регрессии:

$$Y_4 = 0{,}034 + 0{,}392X_{38} - 0{,}43X_9. \quad (4)$$

Коэффициент множественной корреляции равен 0,61. Несмотря на невысокий коэффициент корреляции, эта формула достаточно проста в применении. Подсчет факторов «средняя длина предложения в словах» (9) и «процент числа слов в 6 слогов и больше» (38) не представляет особой трудности как при ручной, так и при машинной обработке текста. Однако низкое значение коэффициента корреляции заставляет продолжить поиски более надежной формулы.

С этой целью в шаговый регрессионный анализ были включены признаки текстов из второй и третьей групп. По результатам анализа было получено две формулы. В первую вошли факторы «процент числа слов длиной в 9 букв и больше» (30), «процент числа всех терминов (в единицах)» (42) и «процент числа условных обозначений в химических реакциях» (76).

$$Y_2 = 20{,}24 + 0{,}48X_{30} + 0{,}58X_{42} + 0{,}41X_{76}. \quad (5)$$

Коэффициент множественной корреляции равен 0,78. Несмотря на то, что в данную формулу не вошел фактор «средняя длина предложения в словах» (наиболее рас-

пространенный фактор трудности текста), следует полагать, что это наиболее удачная из всех разработанных формул.

Необходимо также отметить, что в формуле (3) использовался однородный фактор «процент числа всех терминов (в словах)» (44), сумма интеркорреляций которого наибольшая – 13,24. Но использование его в данной формуле понижает коэффициент множественной корреляции до 0,68. Для анализа текста без химических реакций можно воспользоваться следующей формулой (коэффициент корреляции равен 0,67):

$$Y_2 = 25{,}57 + 0{,}6X_{30} + 0{,}97X_{42}. \quad (6)$$

Прежде чем использовать разработанные формулы, следует определить их надежность и валидность.

Надежность уравнения зависит от следующих показателей:

а) общей длины анализируемых отрывков;

б) точности измерения факторов трудности текста, входящих в уравнение регрессии.

В 60–70-х гг. исследователи уделяли достаточно внимания рассмотрению данных вопросов. Сегодня информационные технологии настолько усовершенствованы, что решение этих двух вопросов сводится к созданию программного обеспечения для оценки трудности восприятия текста определенной категории читателей.

Под *валидностью* инструмента измерений обычно понимается степень, в которой удается измерить именно ту характеристику, которую намеревались измерять при помощи данного инструмента. В нашем случае с помощью разработанных уравнений делается попытка измерить трудность восприятия текста. Дж. Клэар выделяет три характеристики валидности [12]:

а) точность моделирования читабельности текстов, взятых в основу разработки формулы;

б) перекрестная валидность, которая характеризует степень совпадения показателей читабельности, вычисленных по различным формулам;

в) валидность, определяемая по внешнему критерию.

Точность моделирования читабельности текстов можно оценить с помощью следующих основных показателей: множественный коэффициент корреляции (R), множественный коэффициент детерминации (R^2), стандартная ошибка оценки (standard error of estimate), F-критерий Фишера и уровень значимости критерия Фишера (F, p). Значения показателей для разработанных формул приведены в табл. 3.

Показатели	**Номер формулы**					
	1	**2**	**3**	**4**	**5**	**6**
R	0,65	0,83	0,72	0,61	0,78	0,67
R^2	0,42	0,69	0,48	0,38	0,61	0,46
Standard error	4,5	3,63	4,42	0,004	4,08	4,71
F	19,2	18,09	13,8	7,78	12,6	10,7
p	0,0016	0,000002	0,00008	0,0023	0,00003	0,0004

Таблица 3. Значения основных показателей для созданных уравнений

Выполненный анализ показал, что все разработанные формулы являются адекватными моделями, но для практического применения следует использовать формулу, удобную для компьютерной реализации и наиболее надежную по вышеприведенным показателям. Формула (4) достаточно проста для применения, факторы «средняя длина предложения в словах» (9) и «процент числа слов в 6 слогов и больше» (38) легко рассчитать. Но формула (4), наряду с формулами (1), (3) и (6), имеет невысокий множественный коэффициент корреляции, что снижает ее надежность. Самый высокий коэффициент корреляции у формулы (2), но ее использование ограничено трудно определяемым фактором «минимальная трудность связи» (62). Таким образом, наилучшей яв-

ляется формула (5) с коэффициентом корреляции 0,78, *F*-критерием Фишера 12,6 и *p*-уровнем значимости 0,00003. Применение уравнения предполагает наличие словаря химических терминов, что при наличии информационных технологий является выполнимой задачей.

Заключение

Исходя из анализа, можно сделать вывод, что для оценки трудности восприятия текста по химии можно использовать формулу (5), в которую вошли следующие факторы: «процент числа слов длиной в 9 букв и больше» (30), «процент числа всех терминов (в единицах)» (42) и «процент числа условных обозначений в химических реакциях» (76).

Таким образом, в работе впервые разработана количественная методика оценки трудности восприятия учебного материала для высшей школы, что позволило дать практические рекомендации авторам учебных текстов и редакторам таких текстов по их оптимизации в целях их более успешного усвоения.

Результаты работы внедрены в практику ряда отечественных издательств Республики Беларусь («Вышэйшая школа», «Харвест»).

Литература

1. Gray, W.S. What makes a book readable / W.S. Gray, B.Leary. – Chicago: Chicago University Press, 1935.
2. Dale, E. The concept of readability / E. Dale, J.S. Chall // Elementary English. – 1949. – № 26. – P. 23.
3. Шпаковский Ю.Ф. Трудность текста и ее измерение / Ю.Ф. Шпаковский, Л.И. Петрова // Труды Белорусск. гос. технолог. ун-та. Сер. IX, Издательское дело и полиграфия. – 2003. – Вып. XI. – С. 8–11.
4. Vogel, M. An objective method of determining grade placement of children's reading material / M. Vogel, C. Washburne // Elementary school journal. – 1928. – № 28 – P. 373–381.
5. Карапетьянц М.Х. Общая и неорганическая химия: учебник / М.Х. Карапетьянц, С.И. Дракин. – 2-е изд., перераб. и доп. – М.: Химия, 1992. – 592 с.: ил.
6. Пилипенко, А.Т. Аналитическая химия: кн. 1 / А.Т. Пилипенко, И.В. Пятницкий. – М.: Химия, 1990. – 480 с.: ил.
7. Стромберг А.Г. Физическая химия: учеб. для хим.-технол. спец. вузов / А.Г. Стромберг, Д.П. Семченко; под ред. А.Г. Стромберга. – 2-е изд., перераб. и доп. – М.: Высш. шк., 1988. – 496 с.: ил.
8. Фролов Ю.Г. Курс коллоидной химии (поверхностные явления и дисперсные системы): учебник для вузов / Ю.Г. Фролов. – М.: Химия, 1982. – 400 с.: ил.
9. Щербина А.Э. Органическая химия. Реакционная способность основных классов органических соединений: учеб. пособие для студ. хим.-технол. спец. / А.Э. Щербина [и др.]. – Мн.: БГТУ, 2000. – 624 с.: ил.
10. Шпаковский Ю.Ф. Экспериментальное определение трудности текстов по химии / Ю.Ф. Шпаковский, Н.И. Шишкина // Славянские языки: системно-описательный и социокультурный аспекты исследования: материалы респуб. научно-метод. конф. (12–13 ноября 2003 г.). – Брест: Изд-во УО «БрГУ им. А. С. Пушкина», 2003. – С. 298–301.
11. Шпаковский Ю.Ф. Формулы читабельности как метод оценки качества книги / Ю.Ф. Шпаковский // Кваліlогія книги: збірник наукових праць. – Львів, 2003. – С. 39–48.
12. Klare, G.R. The measurement of readability / G.R. Klare. — Ames, Iowa: Iowa State University Press, 1963.

ИСПОЛЬЗОВАНИЕ МНОГОАГЕНТНЫХ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ СОСТАВЛЕНИЯ РАСПИСАНИЙ

И.В. Крысина (Нижегородский филиал государственного университета – Высшей школы экономики)

Научный руководитель – к.т.н., доцент Э.А. Бабкин (Нижегородский филиал государственного университета – Высшей школы экономики)

В рамках работы была предпринята попытка разработки агентного алгоритма составления расписания для вуза. Был разработана программная реализация алгоритма, проведен анализ эффективности работы алгоритма, его сравнение с результатами работы человека.

Введение

Невозможно переоценить роль расписания времени в жизни современного общества. На основе расписаний осуществляется планирование деятельности как каждого отдельного человека, так и огромных промышленных комплексов. Это доказывает, что одним из самых важных навыков современного человека становится умение правильно распределять свое рабочее время. Идеальное расписание обязательно должно обладать следующим свойством: все пользователи получают максимально возможную выгоду от его использования. В связи с этим возникает необходимость учета требований, предъявляемых пользователями к расписанию, и нахождения оптимального расписания при заданных ограничениях. Т.е. решается задача максимизации полезности каждого индивида. Задача усложняется, если пользователи удалены друг от друга: возникает проблема обмена информацией и фиксирования достигнутых соглашений

В связи с этим встает вопрос: нельзя ли заменить пользователей расписания программными агентами, способными обмениваться сообщениями друг с другом с целью поиска наилучшего решения. Тогда рутинный процесс совместного составления расписания можно будет заменить автоматизированным процессом взаимодействия агентов-представителей в рамках некоторого программного комплекса.

Подобные программные комплексы представляются очень перспективными в свете роста актуальности проблемы составления расписаний. В частности, к ним можно отнести задачи составления расписаний (движение поездов и самолетов, занятия учебных групп в вузе, выполнение вычислительных задач на ВЦ коллективного пользования или в многопроцессорной компьютерной системе, операции на станке с программным управлением, задания в годовом плане предприятия, передача пакетов данных в сети Интернет).

Целью данного исследования является разработка агентного алгоритма составления расписания учебного заведения.

Научная новизна исследования заключается в применении агентного подхода к решению задачи составления учебного расписания для вуза. Ранее при решении данной проблемы применялись алгоритмы линейного программирования [3], методы поиска с использованием метаэвристик [5], имитация остывания [2, 6], кластерные методы [7].

При составлении расписания учитываются предпочтения пользователей о времени и месте проведения занятий. Это свойство предложенного алгоритма позволяет повысить качество получаемого расписания. Принципиальное отличие предложенного алгоритма от ранее разработанных заключается в том, что он позволяет распределять ресурсы аудиторного фонда между пользователями расписания и, в принципе, любые другие виды ресурсов. Другие агентные алгоритмы составляют только расписание времени [4]. Большинство современных систем составления расписания учебных заведений не позволяют пользователям напрямую влиять на процесс составления расписания [1]. Практическая значимость данного исследования состоит в том, что предложен спо-

соб обработки предпочтений пользователей, что, в свою очередь, повышает качество получаемых расписаний.

Важной характеристикой любой системы составления расписаний является способность быстро реагировать на изменения входных параметров. Агентный алгоритм в случае изменения предпочтений пользователей расписания изменит лишь ту часть расписания, которая имеет отношение к данному пользователю. Таким образом, исключается повторное решение задачи.

В следующих разделах статьи приведены математическая постановка задачи, описание многоагентного алгоритма и описание архитектуры разработанного приложения.

Математическая постановка задачи [8]

В целях упрощения изложения лиц, заинтересованных в результатах процесса составления расписания, будем называть пользователями расписания. К пользователям расписания относятся преподаватели и учебные группы. Роль первых – провести лекционное или практическое занятие, роль последних – присутствовать на занятиях. Введем следующие обозначения.

Учебные группы и потоки. $g \in \mathbf{G}$ – номер учебной группы. $\mathbf{G}$ – множество номеров всех учебных групп. $|\mathbf{G}| = \gamma$ – количество групп.

Каждая из групп принадлежит одному или нескольким потокам. При объединении нескольких групп в один поток используются следующие принципы.

1) Группы в потоке используют один и тот же аудиторный фонд. Например, поток «Гуманитарные предметы, ИСиТ, 4-й курс», состоящий из групп 04-ПМ, 04-БИ, использует аудиторный фонд из двух аудиторий на 120 мест.

2) Лекции читаются всему потоку сразу.

3) У каждого потока есть хотя бы одно занятие.

4) Поток может состоять из одной учебной группы. Например, поток «Прикладная математика, предметы по специальности, 4-ый курс» состоит из одной группы 04-ПМ.

$\mathbf{R}$ – множество номеров учебных потоков. $|\mathbf{R}| = \rho$ – количество потоков. $r \in \mathbf{R}$ – номер потока. Каждая группа является самостоятельным потоком, поэтому $\rho \geq \gamma$.

$\mathbf{C_r} \subset \mathbf{G}$ – поток. Потоком называется любое подмножество множества групп. $\mathbf{C} = \{\mathbf{C_1}, \mathbf{C_2}, ..., \mathbf{C_\rho}\}$ – множество всех потоков. Потоки могу пересекаться между собой.

Преподаватели. $p \in \mathbf{P}$ – номер преподавателя, P – множество всех преподавателей. $|\mathbf{P}| = \pi$ – количество преподавателей.

Пользователи расписания. Объединение множества всех групп с множеством всех преподавателей $\mathbf{M} = \mathbf{G} \cup \mathbf{P}$, $m \in \mathbf{M}$ – уникальный номер пользователя расписания.

Время. $w \in \mathbf{W_g}$ – номер учебного дня недели. $\mathbf{W_g} \subset \mathbf{W} = \{1,2,...,7\}$ – множество учебных дней группы g, W – множество всех дней недели. $j \in \mathbf{J} = \{1,2,...,8\}$ – номер учебной пары. $\mathbf{T} = \{(w, j) \mid w \in \mathbf{W}, j \in \mathbf{J}\}$ – множество таймслотов (элементарных единиц времени в задаче составления расписания). Например, таймслот (1,2) означает понедельник, вторая пара. Для каждого пользователя m известно множество таймслотов, когда он свободен, т.е. в это время занятие может быть проведено $\mathbf{T}_m^+ \subset \mathbf{T}$, и множество недоступных таймслотов $\mathbf{T}_m^- \subset \mathbf{T}$ ($\mathbf{T}_m^+ \cup \mathbf{T}_m^- = \mathbf{T}; \mathbf{T}_m^+ \cap \mathbf{T}_m^- = \varnothing$), например, выходные дни, когда занятия не могут быть проведены.

Занятия. Преподаватели проводят лекционные и практические занятия. Лекционные занятия проводятся у всего потока сразу, практические занятия – только у одной

группы. Кроме того, практические занятия требуют другого аудиторного фонда. Например, практические занятия по информатике должны проходить в компьютерном классе. Введем следующие обозначения для занятий.

$\mathbf{S_r} = \{1, 2, ..., \sigma_r\}$ – множество номеров лекционных занятий, читаемых на потоке r. $s_r \in \mathbf{S_r}$ – номер лекционного занятия;

$\mathbf{Q_r} = \{1, 2, ..., \theta_r\}$ – множество номеров практических занятий, проводимых на потоке r. $q_r \in \mathbf{Q_r}$ – номер практического занятия.

Можно видеть, что глобально уникальным идентификатором занятия будет пара «номер потока – номер занятия на потоке». Действительно, занятия существенно зависят от потока. Так, «физика на 2-м курсе» отличается по содержанию от «физика на 3-м курсе».

Лекционное занятие однозначно идентифицируется парой значений «номер потока – номер лекционного занятия»: $(r, s_r) \in \mathbf{RS}$, где

$$\mathbf{RS} = \{(r, s_r) \mid r \in \mathbf{R}, s_r \in \mathbf{S_r}\} \tag{0.1}$$

есть множество всех лекционных занятий.

Количество лекционных занятий

$$|\mathbf{RS}| = \sum_{r=1}^{\rho} \sigma_r \,. \tag{0.2}$$

Практическое занятие однозначно идентифицируется тройкой значений «номер потока – номер практического занятия – номер группы»: $(r, g_r, q_r) \in \mathbf{RQG}$, где

$$\mathbf{RQG} = \{(r, g_r, q_r) \mid r \in \mathbf{R}, q_r \in \mathbf{Q_r}, g_r \in \mathbf{C_r}\} \tag{0.3}$$

есть множество всех практических занятий.

Количество практических занятий

$$|\mathbf{RQG}| = \sum_{r=1}^{\rho} |\mathbf{C_r}| \cdot \theta_r \,,$$

где $|\mathbf{C_r}|$ – количество групп в потоке $\mathbf{C_r}$.

При дальнейшем описании задачи нам в большинстве случаев будет неважно, какое занятие рассматривается – лекционное или практическое. Поэтому под занятием будем понимать элемент из объединения двух множеств – множества лекционных и множества практических занятий:

$$\mathbf{E} = \mathbf{RS} \cup \mathbf{RQG} \,. \tag{0.4}$$

Учебный план закрепляет за каждым преподавателем предметы, которые он должен будет провести в течение семестра. Назначение лекционных занятий:

$\delta_1 : \mathbf{RS} \to \mathbf{P}$,

где $\mathbf{P}$ – множество преподавателей; $\mathbf{RS}$ – множество лекционных занятий.

Учебный план практических занятий:

$\delta_2 : \mathbf{RQG} \to \mathbf{P}$,

где $\mathbf{RQG}$ – множество троек «номер потока – номер практического занятия – номер группы». Например, $\mathbf{C_1} = \{3, 4, 5\}$ – поток 1 состоит из групп 3, 4, 5; $\mathbf{Q_1} = \{1, 2\}$ – у потока 1 всего два практических занятия; запись $\delta_2(1, 2, 4) = 7$ означает, что у группы с номером 4, которая входит в поток 1, практическое занятие номер 2 ведет преподаватель 7.

В общем случае учебный план можно записать так:

$$\delta : \mathbf{E} \to \mathbf{P} \,, \tag{0.5}$$

где $\delta(e)=\begin{cases}\delta_1(e), e\in \mathbf{RS}\\ \delta_2(e), e\in \mathbf{RQG}\end{cases}$.

Зная учебный план, можно вычислить множество занятий, проводимых пользователем-преподавателем (или для пользователя-группы) с номером *m*:

$$\mathbf{E}_m=\{e\mid m\in\mathbf{P}\wedge\delta(e)=m\}\cup$$
$$\{e=(r,s)\mid m\in\mathbf{C}_r\wedge s\in\mathbf{S}_r\}\cup \quad . \qquad (0.6)$$
$$\{e=(r,q,m)\mid m\in\mathbf{C}_r\wedge q\in\mathbf{Q}_r\}$$

Аудиторный фонд. Аудиторный фонд представляет собой множество аудиторий, где могут быть проведены занятия.

$\mathbf{A}$ – множество всех аудиторий и помещений.

Для каждого занятия e выделяется некоторое подмножество аудиторий $\mathbf{A}_e\subset\mathbf{A}$; аудитория для проведения занятия выбирается только из этого подмножества.

Неизвестные функции задачи. Неизвестными в данной задаче являются, расписание времени и расписание аудиторий. Расписание времени – это отображение множества предметов на множество таймслотов:

$$\tau:\mathbf{E}\to\mathbf{T}, \qquad (0.7)$$

где $\mathbf{E}$ – множество всех занятий; $\mathbf{T}$ – множество всех таймслотов.

Расписание аудиторий – это отображение множества занятий на множество аудиторий:

$$\alpha:\mathbf{E}\to\mathbf{A}, \qquad (0.8)$$

где $\mathbf{E}$ – множество всех занятий; $\mathbf{A}$ – множество всех аудиторий.

Пример 1. $\alpha(1,2)=101$ означает, что занятие 2 потока 1 проводится в аудитории 101.

Пример 2. $\mathbf{C_1}=\{3,4,5\}$ – поток 1 состоит из групп 3, 4, 5; $\mathbf{Q_1}=\{1,2\}$ – у потока 1 всего два практических занятия; запись $\alpha(1,2,4)=101$ означает, что у группы с номером 4, которая входит в поток 1, практическое занятие номер 2 будет проводиться в аудитории 101.

Ограничения. Введем в задаче следующие ограничения.

1. Один преподаватель в каждый момент времени может проводить не более одного занятия.

$$\forall p\in\mathbf{P},\forall e_1,e_2\in\mathbf{E}: e_1\neq e_2\wedge\delta(e_1)=\delta(e_2)=p\Rightarrow\tau(e_1)\neq\tau(e_2) \qquad (0.9)$$

2. В одной аудитории в каждый момент времени может проводиться не более одного занятия.

$$\forall a\in\mathbf{A},\forall e_1,e_2\in\mathbf{E}: e_1\neq e_2\wedge\alpha(e_1)=\alpha(e_2)=a\Rightarrow\tau(e_1)\neq\tau(e_2) \qquad (0.10)$$

3. У одной группы в каждый момент времени может проводиться не более одного занятия.

$$\forall g\in\mathbf{G}:\left(e_1=(r_1,...),e_2=(r_2,...)\in\mathbf{E}\wedge g\in\mathbf{Cr_1}\wedge g\in\mathbf{Cr_2}\wedge e_1\neq e_2\right)\vee$$
$$\vee\left(e_1=(...,g),e_2=(...,g)\in\mathbf{E}\wedge e_1\neq e_2\right)\Rightarrow\tau(e_1)\neq\tau(e_2) \qquad (0.11)$$

Другими словами, для каждых двух пересекающихся потоков в каждый момент времени не может быть разных лекционных занятий в одно и то же время, и для каждой группы не может быть разных практических занятий в одно и то же время. Это ограничение удобно проверять, предварительно разбив все занятия на γ классов. В класс $\mathbf{K_g}\subset\mathbf{E}, g\in\mathbf{G}\left(g_1\neq g_2\Rightarrow\mathbf{K_{g_1}}\cap\mathbf{K_{g_2}}=\varnothing\right)$ входят все занятия, которые читаются группе

g персонально или в каком-либо потоке. Ограничение 3 будет выполнено тогда и только тогда, когда все занятия групп одного класса проводятся в разное время.

Приоритеты занятий. Очевидно, что не все занятия имеют одинаковую важность в рамках учебного процесса. Для них можно задать порядок по приоритету. Более приоритетные занятия будут в первую очередь занимать время и место, менее приоритетные занятия уже не смогут претендовать на эти ресурсы. Итак, приоритеты занятий представляют собой отношение частичного порядка на множестве занятий

$$e_1 \succ e_2 \Leftrightarrow U(e_1) \geq U(e_2), \qquad (0.12)$$

где $e_1, e_2 \in \mathbf{E}$; $U(e) = |\mathbf{M}_e| + k_1(e) + k_2(e) + k3(p_e)$ – «полезность» занятия e.

$\mathbf{M}_e = \{m \mid (e = (r,s) \in \mathbf{RS} \wedge m \in \mathbf{C}_r) \vee (e = (r,q,m) \in \mathbf{RQG})\}$ – множество групп, у которых проводится занятие e;

$k_1(e) \in \{0,5,10\}$ характеризует степень важности занятия для специальности (0 – «ненужные занятия», 5 – важное, но не по специальности, 10 – занятие по специальности);

$k_2(e) \in \{0,2\}$ – 0 – младшие курсы, 2 – старшие курсы;

$p_e = p, \delta(e) = p$ – номер преподавателя, который ведет занятие e;

$k_3(p_e) \in \{0..5\}$ – оценка преподавателя как научного сотрудника; более высокий коэффициент означает, что преподаватель – более ценный научный сотрудник. В случае равенства $U(e_1) = U(e_2)$ занятия упорядочиваются в лексикографическом порядке.

Предпочтения пользователей. Важной частью предлагаемой модели являются предпочтения пользователей. Каждое предпочтение представляет собой числовую оценку в диапазоне от 0 до 1. 0 соответствует наименьшему уровню предпочтения, 1 – наибольшему. В модели имеют место два вида предпочтений:

1. предпочтения пользователя $m \in \mathbf{M}$ о времени проведения занятия:

$$f_m^1 : \mathbf{E}_m \times \mathbf{T}_m^+ \to [0,1] \qquad (0.13)$$

2. предпочтения пользователя $m \in \mathbf{M}$ о месте проведения занятия:

$$f_m^2 : \mathbf{EA}_m \to [0,1], \qquad (0.14)$$

где $\mathbf{EA}_m = \{(e,a) \mid a \in \mathbf{A}_e \wedge e \in \mathbf{E}_m\}$ – множество допустимых пар «занятие – аудитория».

Графически предпочтения можно представить в виде табл. 1 и 2. Более темный цвет отмечает более предпочтительное время (место) для проведения занятия.

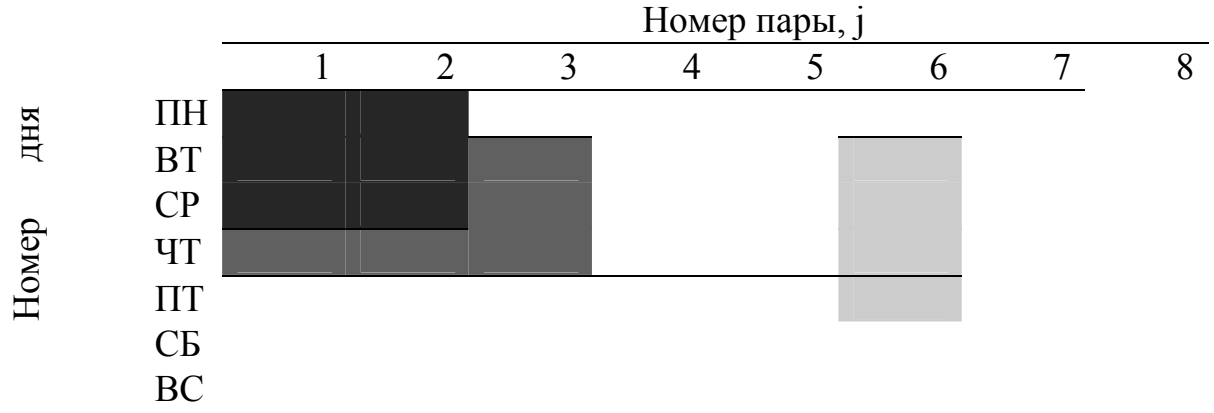


Таблица 1. Предпочтения пользователя m о времени проведения занятия, f_m^1

Аудитория, а	1	2	3	4	5
Оценка, f_m^2	[redacted]		[redacted]		

Таблица 2. Предпочтения пользователя m о месте проведения занятия, f_m^2

Критерий качества решения. Критерий качества представляет собой обобщенный критерий, состоящий из нескольких частных критериев. Критерий оценивает найденное решение, т.е. пару функций $\tau(e), \alpha(e)$.

Частный критерий

$$F_e^1(\tau) = \sum_{m \in \mathbf{M}_e} f_m^1\left(e, \tau(e)\right) \to \max \qquad (0.15)$$

определяет сумму предпочтений пользователей о времени занятия e. Частный критерий

$$F_e^2(\alpha) = \sum_{m \in \mathbf{M}_e} f_m^2\left(e, \alpha(e)\right) \to \max \qquad (0.16)$$

определяет сумму предпочтений пользователей о месте проведения занятия e. Обобщенный критерий

$$F(\tau, \alpha) = \sum_{e \in \mathbf{E}} \left(F_e^1 + F_e^2\right) \to \max \qquad (0.17)$$

вначале максимизирует пару частных критериев для наиболее приоритетного занятия, затем для следующего по приоритету и т.д. При этом в паре критериев вначале максимизируется первый критерий (время), затем второй (место). Такая структура обобщенного критерия гарантирует, что более приоритетные занятия получат лучшее время и место. Считается, что время проведения занятия более важно при составлении расписания, поэтому вначале определяется время, а затем место проведения занятия.

Решение. Решением задачи являются неизвестные функции τ, α при условии выполнения всех ограничений и максимальном значении критерия качества.

Описание алгоритма

В разделе дается описание алгоритма составления расписания учебного заведения. Помимо расписания времени, составляется еще и расписание аудиторий.

Агенты разделяются на агентов преподавателей, агентов учебных групп и агентов аудиторий. Роли организаторов выполняют агенты преподавателей, роли участников выполняют агенты учебных групп и аудиторий. Количество агентов каждого типа определяется количеством соответствующих пользователей. Агент аудиторий – один для каждого расписания.

Поиск времени и места для проведения занятия осуществляется посредством деятельности агента преподавателя. Данный агент способен отправлять и получать различные типы сообщений, переходя при этом из одного состояния в другое. Ниже приводится описание действий, выполняемых в каждом состоянии.

1) Ask_when_avail. Первое состояние, в котором оказывается агент преподавателя. В этом состоянии он посылает агентам групп запрос WHENAVAIL с номером занятия, чтобы они сообщили свое свободное время. Агенты групп отвечают сообщением USERAVAIL, в котором сообщается, когда агент свободен. Если все агенты сообщили свой ответ, тогда агент преподавателя находит пересечение по времени. Если пересечение пусто, тогда агент переходит в состояние (1a) imposs_meeting.

1a) Imposs_meeting. Состояние, в которое агент попадает, если свободное время агентов групп данного занятия имеет пустое пересечение, т.е. решения найти не удалось. Составить расписание для занятия не удалось. Занятие помечается «не имеющее решения».

2) Ask_subj_prefs. В этом состоянии агент преподавателя опрашивает агентов групп, чтобы они сообщили свои предпочтения о времени и месте проведения занятия (сообщение EVALUATE). Агенты сообщают свои предпочтения в сообщении SUBJPREFS. В этом сообщении содержатся предпочтения о времени и предпочтения о месте проведения занятия. Полученные предпочтения времени сортируются в соответствии с критерием алгоритма MSRAC, то есть по возрастанию приоритета. Полученные предпочтения о месте тоже сортируются по убыванию предпочтений, т.е. первая аудитория будет наиболее предпочтительной.

3) Propose_time. Агент выбирает первый таймслот из полученного на предыдущем шаге упорядочения времени. Агент устанавливает это время в сообщении и посылает запрос агентам учебных групп (TIMEPROPOSAL), содержащий номер занятия и предлагаемый таймслот. Те проверяют свой календарь. Если в предложенное время никаких занятий нет, тогда агент группы дает положительный ответ – ACCEPT. Если же в календаре уже есть другое занятие на это время, тогда агент группы сравнивает приоритеты нового и старого занятий. Если приоритет нового занятия больше, то агент группы соглашается (ACCEPT), иначе отказывается (REJECT). В случае равных приоритетов используется критерий метрополиса. В случае согласия агент преподавателя переходит в следующее состояние, в случае отказа агент возвращается в состояние propose_time и выбирает следующий таймслот. Если таймслотов больше нет, значит, решения для данного занятия не существует. Агент переходит в состояние (3a) solnot_found.

3a) Solnot_found. Делается пометка, что для занятия расписания составить не удалось.

4) Propose_location. Агент посылает отсортированный список аудиторий, полученный в состоянии (2) агенту аудиторий. LOCPROPOSAL содержит номер занятия, номер таймслота и список возможных аудиторий, отсортированных по предпочтению. Агент аудиторий получает LOCPROPOSAL и, используя свой календарь «аудитория – время – занятие», находит первую аудиторию из списка, которая в указанно время либо не занята, либо занята менее приоритетными занятиями. Если аудитория найдена, то агент аудиторий посылает ответ ACCEPT с номером аудитории, иначе агент посылает ответ REJECT. При получении положительного ответа агент преподавателя переходит в следующее состояние. Если же был получен отрицательный ответ, то агент преподавателя возвращается в состояние propose_time для определения нового времени занятия.

5) Fix_meeting. Если агент преподавателя приходит в это состояние, значит, время и место были успешно найдены. Агент преподавателя рассылает всем агентам групп и агенту аудиторий сообщение FIXMEETING с номером занятия, номером таймслота и номером аудитории. Если у агентов групп в календаре в указанное время пусто и у агента аудиторий указанное место пусто, то время и место занятия фиксируются. Иначе, если например, время или место уже заняты более приоритетным занятием, то менее приоритетное занятие отменяется с помощью сообщения CANCEL MEETING, отправляемого агенту преподавателя отменяемого занятия. Преподаватель, получив это сообщение, затем пересылает его всем участникам. Участники убирают указанное занятие из своего календаря.

В состояниях (1), (2), (3) и (4), если не все агенты прислали ответ в течение некоторого времени, то агент преподавателя помещает занятие в список отмененных занятий, потом он попробует заново найти для него расписание.

После того, как все агенты преподавателей завершат поиск расписания своих занятий, общее расписание считается составленным. Важным является тот факт, что частичное расписание всегда будет составлено. Полное расписание всех занятий иногда просто не существует, в таком случае рассмотренный алгоритм все равно найдет расписание, просто менее приоритетные занятия будут автоматически отброшены.

Описание архитектуры приложения

При разработке приложения использовались следующие программные продукты и технологии.

- Java – в качестве основного языка программирования. Выбор был обусловлен необходимостью обеспечения возможности переноса приложения с одной платформы на другую, гибкостью и простотой разработки графического интерфейса, возможностью запуска приложения на мобильных платформах (сотовые телефоны, коммуникаторы, КПК).
- MySQL – СУБД. Критерии выбора: свободное распространение, высокая скорость обработки запросов, поддержка транзакций, поддержка JDBC.
- JADE (Java Agent DEvelopment Framework) – многоагентная система, разрабатываемая Итальянским институтом телекоммуникаций при поддержке таких компаний, как Motorola, France Telecom R&D. Критерии выбора – свободное распространение, открытость, полное соответствие стандартам (FIPA), возможность исполнения на мобильных устройствах.

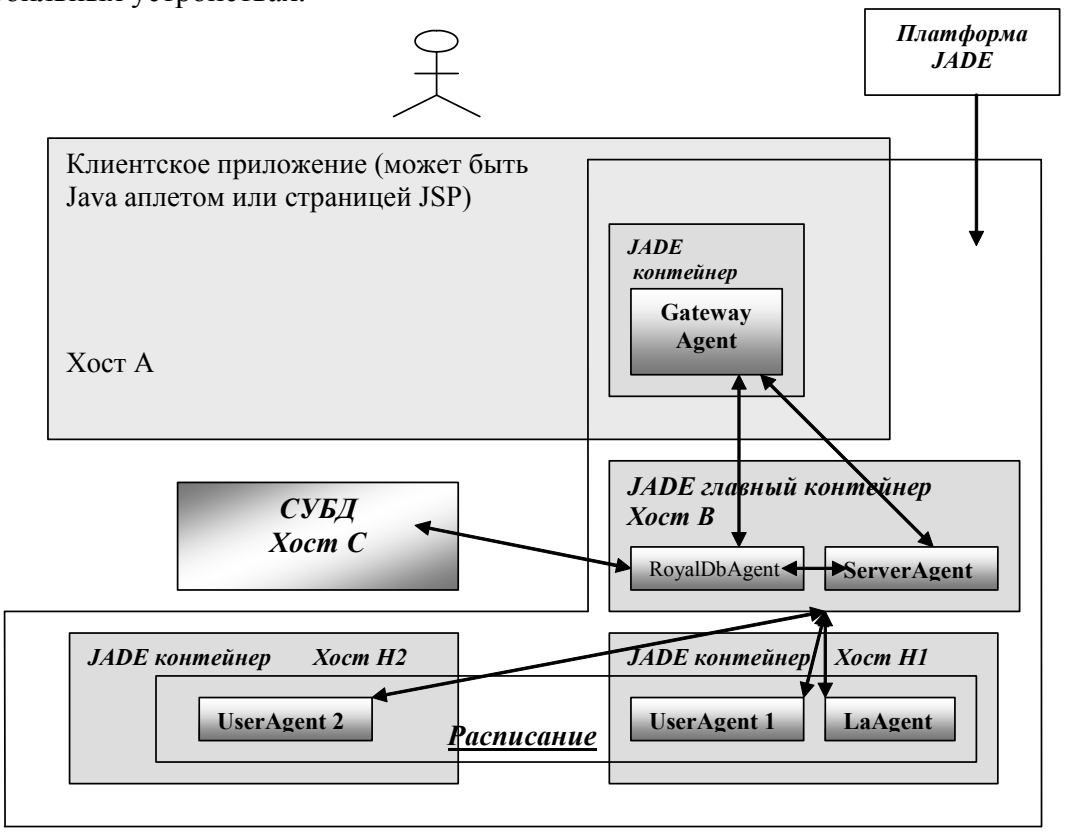


Рисунок. Состав системы и связи между компонентами

Архитектура разрабатываемого ПО составления расписания в некоторой степени зависит от архитектуры используемой многоагентной системы. Прикладная программа, использующая JADE, имеет дело со следующими объектами.

1. Агент. Программная сущность, способная исполняться в рамках системы JADE.
2. Сообщение. Агенты обмениваются информацией друг с другом и с агентной системой с помощью сообщений. Любое взаимодействие основано только на сообщениях.
3. Контейнер. Среда для исполнения нескольких агентов одновременно. Агенты загружаются в контейнер и затем запускаются. На одном компьютере может быть запущено несколько контейнеров одновременно. Один контейнер на нескольких компьютерах запущен быть не может.
4. Платформа. Является логическим объединением нескольких контейнеров. При старте платформы автоматически запуска главный контейнер (main container). Платформа поддерживает службу директорий, которая позволяет находить агентов, используя шаблон для поиска. Агенты разных платформ могут общаться друг с другом.

Состав системы и связи между компонентам условно изображены на рисунке.

Заключение

В рамках данной работы был разработан агентный алгоритм составления расписания для вуза и представлена его программная реализация.

1) Разработан агентный алгоритм составления учебного расписания. Разработанный алгоритм, помимо расписания времени, позволяет составлять еще и расписание аудиторий.
2) Алгоритм дает хорошее приближение к оптимальному решению, т.е. получаемое на выходе расписание хорошо отражает предпочтения пользователей о времени и месте проведения занятий в смысле критериев исходной задачи.
3) Разработан программный комплекс, который демонстрирует работу алгоритма на практике.
4) Измеренное на практике время работы алгоритма хорошо согласуется с теоретическими оценками вычислительной трудоемкости.
5) Эффективность работы алгоритма сопоставима с эффективностью работы человека.

Возможно продолжение исследований в данной области с целью расширения сферы применения алгоритма – например, создание на его основе многоагентной системы планирования маршрутов общественного транспорта.

Литература

1. Севастьянов С.В. Введение в теорию расписаний // Новосибирск: Новосибирский государственный университет. – 2003. – 171 с.
2. Abramson D. and Dang H. School Timetables: A Case Study in Simulated Annealing. Applied Simulated Annealing: Lecture Notes in Economics and Mathematics Systems // Springer-Verlag. – 1993. – V. 2. – P. 89–93.
3. Akkoyunly E.A. A Linear Algorithm for Computing the Optimum University Timetable // The Computer Journal. – 1973. – № 16(4). – P. 347–350.
4. BenHassine A., D´efago X., Ho T.B. Agent-Based Approach to Dynamic Meeting Scheduling Problems // Proceeding to Int. Conf. AAMAS-04. – 2004. – V. 3. – №. 1. – P. 1132–1139.
5. Petrovic S., Burke E. University Timetabling. // University of Nottingham. – 2003.
6. Shaerf A. Local Search Techniques for Large High School Timetabling Problems // IEEE Transactions on Systems, Man And Cybernetics. – 1999. – V. 29. – № 4. – P. 368–377.
7. White G.M., Chan P.W. Towards the Construction of Optimal Examination Timetables // INFOR. – 1979. – V. 17. – P. 219–229.
8. Ретинский И.М. Проектирование и реализация алгоритмов составления расписания на основе многоагентных технологий: Дис. … магистер. – Нижегородский государственный технический университет, 2005. – 97 с.

КОНТРОЛЬ НА ЧЕТНОСТЬ В ПСЕВДОСЛУЧАЙНЫХ КОДОВЫХ ШКАЛАХ ПРЕОБРАЗОВАТЕЛЕЙ ЛИНЕЙНЫХ ПЕРЕМЕЩЕНИЙ

Жуань Чжипэн
Научный руководитель – д.т.н., профессор А.А. Ожиганов

Рассматриваются однодорожечные псевдослучайные кодовые шкалы для преобразователей линейных перемещений, которые только за счет введения дополнительных считывающих элементов (СЭ) без использования дополнительной контрольной кодовой дорожки позволяют осуществлять контроль считываемой со шкалы информации на четность.

Введение

В настоящее время в цифровых преобразователях перемещений [1] широко применяются кодовые шкалы (КШ), рисунок которых выполнен в обыкновенном двоичном коде или коде Грея. Недостатком таких преобразователей является сложность изготовления КШ, так как число информационных дорожек КШ обычно равно разрядности преобразователей. Поэтому масса и габариты преобразователей с увеличением разрядности преобразователя также возрастают.

В работе [2] рассмотрены псевдослучайные кодовые шкалы (ПСКШ) для преобразователей угловых перемещений, которые по сравнению с классическими шкалами более технологичны в изготовлении и имеют меньшие габариты и массу.

Теоретические аспекты построения псевдослучайных кодовых шкал для преобразователей перемещения

ПСКШ имеют всего одну информационную кодовую дорожку, выполненную в соответствии с символами псевдослучайной двоичной последовательности максимальной длины (М-последовательности) с периодом $M = 2^n - 1$ и n СЭ, размещенными вдоль дорожки. Для генерации М-последовательности длиной $M = 2^n - 1$ используется примитивный неприводимый полином $h(x)$ степени n с коэффициентами поля Галуа GF(2) [3], т. е.

$$h(x) = \sum_{i=0}^{n} h_i x^i , \qquad (1)$$

где $h_0 = h_n = 1$, а $h_i = 0, 1$ при $0 < i < n$. Символы М-последовательности a_{n+j} удовлетворяют рекурсивному выражению

$$a_{n+j} = \overset{n-1}{\underset{i=0}{\Xi}} a_{i+j} h_i , \qquad j = 0, 1, \ldots \qquad (2)$$

где знак Ξ означает суммирование по модулю два, а индексы при символах М-последовательности берутся по модулю M. Начальные значения символов М-последовательности $a_0 a_1 \ldots a_{n-1}$ могут выбираться произвольно, за исключением нулевой комбинации. Для определенности при построении круговой ПСКШ символы М-последовательности $a_0 a_1 \ldots a_{M-1}$ отображаются на информационной дорожке по направлению движения часовой стрелки.

М-последовательности относятся к классу циклических кодов и могут задаваться с помощью порождающего полинома

$$g(x) = (x^M + 1) / h(x), \qquad (3)$$

где $h(x)$ определяется в соответствии с выражением (1), а $M = 2^n - 1$. Для каждой М-последовательности длины M существует ровно M различных циклических сдвигов, которые могут быть получены путем умножения порождающего полинома $g(x)$ на x^j,

т.е.

$$x^j g(x),\quad j = 0,1,\ldots,M-1. \qquad (4)$$

Поскольку ПСКШ строятся в соответствии с символами М-последовательности, можно путем циклических сдвигов определить порядок размещения на шкале n считывающих элементов, т.е. m-му СЭ, $m = 1, 2, \ldots, n$, ставится в соответствие j_m-й циклический сдвиг $x^{j_m} g(x)$ М-последовательности. Тогда полином, определяющий порядок размещения n СЭ на шкале, имеет вид

$$r(x) = \sum_{m=1}^{n} x^{j_m}, \qquad (5)$$

где $j_m \in \{0, 1, \ldots, M-1\}$. Положив $j_1 = 0$, согласно (5) получим положения 2-ого, 3-ого,..., n-ого СЭ, смещенные относительно первого СЭ на $j_2, j_3, \ldots, j_n$ элементарных участков информационной дорожки шкалы соответственно.

Линейная ПСКШ разомкнута, поэтому разрешающая способность такой шкалы равна $\delta_л = L/M = L/(2^n - 1)$, где L – длина кодируемого перемещения, а n – разрядность КШ. Для обеспечения заданной разрешающей способности необходимо получить соответствующую последовательность символов $\boldsymbol{A} = \{A_i\}$, $i = 0, 1, \ldots$, пригодную для синтеза единственной информационной дорожки линейной ПСКШ. Очевидно, символы последовательности $\boldsymbol{A}$ должны полностью включать в себя символы M-последовательности $\boldsymbol{a}$, а также некоторые дополнительные символы этой же последовательности, число которых зависит от выбранного полинома размещения $r(x)$ на ПСКШ СЭ.

Общее число символов последовательности $\boldsymbol{A}$ с учетом n задаваемых начальных значений может быть найдено из выражения

$$Q = M + j_n. \qquad (6)$$

Задача генерации последовательности $\boldsymbol{A}$ в общем виде решается с использованием рекурсивного соотношения (2) в предположении, что размещение СЭ на ПСКШ корректно и задается полиномом (5). Для определенности начальные значения символов последовательности $\boldsymbol{A}$ выбираются $A_0 = A_1 = \cdots = A_{n-2} = 0$, $A_{n-1} = 1$. Таким образом, последовательность $\boldsymbol{A} = \{A_i\}$, $i = 0, 1, \ldots, Q-1$ получается на основе рекурсивного выражения (2) с учетом (6).

Рисунок однодорожечной линейной ПСКШ выполняется в соответствии с символами последовательности $\boldsymbol{A}$, при отображении их на информационной дорожке шкалы слева направо в последовательности $A_0, A_1, \ldots, A_{Q-1}$.

На рис. 1 приведен пример четырехразрядной линейной ПСКШ с размещением СЭ в соответствии с полиномом $r(x) = 1 + x^3 + x^6 + x^9$ для построения которой использован примитивный полином $h(x) = x^4 + x + 1$. В данном случае информационная дорожка шкалы соответствует символам последовательности $\boldsymbol{A} = A_0 A_1 \ldots A_{23} = a_0 a_1 \ldots a_{13} a_{14} a_0 a_1 \ldots a_8 =$ 000100110101111000100110 длиной $Q = M + j_n = 15 + 9 = 24$, а начальные значения $A_0 = A_1 = A_2 = 0$, $A_3 = 1$.

Фиксируя считывающими элементами $СЭ_1$, $СЭ_2$, $СЭ_3$ и $СЭ_4$ последовательно кодовую комбинацию при перемещении линейной ПСКШ на один элементарный участок справа налево, получаем пятнадцать различных четырехразрядных кодовых комбинаций: 0111, 0010, 0001, 1111, 0101, 0011, 1110, 1010, 0110, 1101, 0100, 1100, 1011, 1001, 1000.

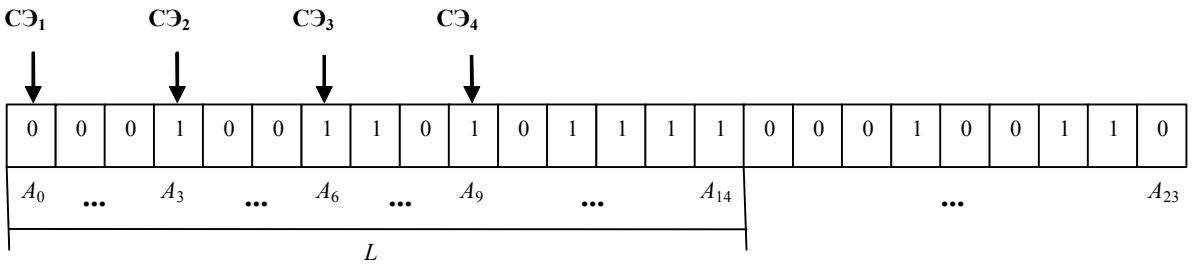


Рис. 1. Четырехразрядная линейная ПСКШ

Контроль на четность в ПСКШ преобразователей линейных перемещений

Одним из путей повышения надежности цифровых преобразователей перемещений может быть использование в них кодовых шкал с возможностью формирования корректирующих кодов. Известные методы построения преобразователей перемещений не позволяют решить эту задачу без увеличения их габаритов и массы. Это связано с применением в преобразователях перемещений классических кодовых шкал, где число информационных дорожек шкалы обычно равно разрядности преобразователя, а корректирующие возможности таких шкал могут быть реализованы только за счет введения в них дополнительных контрольных дорожек и СЭ.

Покажем, что преобразователи с ПСКШ достаточно просто позволяют организовать контроль считываемой с них информации на четность.

Известно [3], что сумма по модулю двух различных циклических сдвигов М-последовательности длины M также является циклическим сдвигом этой же М-последовательности, т.е.

$$g(x)\sum_{m=1}^{n} x^{j_m} = g(x)x^{s}, \qquad (7)$$

где знак $\sum$ означает суммирование по модулю два, $n = 2, 3, \ldots$, $(j_m \neq s) \in \{0, 1, \ldots, M-1\}$. Из (7) следует, что

$$g(x)\sum_{m=1}^{n} x^{j_m} \oplus g(x)x^{s} = 0. \qquad (8)$$

Таким образом, установка дополнительного считывающего элемента в точку ПСКШ, однозначно определяемую n информационными СЭ, позволяет организовать контроль считываемой с них информации на четность.

Теперь полином, определяющий порядок размещения на шкале n информационных и одного контрольного СЭ, имеет вид

$$r_k(x) = x^{s} + \sum_{m=1}^{n} x^{j_m}, \qquad (9)$$

где $j_m, s \in \{0, 1, \ldots, M-1\}$, $(j_m \neq s)$. Положив $j_1 = 0$, согласно (9) получим положения 2-ого, 3-ого,..., n-ого СЭ и контрольного СЭ, смещенного относительно первого СЭ на $j_2, j_3, \ldots, j_n$ и s элементарных участков информационной дорожки шкалы, соответственно. В данном случае число символов последовательности $\boldsymbol{A}$ с учетом (9) должно быть найдено из выражения

$$Q = M + j_l, \qquad (10)$$

где j_l – степень $r_k(x)$.

Рассмотрим пример четырехразрядной линейной ПСКШ, для построения которой использован примитивный полином $h(x) = x^4 + x + 1$, с организацией контроля на четность (рис. 2).

Для конкретного вычисления точки установки на ПСКШ корректирующего СЭк, используем следующий подход. Разделим полином $r(x)=1+x^3+x^6+x^9$ на $h(x)=x^4+x+1$ по модулю 2 со стороны младших степеней до получения остатка в виде одночлена x^s:

$$
\begin{array}{l|l}
1+x^3+x^6+x^9 & 1+x+x^4 \\
1+x+x^4 & \hline
1+x+x^2+x^4+x^8
\end{array}
$$

$$
\begin{array}{l}
x+x^3+x^4+x^6+x^9 \\
x+x^2+x^5 \\ \hline
x^2+x^3+x^4+x^5+x^6+x^9 \\
x^2+x^3+x^6 \\ \hline
x^4+x^5+x^9 \\
x^4+x^5+x^8 \\ \hline
x^8+x^9 \\
x^8+x^9+x^{12} \\ \hline
x^{12}
\end{array}
$$

В результате деления получен номер циклического сдвига М-последовательности ***a***, равный 12 ($x^s=x^{12}$), определяющий точку установки на шкале дополнительного корректирующего СЭ. Теперь полином, определяющий порядок размещения на шкале 4 информационных и одного контрольного СЭ, имеет вид $r(x)=1+x^3+x^6+x^9+x^{12}$.

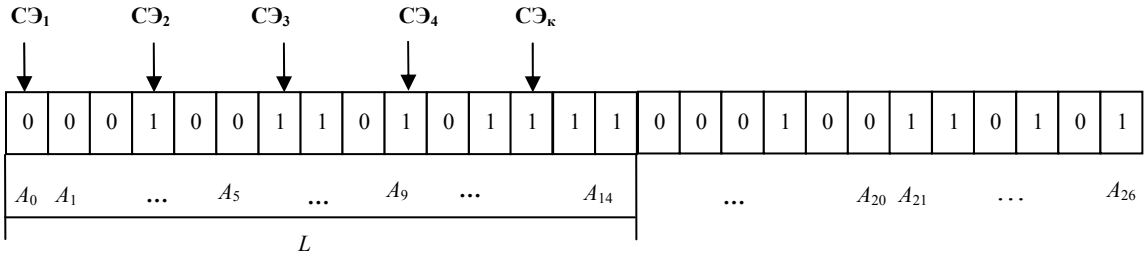


Рис. 2. Четырехразрядная линейная ПСКШ с контролем на четность

На рис. 2 дополнительный корректирующий СЭ*к* смещен относительно первого информационного СЭ на 12 элементарных участков шкалы. Информационная дорожка шкалы соответствует символам последовательности ***A***= $A_0A_1\ldots A_{26} = a_0a_1\ldots a_{13}a_{14}a_0a_1\ldots a_{11}$ = 000100110101111000100110101 длиной $Q=M+j_l=15+12=27$, а начальные значения $A_0=A_1=A_2=0$, $A_3=1$.

Фиксируя считывающими элементами СЭ$_1$, СЭ$_2$, СЭ$_3$, СЭ$_4$ и СЭ$_k$ последовательно кодовую комбинацию при перемещении линейной ПСКШ справа налево, получаем пятнадцать различных четырехразрядных кодовых комбинаций: 01111, 00101, 00011, 11110, 01010, 00110, 11101, 10100, 01100, 11011, 01001, 11000, 10111, 10010, 10001. Эти комбинации содержат четное число единиц и соответствуют пятнадцать различным положениям ПСКШ.

Заключение

Таким образом, ПСКШ для преобразователей линейных перемещений по сравнению с классическими КШ имеют более простую кодовую маску и позволяют всего за счет одного дополнительного СЭ организовать контроль считываемой информации на четность.

Литература

1. Домрачев В.Г., Мейко Б.С. Цифровые преобразователи угла: принципы построения, теория точности, методы контроля. – М.: Энергоатомиздат, 1984.–328 с.
2. Ожиганов А.А. Псевдослучайные кодовые шкалы // Приборостроение. – 1987. – Т.30. – № 2. – С.40–43.
3. Макуильямс Ф.Д., Слоан Н.Д. Псевдослучайные последовательности и таблицы // ТИИЭР. – 1976. – Т.64. – № 12. – С.80–95.

МОДЕЛИРОВАНИЕ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ В РАМКАХ ОДНОГО КОМПЬЮТЕРА

Д.В. Корнев (Уральский государственный университет им. А.М. Горького)
Научный руководитель – М.В. Баклановский
(Уральский государственный университет им. А.М. Горького)

При создании и отладке протоколов разработчику необходимо тестировать свою реализацию, создавая различные сетевые конфигурации и отладочные ситуации. В случае протоколов, подразумевающих одновременное общение большого количества узлов сети, отладка становится затруднительной. Для решения этих задач нами был разработан и реализован открытый документированный моделирующий стенд.

Введение

При создании и отладке протоколов разработчику необходимо тестировать свою реализацию, создавая различные сетевые конфигурации и отладочные ситуации. В случае протоколов, подразумевающих одновременное общение большого количества узлов сети, например в случае peer-to-peer либо *PN*-протоколов [1], отладка становится затруднительной. Решить эту проблему можно при помощи моделирования сетевого взаимодействия в рамках одного компьютера, что позволит с достаточной точностью прогнозировать поведение протокола в реальной сети.

Для решения этих задач нами был разработан и реализован открытый документированный стенд, позволяющий моделировать при помощи одного вычислителя протоколы, начиная от транспортного уровня стека ISO/OSI [2, 3] и выше. При помощи стенда разработчик может собрать сеть с произвольной топологией, задав различные ее физические характеристики. Разработчик может, воспользовавшись уже готовыми примитивами либо реализовав собственные программные модули, отвечающие интерфейсам стенда, задать поведение, сценарий, настроить маршрутизацию каждого узла моделируемой сети. Результаты моделирования доступны как в онлайновом режиме, так и в оффлайновом.

В данный момент, при помощи моделирующего стенда, идет разработка протоколов вычислительного грида [4] Ghostgrid [5] и моделируется игра «бой в кластере».

Мотивация для собственной реализации

Во-первых, при моделировании протокола разработчику необходим достаточный уровень детализации и большое количество настраиваемых параметров, что обеспечивает больший диапазон возможных тестовых ситуаций.

Во-вторых, желательна высокая скорость моделирования, которую невозможно достичь при помощи неспециализированных средств виртуализации.

В-третьих, крайне важна реалистичность результатов, что означает совпадение результатов моделирования протоколов и результатов их применения на практике.

В-четвертых, многие средства моделирования предоставляют язык описания протоколов или набор некоторых базовых примитивов. Данный подход может оказаться неприемлемым, например, в случае разработки протоколов, использующих сложные алгоритмы или структуры данных. В таком случае разработчику удобнее описывать свой протокол на каком-либо широко распространенном языке программирования, используя API для работы с моделируемой сетью.

Моделируемая среда

Ключевую роль в моделировании играет среда передачи данных, обеспечивающая доставку пакетов от одного узла сети к другому. Стенд позволяет задавать сеть произвольной топологии (см. рис. 1), состоящую из подсетей, соединенных между собой

маршрутизаторами. Каждая подсеть работает по принципу CSMA/CD (Carrier Sense Multiple Access with Collision Detection – множественный доступ с контролем несущей сети и обнаружением коллизий). В сети возможно наличие транспортных подсетей.

Помимо топологии, задаются физические характеристики сети, такие как частота случайных коллизий – для моделирования загруженных каналов связи и снижения пропускной способности подсети, надежность подсети – параметр, позволяющий устанавливать процент пропадание пакетов в подсети, например, для отладки отказоустойчивости протоколов.

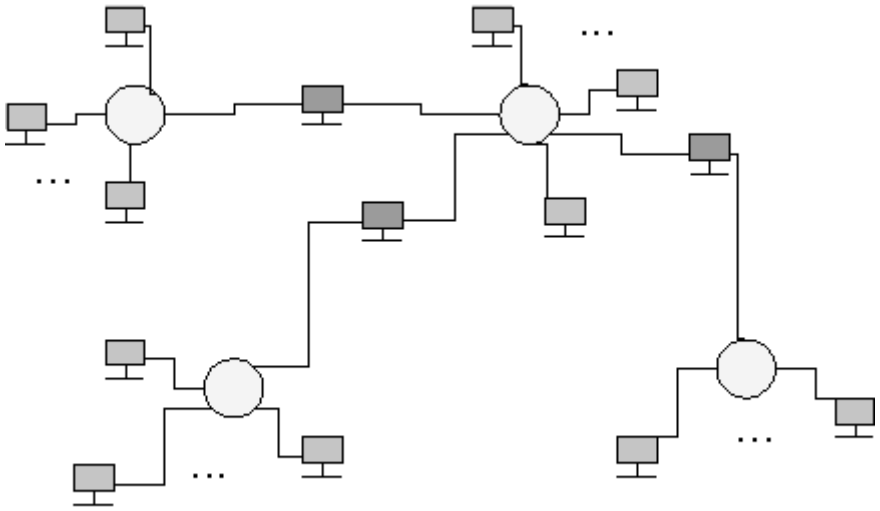


Рис. 1. Сеть произвольной топологии

Среда является тактированной. На каждом такте происходит последовательная обработка каждой из подсетей. Внутри одной подсети происходит последовательная обработка каждого из узлов, в процессе которой узел может обработать очередь входящих пакетов, изменить свое внутреннее состояние, положить некоторые пакеты в исходящую очередь, сделать запись в log-файл, сгенерировать событие.

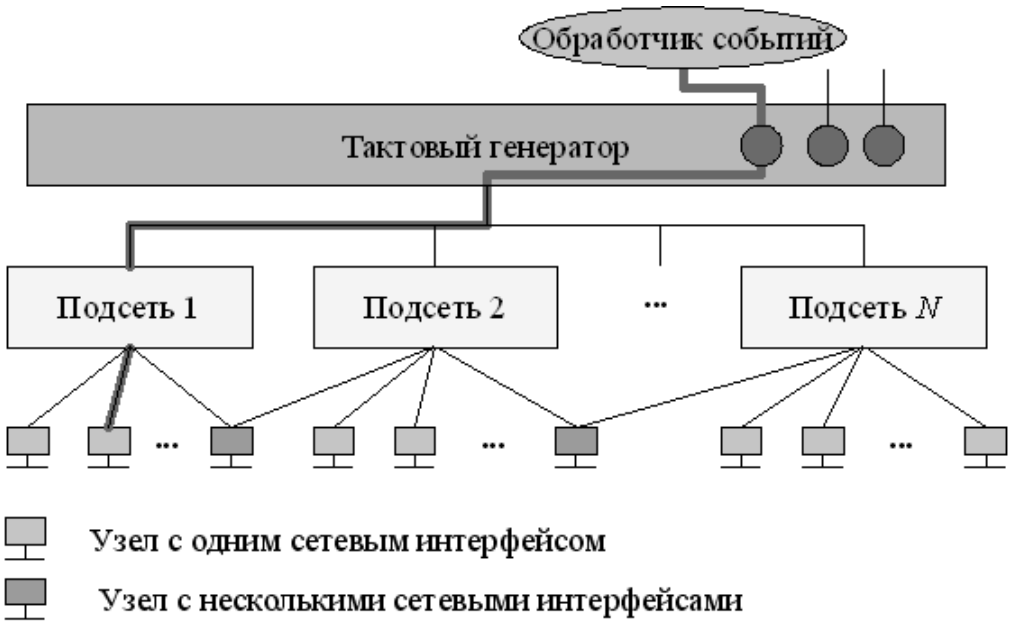


Рис. 2. Устройство стенда, всплывание событий

Каждому узлу разработчиком может быть назначен его квант времени. Перед передачей управления узлу засекается время, по возвращении управления происходит вычисление, какое количество времени узел проработал и сколько квантов времени он выработал. Если узлом было использовано, к примеру, два кванта, то на следующем такте управление ему передаваться не будет. Данная функциональность была добавлена для моделирования протоколов, обеспечивающих параллельные или распределенные вычисления, чтобы вычисления на узлах не происходили всегда за один такт.

Сгенерированные узлом события всплывают (см. рис. 2). На некотором уровне может быть обработчик, зарегистрированный на данное событие. Примером использования данной возможности может служить визуализация внутренних параметров узла во время моделирования. Для этого разработчик пишет модуль с заданным интерфейсом, который умеет обрабатывать необходимый тип событий, графически отображая получаемую из них информацию, затем обучает интересующий узел испускать события данного типа и регистрирует модуль в качестве обработчика. Такая схема очень удобна, поскольку позволяет разработчику наблюдать за ситуацией непосредственно во время моделирования, а не постфактум во время изучения логов.

Моделируемые узлы

Моделирующий стенд предоставляет разработчикам среду передачи данных, API по работе со средой и интерфейсы для подключения узлов. При этом среда реализует физический, канальный и сетевой уровни модели ISO/OSI. Верхние уровни стека должен реализовывать разработчик (см. рис. 3).

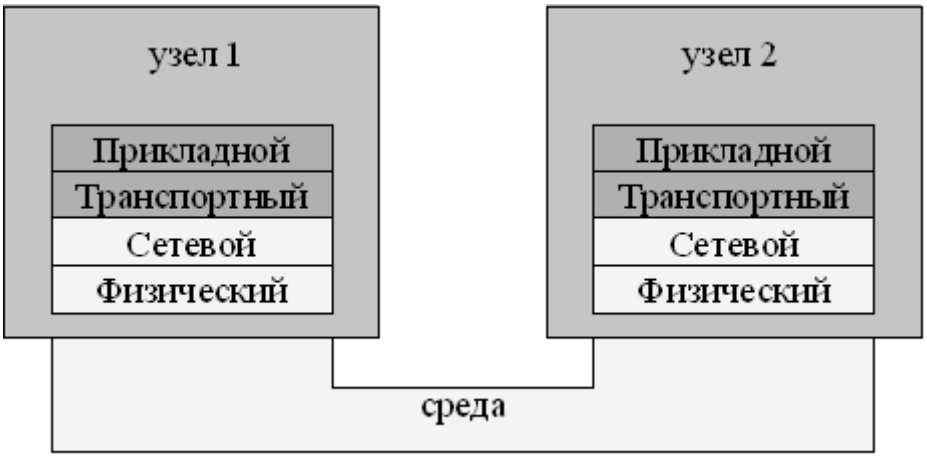


Рис. 3. Разделение стека ISO/OSI между средой и узлами

Для каждого узла в сети можно задать следующие параметры: количество сетевых интерфейсов и их настройки, «автомат» узла, сценарий узла, тип маршрутизации, параметры входящих и исходящих очередей. Все эти настройки производятся в конфигурационном файле. Под «автоматом» узла мы понимаем набор правил приема, обработки и испускания пакетов. Данный набор правил необходимо запрограммировать в виде модуля с заданным интерфейсом, используя предоставляемый стендом API. Автомат узла будет вызываться всегда, когда происходит передача управления от тактового генератора к узлу.

Задание только лишь автоматов всех узлов не приведет к активности в сети, если все узлы пассивны, например, если все узлы являются серверами. Чтобы узлы генери-

ровали пакеты, разработчик должен задать сценарий работы узла так же, как и автомат узла. Если у узла задан сценарий, то он будет вызываться после автомата.

Под типом маршрутизации понимается набор правил, по которым узел должен маршрутизировать свои пакеты. В стенде предусмотрена статическая маршрутизация по кратчайшим по числу хопов маршрутам.

В модулях автомата и сценария узла есть возможность использовать не только сетевой API, но также API по работе с log-файлами и API генерации всплывающих событий.

Применение стенда

При помощи данного стенда ведется разработка системы распределенных вычислений Ghostgrid [5], основной особенностью которой является отсутствие детерминистических модулей принятия решения. Стенд позволяет исследовать поведение протоколов, основанных на нейросетевых технологиях.

Планируется моделировать специализированный протокол CTP (Cluster transport protocol).

Представляет большой интерес моделирование игры «бой в кластере», разработанной в рамках соревнований по системному программированию EACP.

Заключение

Разработчики протоколов часто нуждаются в средствах отладки и тестирования. При этом существующие средства имеют различные ограничения, такие как отсутствие возможности использовать сложные алгоритмы и структуры данных, ограничения в производительности.

Нами был разработан и реализован открытый документированный стенд, позволяющий моделировать при помощи одного вычислителя протоколы, начиная от транспортного уровня стека ISO/OSI и выше.

В данный момент при помощи моделирующего стенда идет разработка протоколов вычислительного грида Ghostgrid, моделируется игра «бой в кластере», исследуется протокол CTP.

Литература

1. Баклановский М.В., Корнев Д.В., Попов В.Ю. Некоторые проблемы моделирования и отладки протоколов // Проблемы теоретической и прикладной математики. Труды 37-й Региональной молодежной конференции. – Екатеринбург: УрО РАН, 2006. – С. 404–407.
2. ISO/IEC standard 7498-1. Open Systems Interconnection, 1994.
3. Таненбаум Э. Компьютерные сети. – СПб: Питер, 2003.
4. Баклановский М.В., Корнев Д.В., Попов В.Ю. Гриды как новое поколение Интернета // Информационно-математические технологии в экономике, технике и образовании. Труды Международной научная конференции. – Екатеринбург: ГОУ ВПО УГТУ-УПИ, 2006. – С. 202–203.
5. Баклановский М.В., Корнев Д.В. Искусственный интеллект в Ghostgrid // Интеллектуальные системы и компьютерные науки. Труды IX Международной конференции. – М.: Изд-во механико-математического факультета МГУ, 2006. – С. 54–55.

МЕТОД КОЛИЧЕСТВЕННОЙ ОЦЕНКИ ЦВЕТОВЫХ РАЗЛИЧИЙ ПРИ ВОСПРИЯТИИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

М.Д. Хорунжий
(Санкт-Петербургский государственный университет кино и телевидения)
Научный руководитель – д.т.н., профессор Г.В. Тихомирова
(Санкт-Петербургский государственный университет кино и телевидения)

Определение цветовых различий между двумя стимулами является важной задачей колориметрии и цветовоспроизведения в информационных системах. Одной из задач моделирования качества цветных цифровых изображений является установление взаимосвязи между субъективной оценкой цветовых ощущений и объективными критериями цветовых различий между парами изображений, без необходимости знания факторов, искажающих изображения.

Введение

Известно, что если некоторое время смотреть на один цвет, а затем перевести взгляд на другой, то первое время будет наблюдаться искаженное восприятие этого цвета. Цвета воспринимаются несколько иначе и под влиянием других, окружающих или соприкасающихся с ними цветов. Подобные явления называются контрастом цветов. Различают одновременный и последовательный контраст [1]. Последовательный контраст – изменение цвета в результате предварительного воздействия на глаз других цветов, а одновременный контраст – изменение цвета под влиянием окружающих цветов [2].

Цель настоящей статьи заключается в получении надежной психофизической базы данных для исследования хроматического и светлотного контраста больших объектов изображений с их точным спектральным измерением. Это задача решается с помощью создания структуры исследуемых цветных стимулов, ранжированных по субъективным предпочтениям, а также аналитически полученным оценкам межстимульного цветового различия. Математическая обработка экспертных оценок и объективных параметров осуществляется с использованием методов многомерного шкалирования. Построенная структура стимулов позволяет проанализировать объективную и субъективную информацию о близостях между стимулами и предъявить требования к цветовоспроизведению цифровых изображений. Предложенный метод представляет первую стадию в разработке модели появления изображений, создаваемую для предсказания цветовых различий, имеющих широкий диапазон, и моделирования качества цифровых изображений.

1. Теоретические основы исследования

Ощущаемое цветовое различие характеризуется пороговой чувствительностью, т.е. минимальным требуемым различием между двумя стимулами, которое выявляется как ощущаемое различие. Если евклидовы расстояния между двумя точками в цветовом пространстве пропорциональны ощущаемым цветовым различиям, цветовое пространство является равномерным, которое оптимально для установления значений допусков на цвет в системах цветовоспроизведения. Известно, что пространство XYZ не является равномерно ощущаемым [3]. Множество попыток исследователей привели к созданию МКО равномерных цветовых пространств – LUV и LAB. Цветовое пространство LUV еще применяется для исследований, но, вероятно, наиболее широко используемое пространство – LAB, несмотря на то, что оно не может рассматриваться как равномерное, как это утверждается, и многие свойства цвета данная модель не может предсказать [4]. Цвет определяется в этом пространстве через координаты L^*, a^*, b^*, которые получают преобразованием координат цвета X, Y, Z [4]*:*

$$L^* = 116 f(Y/Y_n) - 16 \quad (1.1)$$

$$a^* = 500\left[f(X/X_n) - f(Y/Y_n)\right] \quad (1.2)$$

$$b^* = 200\left[f(Y/Y_n) - f(Z/Z_n)\right], \quad (1.3)$$

где функция $f(\cdot)$ определяется как

$$f(x) = x^{1/3}, \quad x > 0{,}008856, \quad (1.4)$$

$$f(x) = 7{,}787x + \frac{16}{116}, \; x \le 0{,}008856, \quad (1.5)$$

где X_n, Y_n и Z_n – нормированные значения координат цвета белой точки [4]. Аналогично могут быть рассчитаны цветовой тон и насыщенность:

$$C^*_{ab} = \sqrt{a^2 + b^2}\,. \quad (1.6)$$

$$h_{ab} = \tan^{-1}(\frac{b^*}{a^*})\,. \quad (1.7)$$

Цветовое различие в LAB рассчитывается как евклидово расстояние между двумя цветами в цветовом пространстве [5]:

$$\Delta E^*_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}\,. \quad (1.8)$$

Преобразования LUV и LAB относительно просты, но они не дают результат в весьма равномерных по ощущениям пространствах, а только в приблизительно равномерных. В добавление, окружающие факторы, такие как внешнее освещение, цвет фона и т.д., также влияют на различие цветов, но не входят в преобразования. Так как в настоящий момент снижается применение пространства LUV, исследования в данной статье основываются на цветовом пространстве LAB.

Использование величин яркости цвета L^*, насыщенности C^*_{ab}, цветового тона h_{ab} может способствовать интуитивному пониманию цветового пространства LAB, относя их к перцептивным свойствам цветов. Важное значение имеет оценивание каждой из величин пространства LAB раздельно. Для этого разбиваем цветовое различие на составляющие яркости, насыщенности и цветового тона, сумма квадратов которых и даст нам искомое цветовое различие ΔE^*_{ab}, а величина ΔH^* называется МКО 1976 различием по цветовому тону, рассчитываемым в соответствии со следующей формулой [5]:

$$\Delta H^* = \sqrt{(\Delta E^*_{ab})^2 - (\Delta L^*)^2 - (\Delta C^*_{ab})^2}\,. \quad (1.9)$$

Формула цветовых различий по выражению (1.8) предполагает заданное измерение цветовых различий, в соответствии с ощущаемым различием. Однако, как было сказано, цветовое пространство МКО LAB не является всецело равномерным, а цветовое различие ΔE^*_{ab} несовершенно. Проводились попытки найти оптимальное выражение для цветового различия, например формула CMC [6]-[7], формула BFD [8], сферическая модель цветоразличения [9], модель МКО94 [10]. Необходимо отметить, что во всех разрабатываемых моделях цветовое различие не должно превышать пяти порогов чувствительности ($\Delta E_{ab} \le 5$). Поэтому в настоящей статье использовался приближенный метод количественной оценки по широкоизвестным выражениям МКО LAB 1976 [11].

Для ранжирования попарно стимулов по степени различия цвета и выявления количественной и качественной взаимосвязи при парном сравнении групп исследуемых свойств стимулов (светлоты, контраста, насыщенности и цветового тона) используются методы многомерного шкалирования [12]. Задача многомерного шкалирования в самом общем виде состоит в том, чтобы выявить структуру исследуемого множества стимулов. Процедура построения структуры опирается на анализ объективной или субъек-

тивной информации о близостях между стимулами либо информации о предпочтениях на множестве стимулов.

Для построения искомого координатного пространства используется достаточно разработанный аппарат линейной или нелинейной оптимизации. Обычно используется метрика Минковского [13]:

$$d_{jk} = \sqrt[p]{\sum_{t=1}^{r} \left| x_{jt} - x_{kt} \right|^{p}}, \qquad (1.10)$$

где r – размерность пространства, d_{jk} – расстояние между точками, соответствующими j-му и k-му стимулам, x_{jt}, x_{kt} – величины проекций j-й и k-й точек на t ось. Количественная оценка хроматического и светлотного контраста рассчитывается по формулам (1.11)–(1.14).

Относительный контраст по насыщенности *T1*

$$T1 = \frac{C^{*}_{\max} - C^{*}_{\min}}{C^{*}_{\max} + C^{*}_{\min}}, \qquad (1.11)$$

где $C^{*}_{\max}$, $C^{*}_{\min}$ – максимальное и минимальное значение насыщенности в сопоставляемых полях. Параметр $K_к$ и $K_з$ можно определить по следующим формулам:

$$K_к = \frac{\Delta H_K}{h_{Кэт}}, \; K_З = \frac{\Delta H_З}{h_{Зэт}}, \qquad (1.12)$$

где $\Delta H_к$, $\Delta H_з$ – различие по цветовому тону для красного и зеленого тестовых полей соответственно; $h_{Кэт}, h_{Зэт}$ – значения цветового тона на эталоном зеленом и красном тестовом поле соответственно.

Относительный контраст по цветовому тону *T2*

$$T2 = \frac{K_{\max} - K_{\min}}{K_{\max} + K_{\min}}, \qquad (1.13)$$

где K_{max} и K_{min} – максимальное и минимальное значение параметра в сопоставляемых полях. Изменение светлоты стимулов ΔL можно определить по следующей формуле:

$$\Delta L = L_{\text{пробы}} - L_{\text{эталона}}, \qquad (1.14)$$

где $L_{\text{пробы}}, L_{\text{эталона}}$ – яркость цвета пробы и эталонного стимула соответственно.

2. Экспериментальная часть исследований

Методика эксперимента

В качестве исследуемого цифрового изображения применили эталонный тестовый стимул «Peppers» из базы изображений размером 512×512 пикселей. В ходе эксперимента в него вносились искажения в виде изменения светлоты, контраста, насыщенности и цветового тона. Набор стимулов был предварительно подготовлен с использованием компьютерной программы MATLAB R2007a. На протяжении всего эксперимента исследовались два цветных поля в изображении, условно названные «перец зеленый» и «перец красный». Все стимулы были представлены в виде полноэкранных цветных слайдов в формате Power Point.

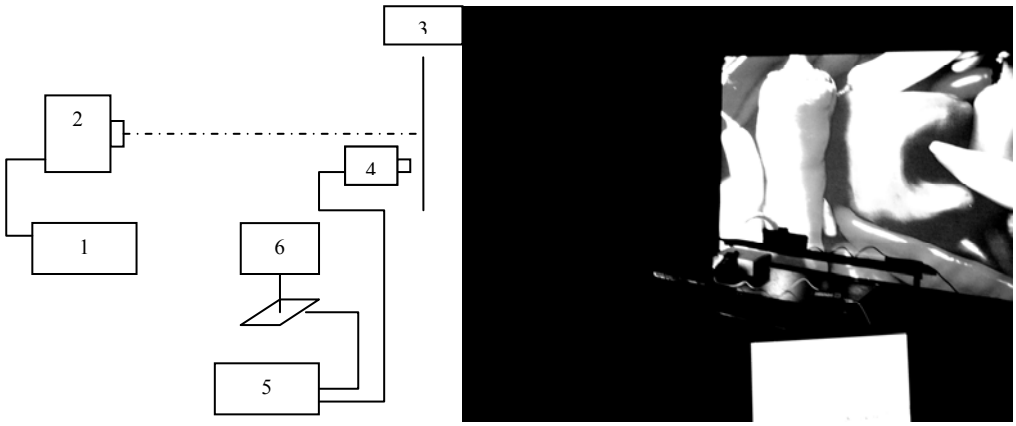


Рис. 2.1. Общая схема экспериментальной установки

Схема экспериментальной установки приведена на рис. 2.1. Сигнал с персонального компьютера 1 поступал в видеопроектор 2, проецирующий цветные стимулы на экран 3. Входной зрачок объектива видеопроектора 2 располагался на расстоянии 3,8 м от экрана. Спектроколориметр 4 был установлен на расстоянии 400 мм от экрана. Высота экранного изображения составила $H_Э$ = 1470 мм, ширина экранного изображения $B_Э$ = 1960 мм. Сигнал с измерительного прибора подавался на персональный компьютер 5, с соответствующим программным обеспечением. Результаты спектральных измерений спектроколориметра в системе МКО 1931 отображались на мониторе 6 и затем были подвергнуты преобразованию в систему МКО LAB 1976 с источником D65 и 2° стандартным наблюдателем.

2.1.1. Измерительный прибор

Спектральное распределение модулированного светового потока и координаты цвета измерялись с использованием спектроколориметра ТКА-ВД/01 и спектрофотометра GretagMacbeth X-Rite [14]. Значения координат цвета для цветных стимулов рассчитывались с учетом МКО 1931 стандартного 2° наблюдателя. Цветовые различия были рассчитаны между соответствующими пробами и эталоном.

2.1.2. Видеопроектор

В экспериментальной установке применялся видеопроектор InFocus LP820, который вначале тестировался на равномерность яркости по белому тестовому полю и координаты цветности МКО x и y. Измерения спектроколориметром ТКА-ВД/01 проводились в проходящем свете. Измерения показали, что девиация координат цветности x и y белой точки для данных условий проекции лежит в допустимых пределах x = 0,304, y = 0,352 (значения координат цветности в соответствии со стандартом DCI составляют $x_н$ = 0,314, $y_н$ = 0,351 [15]).

2.2. Психофизический эксперимент

Цель экспертного оценивания – определить пороговую чувствительность зрительного анализатора (ЗА) человека к изменению влияющих факторов – контраста, светлоты, цветового тона и насыщенности. Каждый эксперимент содержал эталонное изображение и серию из 11 измененных изображений.

Испытуемые. В эксперименте участвовали девять испытуемых студентов СПбГУКиТ (пять студентов мужского пола и четверо женского) на момент проведения эксперимента в возрасте от 19 до 21 лет, с нормальным цветовым зрением, проверенным по полихроматическим таблицам Рабкина и Юстовой [16]–[17].

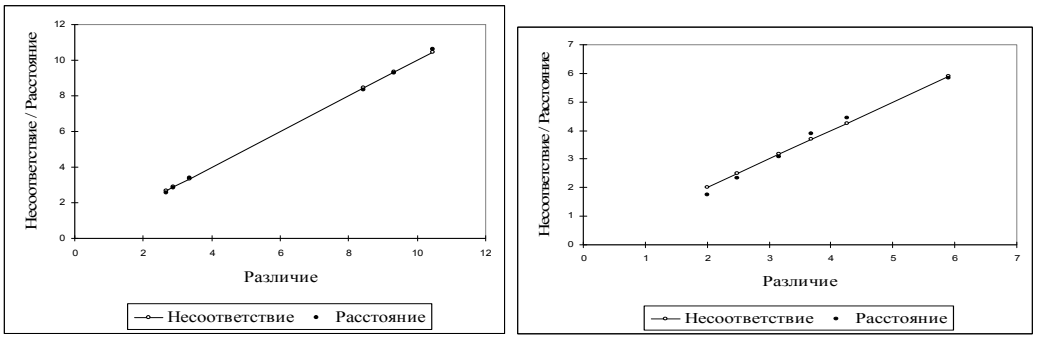
Каждое эмпирическое наблюдение проводилось в совершенно темной комнате, с соблюдением фотопических условий. Изображения были представлены на белом фоне с яркостью 158 кд/м^2. В течение проведения эксперимента каждый испытуемый был полностью независим от других. Расстояние от зрительной системы испытуемых до экрана составляло 3,80 м. В экспериментах использовался эффект повышенной цветовой адаптации зрения испытуемых. Межстимульный интервал составлял 0,5 с, а время предъявления стимула составляло 7 с. Всего было 44 стимула, из них 11 стимулов с изменением светлоты, 11 стимулов с изменением контраста, 11 стимулов с изменением насыщенности, 11 стимулов с изменением цветового тона.

Инструкция. Испытуемому предъявлялись пары стимулов, и он оценивал степень отличия стимулов целыми числами от 0 (идентичные стимулы) до 9 (максимально различные стимулы). Оценивание изображений в баллах производилось с использованием процедуры анкетирования. Предъявления проводились в случайном порядке. Одинаковые стимулы в паре испытуемым не предлагались, но испытуемые не знали об этом. В случае каждого наблюдателя две серии оценивания изображений проводились для объектов, соответствующим спектральным измерениям, на протяжении отображения всех стимулов. Следующие вопросы задавались испытуемым: 1. Обладают ли два стимула одинаково ощущаемым цветом? 2.Обладают ли два предъявляемых стимула одинаково ощущаемым хроматическим контрастом? Максимальное различие испытуемые не определяли, каждый из них оценивал максимальное различие в соответствии со своим зрением.

2.3. Метод количественной оценки цветовых различий

На следующем этапе использовалась методика расчета цветовых различий между изображениями. В случае количественной оценки цветового различия балльные оценки усреднялись по девяти испытуемым и сводились в матрицу попарных цветовых различий между стимулами для четырех исследуемых случаев. Полученная матрица служит входной информацией для дальнейшего анализа. Далее с использованием компьютерной программы xlstat2008 было построено координатное пространство, и в нем размещены исследуемые стимулы таким образом, чтобы расстояния между ними, определяемые по введенной метрике, наилучшим образом соответствовали исходным различиям между стимулами. Затем полученные результаты в виде расстояний, т.е. степени отличий в цвете между группами стимулов, сопоставлялись с результатами аналитических вычислений, прошедшими аналогичную процедуру многомерного шкалирования, через построение сенсорной зависимости, приведенной на рис. 2.5–2.6.

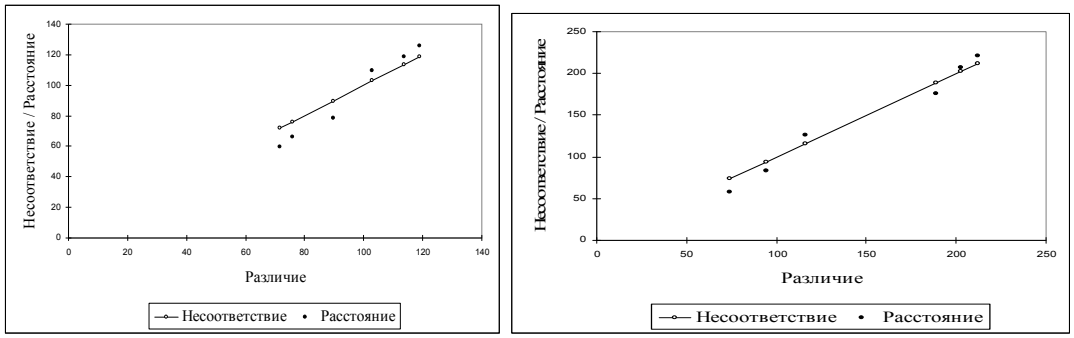
2.3.1. Построение многомерного субъективного и объективного координатного пространства исследуемых цветовых характеристик стимулов



а б

Рис. 2.2. Диаграмма Шепарада
а) при восприятии измененного цвета стимулов (групповой “стресс” = 0,057);
б) при восприятии измененного хроматического контраста (групповой “стресс” = 0,190)

Из графика на рис. 2.4 видно, что рассчитанные аналитически объективные расстояния между парами стимулов «перец зеленый» и «перец красный» увеличиваются с увеличением субъективных расстояний между этими стимулами. Из графика также следует, что в первом случае соотношение пар стимулов светлота–контраст, контраст–насыщенность и светлота–насыщенность для обоих рассматриваемых случаев воспринимается одинаково.



а б

Рис. 2.3. Диаграмма Шепарада
а) при измерении тестового поля “перец зеленый” (групповой “стресс” = 470,804);
б) при измерении тестового поля “перец красный” (групповой “стресс” = 735,657)

№ пары стимулов	Субъективные расстояния между парами стимулов	Объективные расстояния между парами стимулов “перец зеленый”	Объективные расстояния между парами стимулов “перец красный”	Пары стимулов
1	2,556	118,47	125,75	Светлота – Контраст
2	2,819	66,25	58,31	Контраст – Насыщенность
3	3,415	78,75	82,95	Светлота – Насыщенность
4	8,330	126,06	207,22	Светлота – Цветовой тон
5	9,292	59,75	175,99	Насыщенность – Цветовой тон
6	10,609	109,25	221,16	Контраст – Цветовой тон

Таблица. Данные, полученные из модели многомерного шкалирования

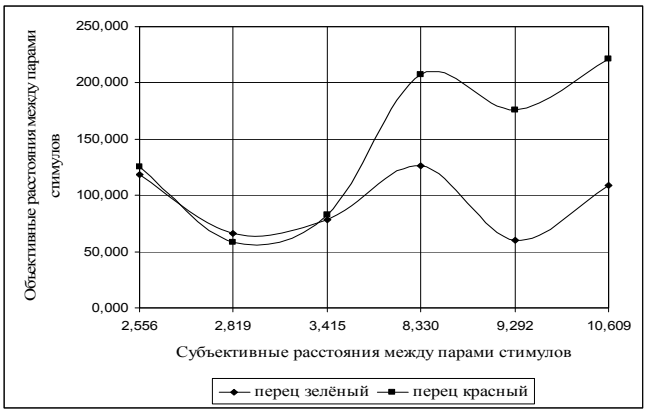


Рис. 2.4. Сенсорная характеристика по межстимульному изменению свойств цвета с использованием метода многомерного шкалирования

Гораздо более существенное различие наблюдается во втором случае для остальных трех пар стимулов: светлота–цветовой тон, насыщенность–цветовой тон и контраст–цветовой тон. Наибольшее различие в расстоянии между стимулами наблюдается в паре контраст–цветовой тон. Следовательно, в соответствии с расчетами испытуемые почти не ощущали цветовых различий в первом случае и, напротив, выявляли изменение свойств цвета во втором случае.

2.3.2. Сенсорные характеристики по хроматическому контрасту и светлоте между парами стимулов

По формулам (1.6), (1.9) и (1.11–1.14) рассчитали относительный хроматический контраст между парами стимулов «перец зеленый» и «перец красный» при относительном изменении насыщенности и цветового тона. Результаты расчетов в виде графиков сенсорных характеристик представлены на рис. 2.5.

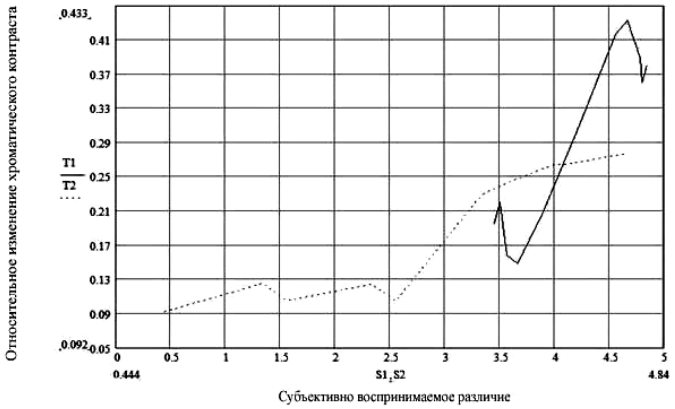


Рис. 2.5. Сопоставление сенсорных характеристик по хроматическому контрасту: T1 – при относительном изменении цветового тона; T2 – при относительном изменении насыщенности

Из анализа графика на рис. 2.5 следует, что хроматический контраст более заметен при изменении цветового тона, чем при изменении насыщенности в изображении. В диапазоне относительных единиц контраста 0,15–0,43 наблюдается линейная зависимость от экспертной оценки цветовых различий. При сопоставлении диапазонов экс-

пертных оценок в логарифмических единицах получим $\frac{20\lg\frac{4,5}{0,5}}{20\lg\frac{4,8}{3,5}} = 6,95$, т.е. диапазон насыщенности примерно в семь раз превышает диапазон цветового тона. Аналогично, при сопоставлении диапазонов по хроматическому контрасту в логарифмических единицах получим $\frac{20\lg\frac{0,28}{0,09}}{20\lg\frac{0,43}{0,15}} = 1,07$, т.е. они примерно совпадают для обоих случаев, что и требовалось обосновать в данной работе. Следовательно, из проведенных расчетов можно сделать вывод, что два цветных изображения будут обладать одинаковыми рассчитываемыми цветовыми различиями от эталонного изображения, несмотря на то, что одно ощущается экспертами более контрастным, а второе менее контрастным. Кроме того, при повышении насыщенности изображения улучшается субъективное восприятие хроматического контраста.

По формуле (1.14) определили значения изменения яркости цвета для обоих рассматриваемых случаев. Результаты расчеты представлены в виде графика на рис. 2.6.

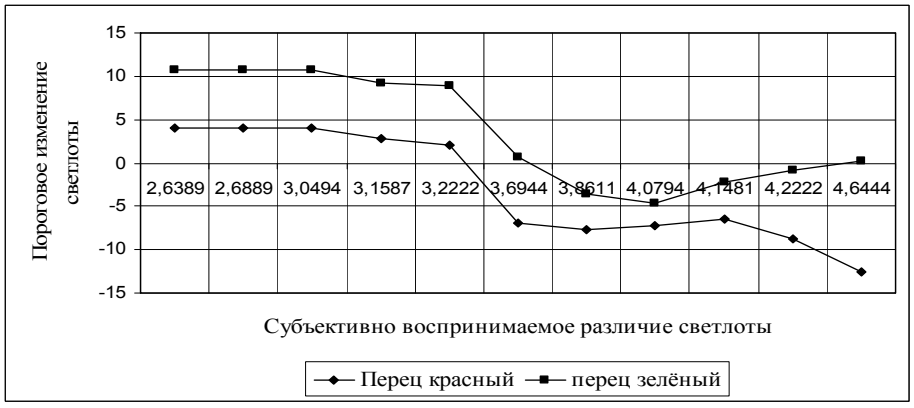


Рис. 2.6. Сенсорная характеристика по светлоте

При повышенной светлоте стимулов хроматический контраст менее заметен, как следует из графика на рис. 2.6, вследствие появления светлотного контраста между стимулами «перец зеленый» и «перец красный», снижающего восприятие хроматического контраста. Это можно объяснить, как более слабую реакцию зрительного анализатора в этих условиях в соответствии с законом Вебера-Фехнера.

Заключение

Определение цветовых различий между двумя стимулами является важной задачей колориметрии и цветовоспроизведения. Основным объективным критерием для разработки формул цветовых различий является обеспечение высокой степени совпадения, насколько это возможно, полученных результатов с экспертными оценками цветовых различий. Полученные результаты позволяют сделать вывод об адекватности применения модели цветового различия LAB. В статье проведено исследование хроматического и светлотного контраста больших объектов изображений с их точным спектральным измерением. Предложенный метод, основанный на многомерном шкалировании, позволяет выявить свойства цвета и структурировать факторы, оказывающие влияние на их изменение. Метод может служить основой для разработки программной модели появления изображений с построением сенсорных зависимостей.

Литература

1. Кирилов Е.А. Цветоведение: Учебное пособие для вузов. – М.: Легпромбытиздат, 1987. – 128 с.
2. Артюшин Л.Ф. Цветоведение. – М.: Книга, 1982. – 200 с.
3. Соколов Е.Н., Измайлов Ч.А. Цветовое зрение. – М.: Издательство, 1984. – 175 с.
4. CIE LAB Color Space. – Режим доступа: http://www.hunterlab.com/appnotes/an07_96a.pdf
5. Fairchild M. Color appearance models. Reading, MA: Addison–Wesley, 1998.
6. Clarke, F.J.J., R. McDonald, and B. Rigg. Modification to the JPC79 colour-difference formula. Journal of the Society of Dyers and Colourists 1984, 100: 128-132 and 281-282.
7. McLaren, K. Colour passing – visual or instrumental? // Journal of the Society of Dyers and Colourists. – 1970. – 86. – P. 389–392.
8. Luo, M.R., and B. Rigg. BFD(l:c) colour difference formula. Part I: Development of the formula // Journal of the Society of Dyers and Colourists. – 1987. – 103. – P. 86–94.
9. Измайлов Ч.А. Сферическая модель цветоразличения. – М.: Изд-во МГУ, 1980.
10. CIE (Commission Internationale de l'Éclairage). 1993. Technical report: Parametric effects in color-difference evaluation, CIE Publ. 101. Vienna: Central Bureau of the CIE.
11. Домасев М. Цвет, управление цветом, цветовые расчеты и измерения. – Режим доступа: http://www.nordicdreams.net.ru/articles/colorimetry.302.html
12. Torgerson W.S. Multidimensional scaling: I Theory and method // Psychometrika. – 1952. – V. 17. – № 3. – P. 401–419.
13. Терехина А.Ю. Многомерное шкалирование в психологии. – Режим доступа: http://psyfactor.org/lib/terehina.html
14. Спектроколориметр "ТКА-ВД". – Режим доступа: http://www.tka.spb.ru/produkt/tka_vd.html
15. DCI Digital Cinema System Specifications V1.0, Digital Cinema Initiatives 2005, доступ с www.dcimovies.com
16. Рабкин Е.Б. Полихроматические таблицы для исследования цветоощущения. – М., 1971. – 244 с.
17. Юстова Е.Н., Алексеева К.А., Волков В.В., и др. Пороговые таблицы для исследования цветового зрения. – М.: Вида, 1993.

ПОСТРОЕНИЕ АНАЛИЗАТОРА ИЗОБРАЖЕНИЙ ГРАНУЛОМЕТРИЧЕСКОГО ТИПА

А.Е. Никитин (Ярославский государственный университет им. П.Г. Демидова)
Научный руководитель – к.т.н., доцент В.В. Хрящев
(Ярославский государственный университет им. П.Г. Демидова)

Статья посвящена проблеме построения анализатора изображений, предназначенного для обработки гранулометрических изображений, встречающихся в медицине и материаловедении. Описывается разработка и тестирование опытного экземпляра программно-аппаратного комплекса, в состав которого входят микроскоп, цифровая фотокамера и персональный компьютер.

Введение

Гранулометрия – совокупность научно-технических знаний, связанных с изучением распределения частиц на сложных изображениях. Как правило, такие изображения представляют собой распределение замкнутых областей на некотором фоне либо имеют вид доменной структуры. Наибольшее количество задач, связанных с гранулометрическим анализом изображений, встречается в медицине, смежных с ней биологии и генетике, а также материаловедении, активно используемом в различных отраслях промышленности, имеющих дело с применением или производством порошкообразных или поликристаллических веществ. В медико-биологических исследованиях гранулометрическая обработка используется во время процедур анализа крови, плоидометрии, расчета ядерно-цитоплазматического отношения, гистосчета клеток, эритроцитометрии, подсчета числа тромбоцитов и т.п. В материаловедении гранулометрия применяется для анализа порошковых препаратов (алмазов, корундов, карбидов кремния, глинозема, нитридов бора, стеклянных шариков, люминофоров), определения размеров и формы проекций объектов в связанных материалах (металлах, керамике, горных породах).

Гранулометрический анализ, как правило, является достаточно трудоемким процессом и при проведении вручную требует ощутимых временных затрат. Большой практический интерес представляет разработка программно-аппаратного комплекса, который позволит систематизировать процесс получения и обработки изображений гранулометрического типа. Такие комплексы, называемые анализаторами изображений, производятся некоторыми зарубежными компаниями, работающими в области микроскопии. Однако их разработки носят узкоспециализированный характер и не применимы для решения схожих задач в различных предметных областях [1]. Кроме того, их стоимость довольно высока и зачастую оказывается неприемлемой для российского потребителя, особенно для учреждений науки и образования.

Целью проекта является разработка опытного образца анализатора изображений гранулометрического типа, представляющего собой специализированный программно-аппаратный комплекс для работы с микрообъектами в области видимого света. Причем проектируемая система должна обладать свойствами универсальности и низкой себестоимостью.

Теория построения анализатора изображений

Анализатор изображений для световой микроскопии, как правило, имеет модульное строение и состоит из системы ввода изображений (микроскоп, камера), компьютера (ПК) и принтера для документирования полученных результатов на бумажном носителе. Модульность строения обеспечивает гибкость систем анализа изображений, которая с легкостью может быть адаптирована к потребностям пользователя в зависимости от задачи, которую он собирается решать с помощью анализатора. Адаптация может

проводиться не только модернизацией или заменой модуля, но также и установкой специализированного программного обеспечения по обработке изображений.

Система ввода изображений состоит из светового микроскопа и камеры. Она определяет качество получаемого на экране компьютера изображения. Выбор системы ввода – один из важнейших этапов при комплектации анализатора изображений. Применение микроскопа того или иного типа, а также дополнение его устройствами для получения различных методов исследования и контрастирования определяется физико-химическими свойствами объекта наблюдения. Роль камеры заключается в том, чтобы зафиксировать полученное с микроскопа изображение без потери разрешения и искажения цвета. Таким образом, можно проследить взаимосвязь «объект наблюдения – микроскоп – камера». В этой цепочке определяющим является объект исследования и вид анализа, от которых, собственно, и зависит выбор системы ввода.

Для обеспечения работы микроскопа в составе анализатора изображений класс микроскопа должен быть не ниже рабочего, а для некоторых специализированных методик, связанных с цветом объекта и его разрешением, – не ниже лабораторного [1]. Важным условием при сборке являются правильные подбор и установка камеры на микроскоп. Микроскоп должен быть снабжен бинокулярной насадкой с фото-видеовыходом или специальным выходом на штативе, на который с помощью специального видеоадаптера монтируется камера или цифровой фотоаппарат. Основное внимание при выборе цифровой фотокамеры, которую планируется использовать в анализаторе изображений, необходимо уделить следующим параметрам: разрешающей способности, динамическому диапазону, отношению сигнал/шум и качеству цветопередачи. Адаптер должен быть рассчитан таким образом, чтобы обеспечить передачу на матрицу камеры максимальную часть анализируемого поля.

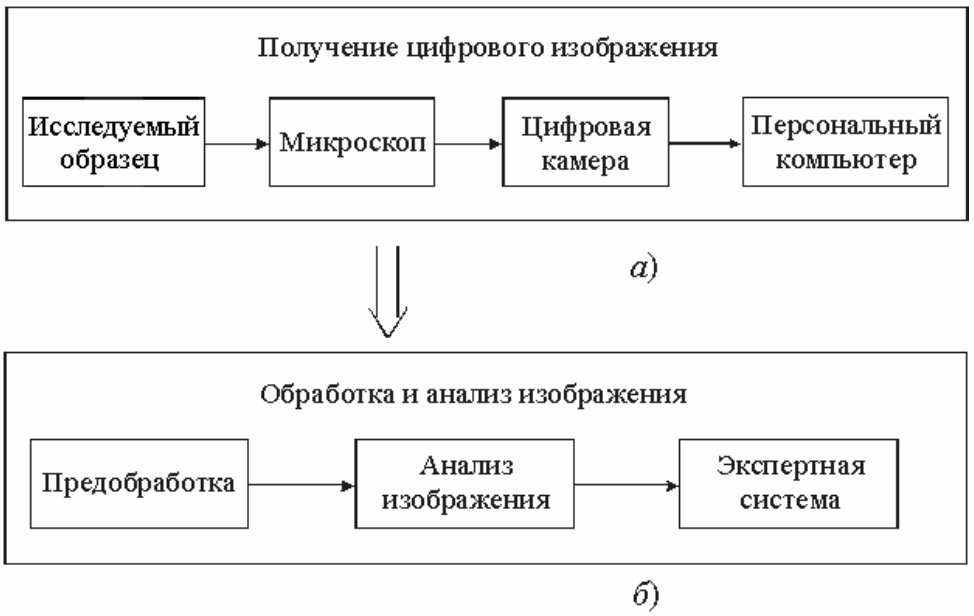


Рис. 1. Общая схема анализатора изображений

Программное обеспечение, входящее в состав анализатора изображений, определяет технологию его работы. В зависимости от специализации анализатора набор имеющихся функций ПО может сильно варьироваться, но в самом общем виде этот набор включает в себя блоки предобработки и анализа (распознавания объектов), сопряженные с экспертной системой. Общая схема программно-аппаратного анализатора представлена на рис. 1.

Аппаратная часть

Для получения достоверных гранулометрических изображений необходимо использовать качественное оборудование, которое является широко доступным и имеет относительно невысокую совокупную стоимость. На основе проведенного исследования различного оборудования, применяемого для гранулометрической обработки, было принято решение использовать в качестве системы ввода разрабатываемого анализатора набор, состоящий из микроскопа МБС-10 (рис. 2) и цифровой фотокамеры Olympus Camedia C-7070 (рис. 3). Бинокулярный стереоскопический микроскоп МБС-10 с увеличением до 100 крат предназначен для наблюдения миниатюрных деталей объемных предметов как при искусственном, так и при естественном освещении в отраженном и проходящем свете. Через специальный переходник окуляр микроскопа соединяется с объективом фотоаппарата Olympus Camedia C-7070 WIDE ZOOM, который благодаря высокому разрешению снимков (3072×2304 пикселей) позволяет передать даже мельчайшие детали объекта. Исследуемый образец (медицинский препарат, алмазный порошок и т.п.) помещается в микроскоп, оператор делает снимок, который через интерфейс USB поступает в компьютер и дальше обрабатывается программной частью комплекса.

Рис. 2. Микроскоп МБС-10

Рис. 3. Фотоаппарат Olympus C-7070

Программная часть

Процесс гранулометрической обработки и классификации полученных снимков автоматизированным комплексом ведется с помощью программной части, работу которой можно разделить на следующие основные стадии: предобработка исходного изображения, анализ изображения, принятие решения экспертной системой.

Предобработка включает в себя удаление шума и различных искажений, которые в ряде случаев возникают при регистрации снимков. Для восстановления изображения предлагается использовать линейную и медианную цифровую фильтрацию, выравнивание гистограммы, настройку яркости и контрастности [2]. В результате первого этапа работы алгоритма улучшенное изображение готово к анализу.

На этапе анализа производится выделение кластеров на изображении [3]. Процедура выделения кластеров включает операции детектирования краев объектов, утолщение контуров, заполнение объектов. Предлагается перевести исходное изображение в бинарное представление с помощью фильтра Собеля [4] и затем обработать с помощью морфологических операций [5], среди которых особо можно выделить алгоритм морфологической скелетонизации, применяемый для разделения «слипшихся» объектов.

Алгоритм этого преобразования основан на базовых морфологических операциях эрозии и дилатации, являясь идейным наследником методики сегментации по водоразделам [2].

На первом этапе работы алгоритма многократно проводится эрозия изображения до тех пор, пока объект не выродится в точку или закончится *N*-ая итерация эрозии (число *N* задается до начала работы процедуры скелетонизации). Это позволяет достигнуть максимально возможной степени разделения объектов.

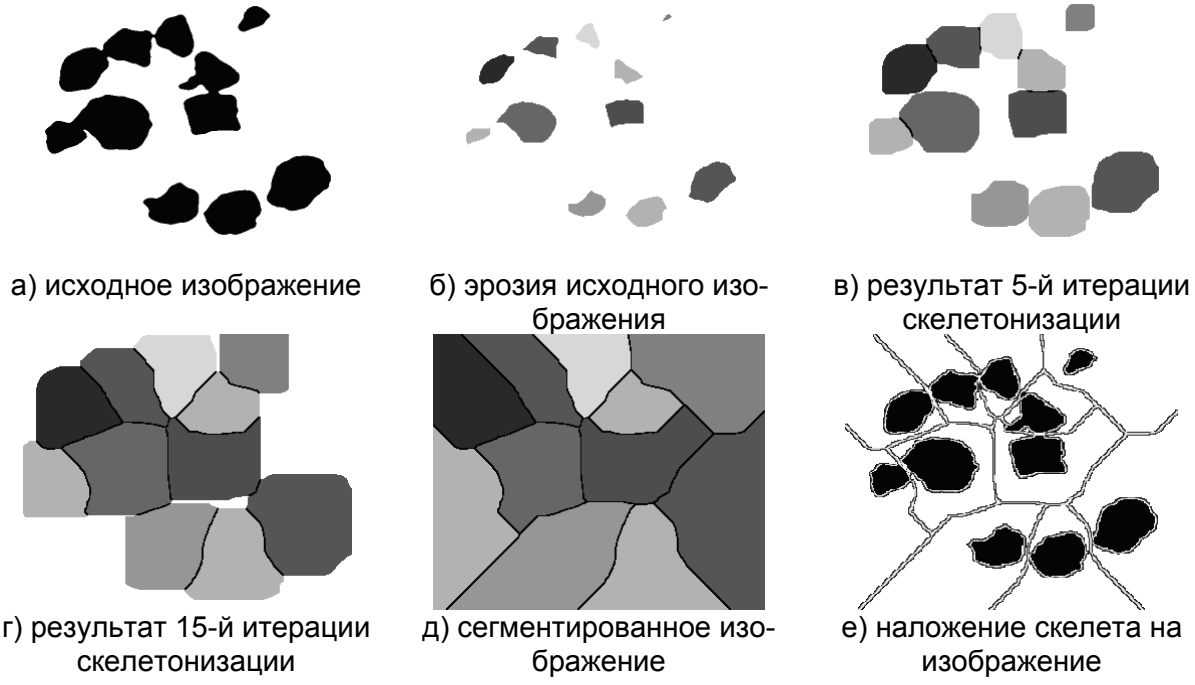
На второй стадии множества точек, принадлежащих различным объектам, обозначаем через $\boldsymbol{M_1, M_2, \ldots, M_n}$. Распределение точек по множествам происходит с помощью алгоритма заливки. Затем к множествам $\boldsymbol{M_1, M_2}$ и т.д. многократно применяем операцию дилатации, которая приводит к постепенному расширению исходных множеств. На некоторой *k*-ой итерации дилатации может возникнуть пересечение постоянно растущих множеств $\boldsymbol{M_x}$ и $\boldsymbol{M_y}$, где $x, y \in (1,..,n)$. Это пересечение мы будем считать частью скелета изображения, и обозначать через $\boldsymbol{G_k}$. Объединение всех множеств $\boldsymbol{G_k}$ даст полный скелет изображения. Таким образом, скелет $\boldsymbol{G}$ можно определить по формуле

$$G = \bigcup_{k=0}^{N} G_k \quad G_k = M_{x_k} \cap M_{y_k} \quad x, y \in (1,..,n),$$

где

$$M_{x_k} = M_{x_{k-1}} \oplus B .$$

Скелетонизация прекрасно сегментирует изображение, разделяя его на ряд связанных областей. Получившийся в результате преобразования скелет накладывается на исходное бинарное изображение. В тех местах, где скелет пересекает объекты, ставится граница между слипшимися предметами. На рис. 4 показаны основные этапы применения морфологической скелетонизации к гранулометрическому изображению.



а) исходное изображение; б) эрозия исходного изображения; в) результат 5-й итерации скелетонизации; г) результат 15-й итерации скелетонизации; д) сегментированное изображение; е) наложение скелета на изображение

Рис. 4. Этапы алгоритма морфологической скелетонизации

После того, как объекты выделены, часть их можно отсеять, произведя измерительные процедуры, описывающие форму, размеры, оптические свойства. В результате мы можем оставить лишь изучаемые объекты, которые в дальнейшем можно классифицировать, т.е. отнести объект к нужному классу.

Перед измерением размеров объект ориентируется подобающим образом. После проведения операции поворота не представляет трудности классифицировать выделенные объекты в программе с помощью дескрипторов площади, периметра, компактности, диаметра, большой и малой оси, а также цвета. Пользователю предоставлена возможность определять собственные дескрипторы на основе уже имеющихся базовых. Например, в соответствии с ГОСТ 9206-80 – стандартом качества алмазных порошков – размер алмаза определяется как полусумма длины и ширины прямоугольника, условно описанного вокруг проекции зерна так, что большая сторона прямоугольника соответствует наибольшей длине проекции зерна [6]. Такого рода дескриптор может быть записан через сумму большой и малой оси объекта, деленную на два.

В зависимости от наперед заданных оператором установок на предмет отнесения объекта к тому или иному классу может происходить отсеивание лишних предметов, не полностью удовлетворяющих требованиям по своим дескрипторам.

В подавляющем большинстве гранулометрические задачи требуют не только получения данных о распределении объектов по некоторым параметрам, но также и экспертного заключения о качестве рассматриваемого образца. На последнем этапе анализа изображения предлагается использовать нейросетевую экспертную систему, которая обеспечивает адекватную оценку образца и выносит решение о качестве. Ее разработка пока не завершена, но в ближайшей перспективе планируется.

Тестирование разработанного анализатора

В ходе работы был создан пробный образец программно-аппаратного анализатора изображений гранулометрического типа (рис. 5), на котором проверяются реализованные в ходе работы методики цифровой обработки, тестируются внедряемые программные модули.

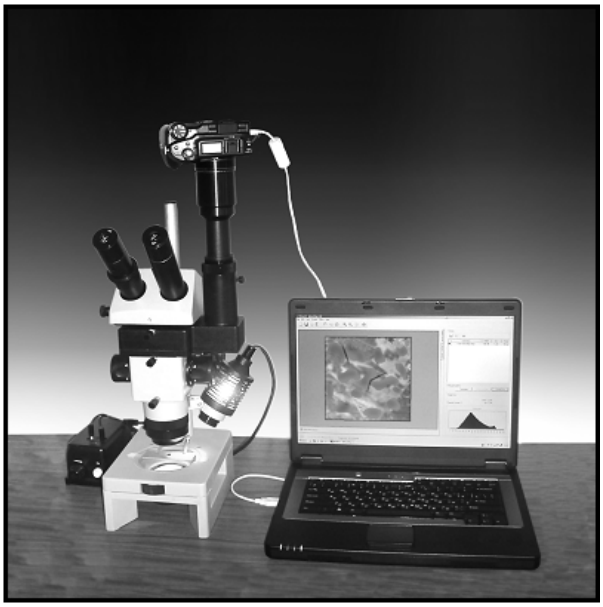


Рис. 5. Опытный образец анализатора изображений гранулометрического типа

Тестирование системы на распознавание объектов проводилось для набора характерных изображений гранулометрического типа, в качестве которых были выбраны снимки микропорошков и субмикропорошков алмазов. Эти наборы содержат равное количество удовлетворяющих и не удовлетворяющих ГОСТ изображений. Неправильными считаются ошибочные ответы системы и неопределенные состояния (когда экспертная система затрудняется дать точный ответ). Тестирование проводилось не только

для законченной системы, но и для различных ее модификаций. При этом модифицированная система отличается от исходной изменением всего в одном блоке. В таблице приведены результаты тестирования (доля правильной оценки экспертной системы) в зависимости от используемых компонентов комплекса. В графе «модификация» курсивом выделены элементы, позволяющие более точно выносить экспертную оценку.

Использование блоков программного пакета MicroVision	Микропорошки	Субмикропорошки
Предобработка		
Без блока предварительной фильтрации	79	75
С блоком предварительной фильтрации	90	92
Морфологический анализ		
Без процедуры морфологической скелотонизации	85	85
С процедурой морфологической скелотонизации	94	96
Распознавание объектов		
Без проведения отсеивания объектов по дескрипторам	91	90
С проведением отсеивания объектов по дескрипторам	96	98

Таблица. Доля правильной оценки экспертной системы, %

Результаты проведенной процедуры тестирования автоматизированной экспертной системы показывают, что:

1. предварительная обработка изображения (обработка медианным фильтром, сглаживание линейным фильтром, регулировка яркости и контрастности методом модификации гистограмм) существенно влияет на точность распознавания объекта;
2. доля правильной оценки экспертной системы для микропорошков увеличивается на 11%, а для субмикропорошков на 17%;
3. введение алгоритма морфологической скелетонизации в систему увеличивает долю правильной оценки системы на 9–11%;
4. проведение процедуры отсеивания объектов по дескрипторам позволяет устранить 5–8% ошибок, допущенных на предыдущей стадии.

Заключение

Разработанный программно-аппаратный комплекс проведения автоматизированного гранулометрического анализа изображений позволяет значительно сократить вре-

мя и трудозатраты, необходимые для решения гранулометрической задачи, а также снизить вероятность ошибки при экспертной оценке, так как данный алгоритм свободен от человеческого фактора. Следует отметить, что алгоритмическое ядро комплекса является универсальным, т.е. при соответствующей настройке и переобучении нейросетевой экспертной системы его можно использовать для решения схожих гранулометрических задач, возникающих в различных сферах деятельности.

Литература

1. Пантелеев В.Г., Клыкова Е.И., Егорова О.В. Компьютерная микроскопия. – М.: Техносфера, 2005. – 304 с.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. – 1072 с.
3. Андреев С.Е. Выделение сегментов на изображении // Докл. 6-й междунар. конф. «Цифровая обработка сигналов и ее применение», Москва, 2004. Т.2. С. 100–103.
4. Методы компьютерной обработки изображений. Учебное пособие (2-е издание) / Под ред. Сойфера В.А. – М.: Физматлит, 2004. – 317 с.
5. Soille P. Morphological Image Analysis: Principles and Applications, 2nd ed. / Springer-Verlag. – NY, 2003. – 391 p.
6. Хрящев В.В., Апальков И.В., Соколенко Е.А., Куйкин Д.К. Обработка изображений алмазных порошков в Matlab Image Processing Toolbox // Тр. всерос. науч. конф. «Проектирование научных и инженерных приложений в среде MATLAB», Москва, 2004. С. 1671–1675.

МЕХАНИЗМ РЕАЛИЗАЦИИ ШАБЛОНОВ ТЕСТОВЫХ ЗАДАНИЙ В СИСТЕМЕ ACADEMICNT

А.В. Ольшевская
Научный руководитель – к.т.н., доцент А.В. Лямин

В данной статье отражены основные достоинства технологии создания шаблонов тестовых заданий. Рассмотрены наиболее заинтересовавшие аналоги. Показан механизм реализации шаблонов тестовых заданий в СДО AcademicNT. Рассмотрено DOM-представление XML-описания шаблонов. Представлен пример тестового задания, реализованного с помощью новой технологии.

Введение

В связи с повсеместным распространением систем дистанционного обучения (СДО) происходит все большая информатизация процесса обучения. На базе центра дистанционного обучения (ЦДО) Санкт-Петербургского государственного университета информационных технологий, механики и оптики (СПбГУ ИТМО) создана система AcademicNT, которая является сетевой информационной средой для организации и проведения учебного процесса средствами современных дистанционных технологий [1, 2]. Она способствует внедрению в учебный процесс современных компьютерных технологий. Основной целью использования данной системы является повышение качества обучения посредством активизации самостоятельной работы студентов.

Для проверки уровня подготовки обучаемых в системе AcademicNT предусмотрены виртуальные лаборатории, электронные тесты и практикумы [1–3]. Однако наиболее востребованными среди преподавателей являются электронные тесты, которые используются не только для проведения аттестаций и репетиций, но и для построения адаптивных обучающих диалогов, наиболее эффективного инструмента электронного обучения. Создание адаптивных обучающих диалогов является трудоемким процессом, занимающим порой длительное время.

Интенсивное развитие СДО приводит к появлению новых разработок, применение которых должно упростить процесс создания и расширить функциональные возможности электронных тестов. Для уменьшения трудозатрат, минимизации времени разработки, а также повышения точности педагогических измерений в СДО начали применять новую технологию создания тестовых заданий, в основе которой лежат понятия шаблона и обратной связи. Понятие шаблона и обратной связи, их формальное описание и примеры использования для построения адаптивных диалогов можно найти на сайте всемирного образовательного консорциума IMS (http://www.imsglobal.org/).

Основной целью данной работы является создание эффективного механизма реализации шаблонов тестовых заданий в системе AcademicNT. В работе рассматривались следующее вопросы:

(1) механизмы реализации шаблонов в СДО AcademicNT;
(2) DOM-представление XML-описания шаблонов;
(3) примеры разработанных шаблонов для тестового задания, реализованного в СДО AcademicNT.

Механизмы реализации шаблонов в СДО AcademicNT

Для начала необходимо определиться с основной терминологией, используемой в данной статье.

Класс – это логическая конструкция, определяющая форму и природу объекта. Он создает новый тип данных. После определения новый тип можно использовать для создания (инициализации) объектов этого типа, т.е. экземпляров данного класса.

Шаблон определяет класс тестовых заданий. Конкретное задание является экземпляром этого класса, инициализирующимся при передаче тестового задания испытуемому.

Анализатор – логическое выражение, предназначенное для анализа ответа пользователя.

Обратная связь – реакция системы на ответ пользователя.

Адаптивный диалог – диалог, траектория которого определяется в процессе взаимодействия пользователя с системой.

Построение шаблонов в СДО AcademicNT происходит с использованием переменных. Такими переменными можно заменять элементы всех структурных единиц тестового задания, будь то вопрос, ответ (варианты ответов) или вложения (например, рисунки). Механизм реализации шаблонов в СДО AcademicNT можно проиллюстрировать алгоритмом, представленным на рис. 1.

Рис. 1. Алгоритм формирования задания и обработки ответа

На первом этапе происходит объявление и инициализация переменных, каждая из которых должна обладать уникальным идентификатором, иметь логическое имя. При инициализации переменной присваивается определенное значение. На этапе формирования задания происходит добавление переменных к структурным единицам тестового задания. На этапе предварительной обработки ответа, полученного от испытуемого, происходит разбор ответа пользователя. Этап анализа ответа включает в себя проверку определенных условий. В зависимости от условия происходит изменение значений переменных, после чего возможен либо выход из тест-кадра, либо переход на этап формирования нового задания (обратная связь).

Проверка условий правильности ответа происходит по схеме, представленной на рис. 2. Если Условие 1.1, Условие 1.2 и Условие 1.3 выполняются, то происходит вычисление переменных с последующим выходом из проверки. Если хотя бы одно из этих условий оказывается неверным, то осуществляется переход к другому блоку проверки, в котором происходит аналогичный процесс.

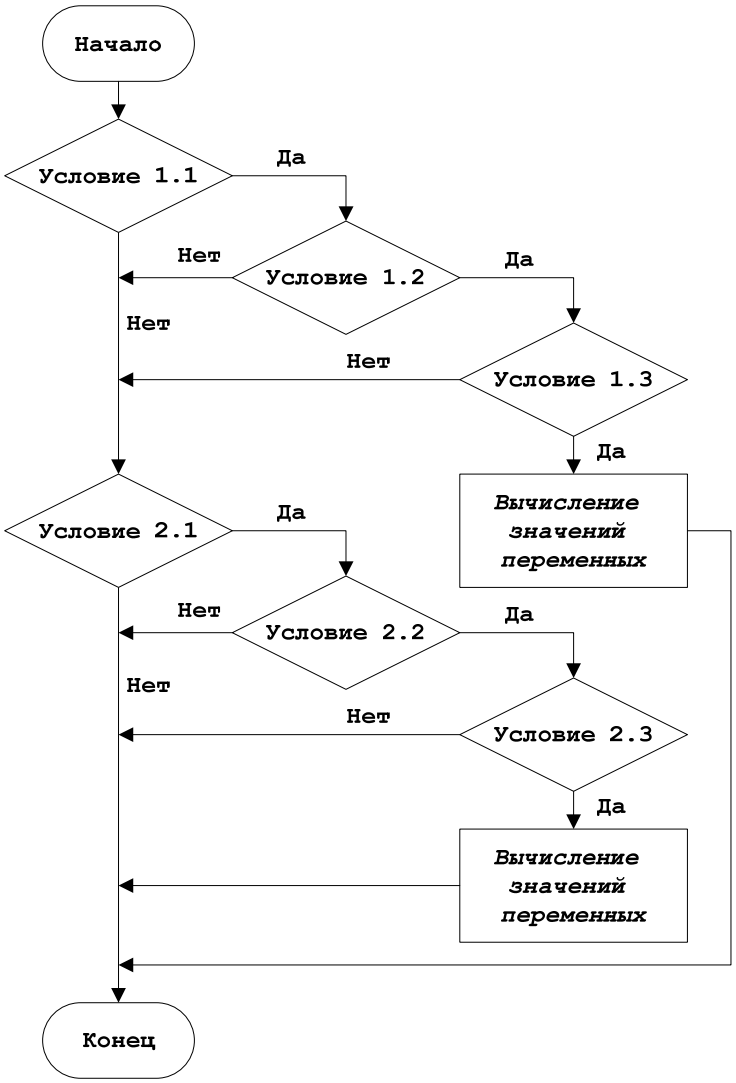


Рис. 2. Блок-схема анализатора ответа

Использование технологии построения шаблонов обеспечивает:

(1) индивидуальный подход к каждому обучающемуся;

(2) высокую надежность педагогических измерений;

(3) уменьшение трудозатрат разработчиков контрольно-измерительных материалов, что определяет актуальность проведения исследований по данной теме.

DOM-представление XML-описания шаблонов

Для большей наглядности и иллюстрации иерархии объектов было разработано представление DOM (Document Object Model) XML-описания шаблонов, отдельные фрагменты которого представлены ниже.

Элемент <TestFrameVariables>, представленный на рис. 3, описывает переменные кадра. Он включает в себя один или несколько вложенных элементов <TestFrameVariable>, которые имеют атрибуты: TestFrameVariableID – внутренний числовой иденти-

фикатор переменной; VariableID – логическое имя переменной; Name – описание переменной. Элемент <TestFrameVariable> обязательно включает в себя один из элементов <Single> – простая переменная или <Multiple> – составная переменная. Также он содержит элемент <Comment> – комментарий, являющийся необязательным для использования.

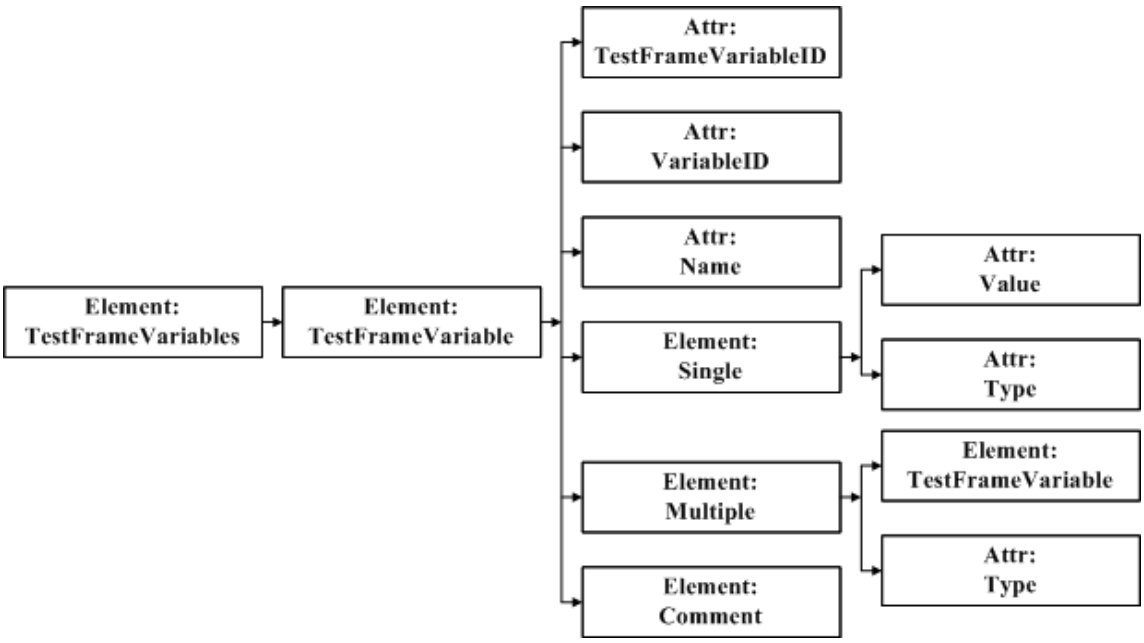


Рис. 3. DOM-представление фрагмента, описывающего переменные

Элемент <Single> имеет атрибуты Value – значение переменной; Type – тип переменной, который может быть либо числовым (значение по умолчанию), либо строковым. Элемент <Multiple> включает в себя вложенный элемент <TestFrameVariable> – переменная кадра и обязательный атрибут Type, принимающий значение, используемое по умолчанию, Array (Массив) или значение Record (Запись).

Элемент <TestResponseProcessing>, представленный на рис. 4, описывает обработку результатов тестового задания. Он содержит произвольное количество элементов <TestResponseInit>, в которых происходит дополнительная инициализация переменных. Данный элемент не содержит данных, но имеет три обязательных атрибута: InitID – идентификатор инициализации, VariableID – логическое имя инициализируемой переменной, Value – присваиваемое значение.

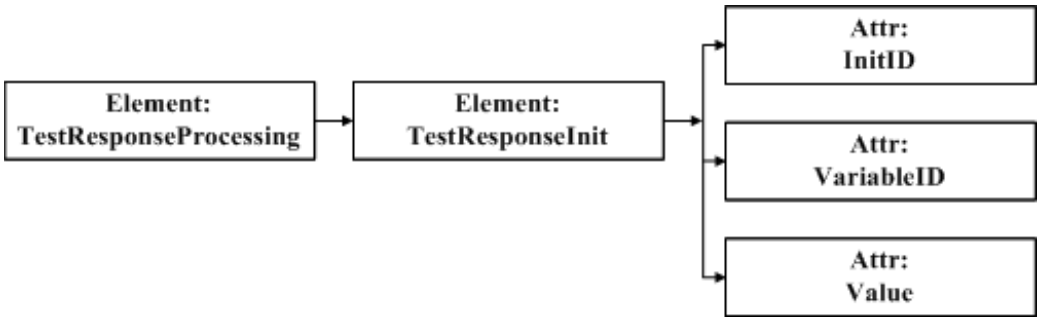


Рис. 4. DOM-представление фрагмента, описывающего инициализацию переменных

Элемент <TestResponseCondition>, представленный на рис. 5, также является вложенным в элемент <TestResponseProcessing>. Он может встречаться один или более

раз. Элемент <TestResponseCondition> содержит произвольное количество вложенных элементов:

(1) <Command> – команда, инициализировавшая передачу реакции пользователя на сервер, имеет атрибут CommandID, принимающий одно из следующих значений: "Check" – проверка введенного пользователем ответа; "TimeOut" – время, отведенное на тест, вышло; "Forward" – переход к следующему кадру; "Backward" – переход к предыдущему кадру; "Skip" – пропуск кадра; "Refuse" – нет ответа; "Help" – вызов помощи;
(2) <TestResponseMatch> – элемент, описывающий сравнение переменной с эталонным значением;
(3) <TestResponseOutcome> – элемент, описывающий изменение значения переменной после выхода из кадра при выполнении условия.

Также он включает в себя следующие атрибуты:

(1) TestResponseConditionID – идентификатор условия;
(2) Output определяет правильность ответа (0 – ответ неверный, 1 – ответ верный);
(3) Message – сообщение, которое выдается после получения правильного ответа от пользователя;
(4) NextTestFrame – тест-кадр, на который осуществляется переход после выполнения условия;
(5) Time – время, которое отводится на проверку условия (если указано значение –1, то данный атрибут не указан);
(6) TimeScale – временная шкала, принимающая одно из следующих значений: Day"; "Hour"; "Minute"; "Second";
(7) TryNumber показывает количество попыток в обратной связи (если указано значение –1, то данный атрибут не указан).

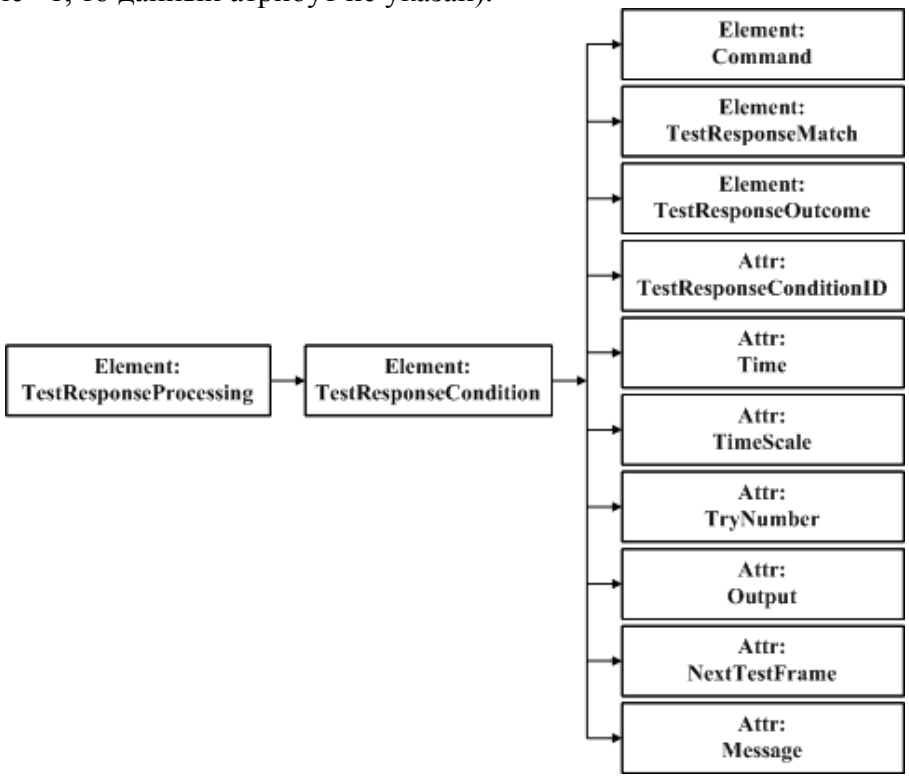


Рис. 5. DOM-представление фрагмента, описывающего условия проверки ответа

Элемент <TestResponseMatch>, представленный на рис. 6, описывает сравнение значения переменной, полученного после ее инициализации, с некоторым эталонным значением. Он включает в себя один из следующих вариантов сочетаний элементов, содержащих символьную информацию:

(1) <Pattern> – элемент, описывающий сравнение по шаблону (имеет атрибут OrdinaryChars, устанавливающий знаки-разделители);
(2) <Value> – элемент, описывающий сравнение по определенному значению (употребляется с необязательным элементом <Error>);
(3) <Min> и <Max> – элементы, описывающие сравнение переменной с минимальным и максимальным допустимыми значениями;
(4) <Min> – элемент, описывающий сравнение по минимальному допустимому значению;
(5) <Max> – элемент, описывающий сравнение по максимальному допустимому значению,

а также два атрибута:

(1) VariableID – логическое имя проверяемой переменной;
(2) IsTrue – признак желаемого результата проверки, принимающий значение "Yes" (значение по умолчанию), если при успешной проверке происходит выполнение условия или "No", если проверка не прошла, но условие все равно выполняется.

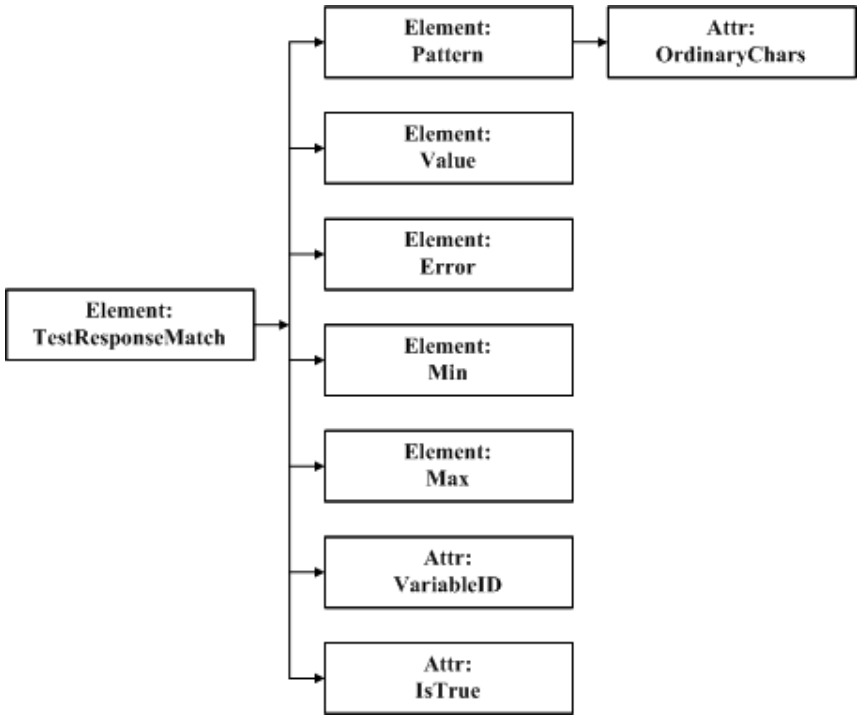


Рис. 6. DOM-представление фрагмента, описывающего сравнения переменных

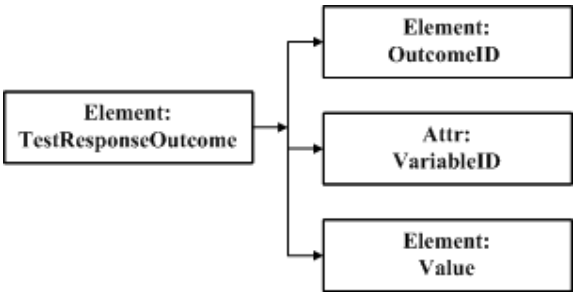


Рис. 7. DOM-представление фрагмента, описывающего переменную после проверки условий

Элемент <TestResponseOutcome>, представленный на рис. 7, задает значение переменной после выхода из кадра при выполнении условия. Включает в себя обязательные для использования атрибуты:

(1) OutcomeID – идентификатор;
(2) VariableID – логическое имя переменной;
(3) Value – присваиваемое значение.

Примеры разработанных шаблонов для тестового задания, реализованного в СДО AcademicNT

Использование шаблонов в системе AcademicNT можно рассмотреть на примере XML-описания тестового задания на вычисление корней квадратного уравнения:

```
<TestFrameVariables>
   <TestFrameVariable
      TestFrameVariableID="1"
      VariableID="r1"
      Name="Корень"
   >
      <Single
         Value="round(dbms_random.value*10)"
      />
   </TestFrameVariable>
   <TestFrameVariable
      TestFrameVariableID="2"
      VariableID="r2"
      Name="Корень"
   >
      <Single
         Value="round(dbms_random.value*10)+r1+1"
      />
   </TestFrameVariable>
   <TestFrameVariable
      TestFrameVariableID="3"
      VariableID="a1"
      Name="Коэффициент"
   >
      <Single
         Value="-(r1 + r2)"
      />
   </TestFrameVariable>
   <TestFrameVariable
      TestFrameVariableID="4"
      VariableID="a0"
      Name="Коэффициент"
   >
      <Single
         Value="r1 * r2"
      />
   </TestFrameVariable>
   <TestFrameVariable
      TestFrameVariableID="5"
      VariableID="num1"
      Name="Ответ пользователя"
   >
```

```
        <Single
            Value="0"
        />
    </TestFrameVariable>
    <TestFrameVariable
        TestFrameVariableID="6"
        VariableID="num2"
        Name="Ответ пользователя"
    >
        <Single
            Value="0"
        />
    </TestFrameVariable>
    <TestFrameVariable
        TestFrameVariableID="7"
        VariableID="dA1"
        Name="Погрешность"
    >
        <Single
            Value="0"
        />
    </TestFrameVariable>
    <TestFrameVariable
        TestFrameVariableID="8"
        VariableID="dA0"
        Name="Погрешность"
    >
        <Single
            Value="0"
        />
    </TestFrameVariable>
</TestFrameVariables>
```

В элементе <TestFrameVariables> определяются и инициализируются восемь переменных. Переменные с логическими именами r1 и r2 являются корнями квадратного уравнения. Их генерация происходит на основе псевдослучайного алгоритма. При генерации чисел для переменных r1 и r2 главным аспектом является их неравенство между собой. В случае совпадения этих двух значений квадратное уравнение будет иметь только один действительный корень, что значительно упростит задачу испытуемого. Переменные a1 и a0 – это коэффициенты квадратного уравнения, вычисляемые с использованием r1 и r2. Переменные num1 и num2, dA0 и dA1 используются во время анализа ответа.

```
<Data>
    <!--
    После записи ответа нажимайте кнопку "Ответ готов". Для окончания
    сеанса тестирования кнопку "Завершить".<br/>
    ВЫЧИСЛИТЕ КОРНИ КВАДРАТНОГО УРАВНЕНИЯ:<br/>
    x<sup>2</sup>+<TestFrameVariable VariableID="a1"/>x+<TestFrameVariable
    VariableID="a0"/>=0.
    -->
</Data>
```

В объекте <Data> расположен вопрос тестового задания, включающий в себя краткую инструкцию и само задание, в котором вместо коэффициентов квадратного уравнения используются переменные с логическими именами a1 и a0.

```
<Open>
   <String
      Format="x1 = DD, x2 = DD"
   />
</Open>
```

Тестовое задание, рассматриваемое в качестве примера, относится к открытому типу. Для его описания необходимо было применить элемент Open, который содержит информацию об ответе студента. В данном случае ответ студента предполагается получить в формате строки, поэтому использован вложенный объект String, включающий необязательный атрибут Format – маска, предоставляемая испытуемому в текстовом поле задания, пустые фрагменты которой ему необходимо заполнить, что делает невозможным пропуск любого из введенных ответов.

```
<TestResponseProcessing>
   <TestResponseInit
      InitID="1"
      VariableID="num1"
      Value="to_number(substr(Response,6,2))"
   />
   <TestResponseInit
      InitID="2"
      VariableID="num2"
      Value="to_number(substr(Response,15,2))"
   />
   <TestResponseInit
      InitID="3"
      VariableID="dA1"
      Value="- (num1 + num2) - a1"
   />
   <TestResponseInit
      InitID="4"
      VariableID="dA0"
      Value="num1 * num2 - a0"
   />
   <TestResponseCondition
      TestResponseConditionID="1"
      Output="1"
   >
      <Command
         CommandID="Check"
      />
      <TestResponseMatch
         VariableID="dA1"
      >
         <Value>
            <!-- 0 -->
         </Value>
      </TestResponseMatch>
      <TestResponseMatch
```

```
            VariableID="dA0"
        >
            <Value>
                <!-- 0 -->
            </Value>
        </TestResponseMatch>
    </TestResponseCondition>
    <TestResponseCondition
        TestResponseConditionID="2"
        Output="0"
    >
        <Command
            CommandID="Check"
        />
        <TestResponseMatch
            VariableID="Response"
        >
            <Pattern>
                <!-- <*> -->
            </Pattern>
        </TestResponseMatch>
    </TestResponseCondition>
</TestResponseProcessing>
```

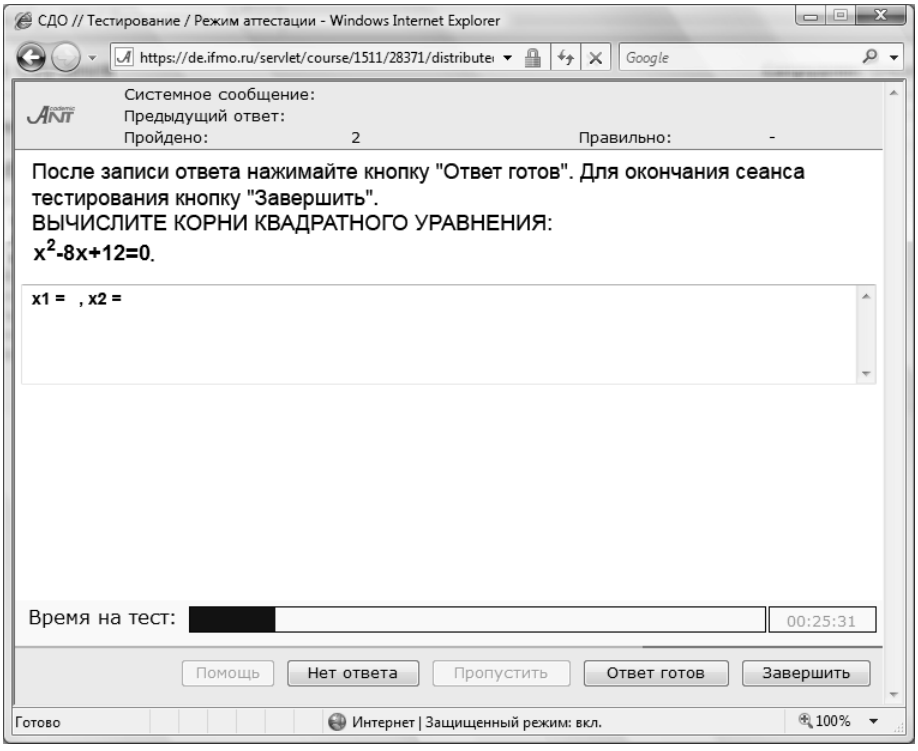


Рис. 8. Тестовое задание на вычисление корней квадратного уравнения

В объекте <TestResponseProcessing> описывается процесс обработки результатов теста. Инициализация переменных с логическими именами: num1, num2, dA0, dA1 происходит в объектах <TestResponseInit>. На этом этапе каждой переменной присваивается новое значение. Response – фиксированная переменная, в которой хранится ответ пользователя. При разборе ответа пользователя находятся значения переменных num1 и

num2, которые используются в дальнейшем для вычисления погрешностей ответов пользователя, переменных dA0, dA1. После проверки условия в атрибут Output заносится значение 0, если ответ неверный, и 1, если ответ верный. В объекте <TestResponseMatch> происходит сравнение переменной со значением, полученным из объекта <TestResponseInit>. В первом блоке проверки происходит сравнение переменных dA0 и dA1 с их эталонными значениями, которые указаны в соответствующих им атрибутах Value, а во втором используется проверка по шаблону <Pattern>, пропускающая в данном случае любой ответ. Подобная проверка может понадобиться, например, если правильный ответ так и не получен от испытуемого, а переход на следующий кадр или окончание теста необходимо осуществить.

Тестовое задание, фрагменты XML-описания которого были рассмотрены выше, представлено на рис. 8.

Заключение

В работе были рассмотрены механизмы реализации шаблонов в СДО AcademicNT; приведены и описаны DOM-представления XML-описания шаблонов, на основе которого были продемонстрированы примеры разработанных шаблонов для тестового задания, реализованного в СДО AcademicNT. Работа над темой продолжается, и новые результаты проделанной работы в скором будущем будут представлены.

Литература

1. Васильев В.Н., Лямин А.В., Чежин М.С. Система дистанционного обучения второго поколения // Научно-технический вестник СПбГУ ИТМО. – 2007. – Выпуск 45. Информационные технологии. – С.148–157.
2. Лямин А.В., Чежин М.С. Система дистанционного обучения СПбГУ ИТМО // Труды II Международной конференции «Информационные системы для HRM и обучения». – Санкт-Петербург, 2005. – С. 80–83.
3. Васильев В.Н., Лисицына Л.С., Лямин А.В. Технология проведения ЕГЭ по информатике в компьютерной форме // Научно-технический вестник СПбГУ ИТМО. – 2007. – Выпуск 45. Информационные технологии. – С. 126–143.

МОДУЛЬНЫЙ ПОДХОД К ПОСТРОЕНИЮ СИСТЕМ УПРАВЛЕНИЯ ТРЕБОВАНИЯМИ

А.Ю. Зырянов
Научный руководитель – к.т.н., доцент Н.Ф. Гусарова

В статье предложен модульный подход к построению инструментальных средств управления требованиями, ориентированный на работу в небольших командах разработчиков.

Введение

Аналитические исследования и обзоры в области разработки программного обеспечения показывают, что большая часть проектов завершается с опозданиями, превышениями бюджетов, в них не реализуются в полном объеме или аннулируются до завершения требуемые функции. Так, по данным аналитических исследований компании The Standish Group, за последние десять лет не более 30% программных проектов завершаются в положенный срок и в рамках бюджета [1]. При этом проблемы более чем трети программных проектов непосредственно связаны с работой с требованиями. Сюда относятся:

- недостаток исходной информации от клиента;
- неполные требования и спецификации;
- изменение требований и спецификаций.

В цикле разработки программного обеспечения этап выработки требований предшествует всем другим этапам, следовательно, внесенные на этом этапе дефекты более сильно сказываются на качестве разработки. Так, плохие требования могут стать причиной неверной оценки объема необходимых работ и, как следствие, нереалистичных планов проекта и завышенных ожиданий заказчика.

Основным же следствием проблем с требованиями является переделка того, что, как казалось, уже готово. На это расходуется от 30 до 50% общего бюджета разработки, а ошибка в требованиях стоят от 70 до 85% стоимости переделки. По данным компаний Hewlett-Packard, IBM, Hughes Aircraft, TRW [2], исправление ошибки до начала этапа конструирования обходится в 10–100 раз дешевле, чем ее устранение в конце работы над проектом, во время тестирования приложения или после его выпуска. В десятках компаний было обнаружено, что политика раннего исправления дефектов может в два и более раз снизить финансовые и временные затраты на разработку [2].

Таким образом, работа с требованиями – важнейшая составляющая успеха программного проекта, где успех означает выпуск продукта, который удовлетворяет ожиданиям пользователей в качестве и функциональности при соблюдении бюджета и графика проекта. Повышение качества процессов работы с требованиями может принести целый ряд преимуществ, в том числе:

- уменьшение числа дефектов в требованиях, переделок и ненужных функций;
- снижение стоимости модификации;
- ускорение разработки;
- уменьшение разобщенности разработчиков и «расползания границ» проекта;
- более точные оценки при тестировании,

и тем самым повысить удовлетворенность заказчиков и разработчиков.

Управление требованиями

Понимание значимости процессов работы с требованиями привело к специализации и выделению отдельной роли – аналитика требований. Их также называют системными аналитиками, инженерами по требованиям, менеджерами по требованиям и про-

сто аналитиками. Методы извлечения и формализации качественных требований носит во многом эмпирический характер. Однако в практике разработки программных систем накопились определенные представления о том, какими свойствами должны обладать требования к программной системе [3]:

- полнота;
- ясность;
- корректность;
- согласованность;
- верифицируемость;
- необходимость;
- полезность при эксплуатации;
- осуществимость;
- модифицируемость;
- трассируемость;
- упорядоченность по важности и стабильности;
- наличие количественной метрики.

Процессы по работе с требованиями разделяются на две области – разработка требований и управление требованиями. В частности, методология усовершенствования процессов CMMI (Capability Maturity Model Integration) относит «управление требованиями» (Requirements Management) к процессам второго уровня зрелости, а «разработку требований» (Requirements Development) – к третьему [4].

Разработка инструментальных средств в области процессов «разработки требований» – задача крайне сложная. Причина связана с плохой формализуемостью этих процессов и их большой вариативностью. Задачи же области «управления требований» много проще с точки зрения их автоматизации. Инструментальные средства могут успешно решать следующие задачи:

- управление версиями и изменениями;
- хранение атрибутов требований;
- облегчение анализа воздействия (за счет хранения связей между требованиями);
- трассировка статусов требований;
- контролируемый доступ;
- связь со всеми заинтересованными в проекте лицами;
- повторное использование требований;
- интеграция с другими средствами управление проектами;
- сбор и обработка статистики требований.

Подходы к управлению требованиями

Широко распространен поход работы с требованиями с помощью создания структурированных документов (ТЗ, Software Requirements Specification, RUP Visio и т.д.). В лучшем случае такие документы размещаются в репозиториях с контролем версий. Данный подход имеет ряд существенных ограничений:

- трудность хранения дополнительной информации (атрибутов) для каждого требований;
- трудность определения взаимосвязей между требованиями разного уровня и другими элементами системы;
- трассировка статуса требований представляет собой трудный и неудобный процесс;
- одновременное управление наборами требований, запланированных для различных выпусков, затруднено;

- возможные проблемы в случаях географической распределенности команды разработки.

С этими ограничениями призваны бороться специализированные системы управления требованиями. Такие инструменты условно можно разделить на две группы. К первой группе относятся дорогие инструментальные средства, ориентированные в первую очередь на крупных и средних производителей программного обеспечения. Ко второй группе относятся недорогие инструментальные средства преимущественно с веб-интерфесом, ориентированным на небольшие компании.

Инструментальные средства первой группы обладают широким спектром предоставляемых возможностей и богатым пользовательским интерфейсов. Лидерами среди таких продуктов являются Borland CaliberRM, IBM RequisitePro, Telelogic Doors. По мнению экспертов [5], одной из проблем подхода применяемого в данных инструментах является излишняя обобщенность. Зачастую крупные и средние компании разработчики уже имеют вполне определенные процесс по работе с требованиями, и в качестве основного средства настройки инструментов под процесс предлагается создание новых типов требований с заданными атрибутами. При этом интерпретация этих атрибутов возлагается на пользователя.

Шагом в направлении адаптивности инструментальных средств под процессы может стать модульный подход в построение таких инструментов. Компании под свои нужды смогут «собирать» инструментальное средство из специализированных модулей. Примерами таких модулей могут быть модуль, отвечающий за определенный тип требований в системе, и модуль, отображающий некоторые важные для компании метрики требований.

Пользователи второй группы средств по управлению требованиями зачастую лишены каких-либо средств адаптации инструмента по свои нужды. Модульный подход позволит расширить возможности таких средств силами сторонних разработчиков.

Таким образом, решение задачи построения расширяемой за счет модулей системы управления требованиями позволить увеличить их адаптивность и, как следствие, применимость в компаниях-разработчиках программного обеспечения.

Реализация модульной системы управления требованиями

В качестве платформы для реализации модульной системы управления требованиями выбрана платформа .NET. Интерфейс пользователя реализован в виде веб-приложения с помощью технологии ASP.NET. Платформа .NET включает в себя технологию Reflection, которая позволяет гибко расширять программную систему во время исполнения. Технология ASP.NET позволяет компилировать страницы и пользовательские элементы управления при первом обращения пользователя, что позволяет расширять уже развернутую систему.

Разработанная система позволяет изменять свой функционал за счет установки новых модулей. Новый модуль устанавливается на серверной стороне и представляет собой zip-архив, который состоит из манифеста модуля и компонент (рис. 1). Манифест – это файл в формате XML, который содержит необходимые описания для установщика модулей, например импортируемые элементы управления (Controls). Компонентами являются любые сущности платформы ASP.NET, например страницы, элементы управления, сборки и т.д.

Вся логика пользовательского интерфейса системы сосредоточена во взаимодействующих элементах управления. Расположение элементов на странице реализуется с помощью файла шаблона. Страница же представляет собой лишь точку доступа в систему и не хранится в виде файла. Это позволяет модулям легко создавать нужные страницы на основе элементов управления и шаблонов. На рис. 2 представлена диаграмма основных классов пользовательского интерфейса.

Для гибкого изменения пользовательского интерфейса запрашиваемые страницы компонуются из элементов управления динамически, а не статически. Информация о том, какие элементы управления необходимо поместить на страницу и как их расположить, хранится в базе данных. Эту информацию могут менять устанавливаемые модули, тем самым изменяя пользовательский интерфейс.

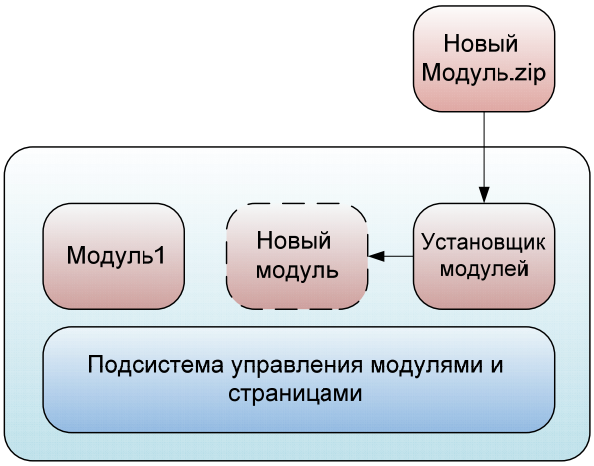


Рис. 1. Расширяемость за счет модулей

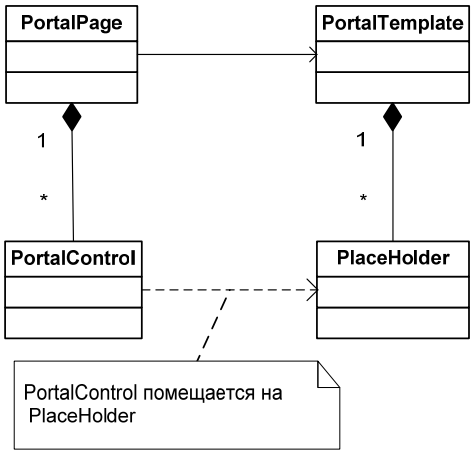


Рис. 2. Основные классы пользовательского интерфейса

Динамическая компоновка страниц делает задачу взаимодействия элементов управления на странице более сложной. Так, элементы управления не имеют информации от других элементах на скомпонованной странице и не могут вызывать их методов. Для решения данной проблемы предлагается использовать типовое решение «дополнительный модуль» (Plugin) [6], которое позволяет связывать классы во время исполнения, а не во время компиляции. Например, некоторому элементу на странице необходимо отображать требования, а за отображение требования отвечает другой элемент управления (рис. 3). Классы отделаются за счет выноса интерфейса, а их связывание производит после компоновки страницы.

Рис. 3. Типовое решение «дополнительный модуль»

Классы элементов управления декларативно публикуют реализуемое ими поведение с помощью атрибутов Export.

```
[Export(typeof(IViewRequirement))]
public partial class ViewRequirement : PortalControl, IViewRequirement
{
    protected void Show(Requirement req)
    {
        //...
        InitWebForm(req);
        //...
    }
}
```

Аналогично элементы публикуют необходимое поведение через свойства класса с атрибутами Import.

```
public partial class UserControl : PortalControl
{
    [Import()]
    public IViewRequirement view
    { get; set; }

    protected void Page_Load(object sender, EventArgs e)
    {
        Requirement r = GetLastRequirement();
        view.Show(r);
    }
}
```

Собрав публикуемую информацию о поведении, страница связывает свои элементы управления. Таким образом, элементы управления на страницах получают возможность совместно работать, ничего не зная, друг о друге, что позволяет динамически собирать сложные пользовательские интерфейсы по запросу пользователя.

На рис. 4 изображен пример пользовательского интерфейса редактирования требований. В левой части интерфейса расположено дерево пользовательских требованиями, в правой – редактор атрибутов требований.

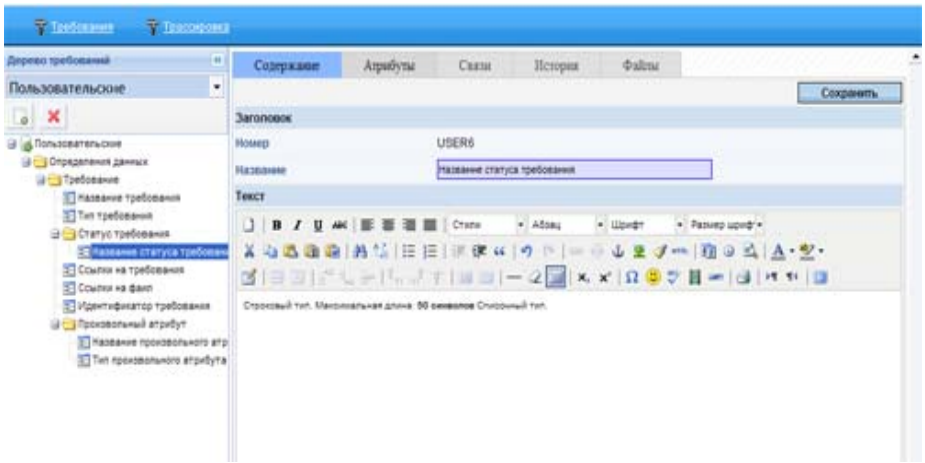


Рис. 4. Интерфейс редактирования требований

Заключение

Процессы работы с требованиями – это фундамент успеха программного проекта. Качество этих процессов в первую очередь определяется эффективностью применяемых методов извлечения, анализа, документирования, проверки и управления требова-

ниями. Инструментальные средства могут в значительной степени упростить решение ряда задач, связанных с работой с требованиями.

В статье предложен модульный подход к построению таких инструментальных средств. Этот подход позволил создать ядро расширяемой системы управления требованиями. За счет установки необходимых модулей можно настроить систему под существующие в компании процессы работы с требованиями, а также изменять ее по мере развития этих процессов.

Литература

1. Elizabeth Hull, Ken Jackson, Jeremy Dick. Requirement Engineering – Springer, 2005.
2. Макконелл С. Совершенный код. Мастер-класс. – СПб: Питер, 2008.
3. Вигерс К. Разработка требований к программному обеспечению – М.: Русская редакция, 2004.
4. Мильман К., Мильман С. CMMI – шаг в будущее. Часть 3. Разработка управления требованиями // Открытые системы. – 2005. – №9.
5. Anthony Finkelstein, Wolfgang Emmerich. The Future of Requirements Management Tools – Режим доступа: http://www.cs.ucl.ac.uk/staff/W.Emmerich/publications/OeCG/traunpaper.pdf
6. Фаулер М. Архитектура корпоративных программных приложений. – М.: Издательский дом «Вильямс», 2007.
7. Леффиигуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход – М.: Вильямс, 2002.
8. Орлик С., Булуй Ю. Введение в программную инженерию и управление жизненным циклом ПО. Программная инженерия. Программные требования. – Режим доступа: http://www.sorlik.ru/swebok/3-1-software_engineering_requirements.pdf

2 ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ СИСТЕМЫ

О НЕКОТОРЫХ ЗАДАЧАХ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

К.В. Рубинов
Научный руководитель – д.т.н., профессор А.А. Шалыто

Работа посвящена рассмотрению актуальных задач тестирования программного обеспечения, а также перспективных направлений их решения.

Введение

В настоящее время вопросу тестирования программного обеспечения (ПО) уделяется все больше внимания как со стороны производителей ПО, так и со стороны научной общественности. Потребность в решении задач тестирования ПО возникает при создании практически каждого программного продукта. При этом подходы к их решению могут существенно отличаться в зависимости от характеристик ПО, подлежащего тестированию.

Обзор литературы [1– 3] позволяет сделать вывод о том, что в области тестирования ПО ведутся активные исследования, а решаемые задачи по этой тематике актуальны. Настоящая работа посвящена рассмотрению наиболее важных, по мнению автора, задач тестирования ПО, а также перспективных направлений их решения. Предложенная классификация задач позволит проследить связи между зависимыми задачами с целью оптимизации процесса тестирования, а также поможет при выборе инструментов и подходов к их решению.

1. Постановка задачи

В работе приводится описание задач, связанных с процессом тестирования программного обеспечения. Чтобы показать связь этого процесса и возникающих в нем задач, отметим его основные этапы:

1. подготовка тестирования,
2. проектирование тестов,
3. разработка тестов,
4. создание тестовой среды,
5. выполнение тестов,
6. повторное (регрессионное) выполнение тестов.

В дальнейшем при рассмотрении задач будем обращаться к указанным этапам и порядку их следования.

В работе выделены не все задачи тестирования ПО, а только те из них, которые специфичны для этого процесса и могут быть интересны с точки зрения научных исследований, а также требуют специальных подходов к решению:

1. генерация и выбор тестовых данных,
2. создание модели тестирования,
3. задача тестового оракула,
4. выбор критериев тестового покрытия и оценка тестового покрытия,
5. генерация тестовых сценариев,
6. выбор тестов для повторного (регрессионного) тестирования.

Эти задачи появляются на разных этапах рассматриваемого процесса и, в общем случае, связаны между собой. Далее в работе будем исследовать каждую из задач в отдельности. При этом излагаются:

- описание задачи;
- предпосылки появления задачи;
- сложность решения задачи и ограничения;
- направления решения задачи;
- возможность автоматизации решения задачи;
- связи с другими задачами.

2. Задачи тестирования программного обеспечения

Основываясь на информации из современных источников (например, [1]) можно утверждать, что задачи тестирования основаны на следующих постулатах тестирования программного обеспечения:

- полное тестирование невозможно;
- стопроцентная автоматизация тестирования невозможна;
- общая теория тестирования отсутствует.

Поэтому для всех задач существует одна общая и, вероятно, основная причина их появления – это проблема выбора и нахождения баланса между подробным, трудоемким по времени и ресурсам тестированием, и менее глубоким, а, следовательно, более дешевым тестированием. Другими словами, большинство задач решаются с целью улучшения качества и глубины тестирования в условиях ограничений времени и других ресурсов.

2.1. Генерация и выбор тестовых данных

В настоящее время для большинства существующих и создаваемых программных систем существует проблема тестирования, известная под названием взрыв числа состояний (*State space/combinatorial explosion*). Проблема заключается в том, что невозможно выполнить тесты со всеми возможными комбинациями данных, которые могут быть использованы при функционировании систем. Это вызывает значительные сложности в процессе тестирования, а указанная проблема ставит перед разработчиками тестов ряд задач, основной из которых является задача генерации и выбора тестовых данных.

Рассматриваемая задача состоит в том, чтобы автоматически сгенерировать данные или комбинации данных. Помимо генерации, необходимо произвести отбор/выбор существенных данных из сгенерированных.

Выбор тестовых данных совершается для того, чтобы уменьшить их общее число и извлечь из всех возможных комбинаций для тестирования репрезентативные, которые позволяют выявить наибольшее число неисправностей, а также реалистичные данные, наиболее близкие к используемым при эксплуатации системы. При этом требуется учитывать ограничения на тестовые данные. Например, при генерации данных для загрузки памяти важно обеспечить возможность получить их в реальности в виде физических ресурсов памяти. Генерация сложных структур данных также является трудно разрешимой задачей.

Принимая во внимание ограничения и требования, предъявляемые к тестовым данным, для их выбора используются различные критерии отбора. Эти критерии могут быть основаны на используемых техниках тестирования, анализе рисков для тестируемой системы, критических обзорах документации к системе, выбранной стратегии тестирования, а также на основе других источников и их комбинациях.

Некоторые техники тестирования сами по себе нацелены на отбор данных – например, техника анализа граничных значений (*Boundary value analysis*), когда для тес-

тирования отбираются граничные значения данных, используемых в тестируемой системе [4]. При этом утверждается, что применение именно этих значений приводит к обнаружению часто встречающихся неисправностей.

Считается, что данная задача хорошо подходит для автоматизации, а отбор данных может быть совмещен с процессом генерации и производиться автоматически.

Для решения задачи генерации и выбора тестовых данных существует ряд подходов, начиная со случайной генерации данных (*Random test data generation*) и использования метода полного перебора и заканчивая подходами с применением генетических алгоритмов, комбинаторного анализа, а также построения и исследования моделей использования данных. Подходы могут быть основаны как на информации об исходном коде системы, предназначенной для тестирования, так и на ее спецификации [5].

Исследования по генерации тестовых данных ведутся повсеместно, а большое число публикаций приводит к тому, что разрабатываемые подходы достаточно сложно классифицировать. Широкий обзор направлений генерации тестовых данных представлен в работе [6]. Перечислим наиболее известные подходы:

1. случайная генерация данных (*Random test data generation*);
2. генерация данных на основе алгоритмов поиска (*Search-based test data generation*), в том числе генерация данных посредством генетических алгоритмов;
3. символьное исполнение программ (*Symbolic execution*);
4. генерация данных на основе ограничений (*Constraint-based techniques*).

Для отбора наиболее эффективных в плане тестирования данных существуют различные техники, такие как:

1. техники тестирования с выделением эквивалентных классов данных (*Category partitioning*),
2. ортогональные массивы и попарное тестирование (*Orthogonal arrays and pairwise testing*),
3. техника анализа граничных значений (*Boundary value analysis*).

Генерация и выбор тестовых данных производятся на этапах подготовки тестирования и проектирования тестов. От результатов решения этой задачи непосредственно зависит эффективность последующих этапов процесса тестирования – разработки и выполнения тестов.

2.2. Создание модели тестирования

На протяжении более десятилетия существует и развивается подход под названием тестирование на основе моделей (*Model-Based Testing – MBT*) [7, 8]. В его основе лежит использование моделей для контроля и управления процессом тестирования, в частности, для автоматической генерации тестов.

Известны работы [2, 9], в которых делаются попытки применить существующие модели для этих целей. Однако универсального подхода к использованию моделей для тестирования не существует. Поэтому вместо того, чтобы выбирать некоторую известную модель и исследовать, как она может быть использована для тестирования, можно продумать, как создать такую модель, чтобы с ее помощью ПО было эффективно протестировано [1].

В настоящее время для тестирования ПО применяются различные модели, такие как таблицы решений (*Decision tables*), конечные автоматы (*Finite state machines*), формальные грамматики (*Grammars*), модели Маркова (*Markov chains*), сети Петри (*Petri nets*), *UML*-модели (*UML models*), диаграммы состояний (*Statecharts*) и т.д. [8]. Эти модели могут быть использованы в тестировании с разной степенью эффективности. Причем оценить успех применения моделей можно только при анализе результатов тестирования и сравнении их с результатами других подходов. Немаловажно иметь сведения – подходит ли модель для тестирования конкретной системы или нет.

Эффективное применение модели тестирования может характеризоваться следующими показателями:

- возможностью автоматизации этапов процесса тестирования;
- применимостью модели для генерации тестовых данных и тестовых сценариев;
- однозначностью соответствия модели и спецификации тестируемой системы;
- возможностью оценки тестового покрытия для сгенерированных по модели тестов;
- удобством внесения изменений, сопровождения и понимания модели;
- сокращением ручного труда и времени разработки тестов.

Можно утверждать, что задача создания модели тестирования состоит в подборе и описании формальной модели, обладающей требуемым соотношением перечисленных выше показателей.

Трудоемкость процесса разработки моделей, сложность перехода от спецификации и исходного кода ПО к моделям вызывают трудности при их создании и использовании. В настоящее время развиваются подходы извлечения моделей из существующих программных систем [10]. Вместе с тем точность этих моделей и их пригодность для тестирования требуют исследования.

Помимо удобства использования моделей, необходимо, чтобы процесс работы с моделями был поддержан инструментальными средствами разработки программного обеспечения. При этом отметим, что в свободном доступе мало таких средств – многие являются либо закрытыми разработками компаний и учебных заведений, либо коммерческими приложениями [11].

Необходимо отметить, что решение задачи создания модели тестирования может непосредственно влиять и определять подходы к решению других задач тестирования ПО.

2.3. Задача тестового оракула

Приведем одно из определений тестирования. Тестирование – это оценивание программного обеспечения посредством наблюдения за его работой [2]. В данном случае будем рассматривать вопрос оценки поведения ПО: является ли поведение системы корректным в конкретной ситуации? Для ответа на этот вопрос необходим тестовый оракул.

Тестовый оракул – это система, метод или методика для предсказания или оценки корректности поведения системы, предназначенной для тестирования в определенных условиях [12]. Из этого определения следует, что без тестового оракула проведение тестирования не имеет смысла, так как невозможно оценить и контролировать процесс тестирования, а создание оракула – одна из важнейших задач тестирования ПО.

Сложность создания оракула связана с тем, что практически всегда во время проектирования и разработки тестов имеются допущения и обобщения относительно данных и поведения тестируемой системы. Вместе с тем необходимо создать такой оракул, который позволит однозначно определить корректность поведения системы в условиях неопределенности.

Разнообразные входные данные и воздействия, которые могут быть использованы в системе, весь спектр состояний, возможные реакции системы – все это должно быть учтено при создании тестового оракула. Процесс сравнения и вынесения вердикта существенно усложняется при оценке корректности поведения систем с данными со сложным информационным наполнением, например, аудио или видео, а также систем с временными ограничениями.

В роли тестового оракула может выступать человек. Это происходит при ручном тестировании, когда вердикт о результате теста выносит исполнитель теста. Опыт человека, выполняющего тест, позволяет проводить такие виды тестирования, как специальное/исследовательское (*ad hoc/exploratory testing*), когда не используются специальные документы, регламентирующие процесс тестирования [13]. В случае автоматиза-

ции тестирования можно говорить о том, что создание автоматических тестов возможно только с использованием тестовых оракулов [14].

Итак, описываемая задача состоит в том, чтобы создать тестовый оракул для программной системы с учетом ее характеристик в условиях существующей неопределенности с целью разработки автоматических тестов.

Существуют различные подходы к решению этой задачи. Например, сравнение поведения тестируемой системы с поведением предыдущей версии того же продукта или некоторой эталонной системы. Для тестирования графических интерейсов пользователей (*Graphical User Interface – GUI*) используют средства записи/воспроизведения (*record/playback*). В этом случае выполняется сравнение с эталонными образами и использование модели *GUI*. Иногда создание тестового оракула может быть более трудоемким и сложным процессом, чем создание тестируемой системы. Для ответственных систем (*Mission-Critical Systems*) могут быть разработаны и использованы несколько тестовых оракулов. Показателем корректности в этом случае будет среднее значение, полученное при сравнении результатов всех оракулов.

В заключение раздела отметим, что специфика рассматриваемой задачи определяется тем, что вопрос контроля корректности создаваемого тестового оракула остается открытым – необходимы способы обеспечения создания достоверных тестовых оракулов.

2.4. Выбор критериев тестового покрытия и оценка тестового покрытия

Для тестирования ПО разрабатываются тесты с различным целевым назначением, например, тесты, проверяющие базовую функциональность, нагрузочные тесты, тесты на производительность, тесты взаимодействия с другими компонентами или системами и т.д. Такие тесты объединяются в наборы, которые должны комплексно тестировать свойства и функции системы. Каждый набор тестов характеризуется целевым назначением и тем, какую функциональность он проверяет.

При этом возникает необходимость оценки этих характеристик, чтобы определить качество и полноту тестового набора по отношению к тестируемой системе. Существуют различные критерии оценки тестового покрытия, которые могут быть применены для этих целей. Число типов тестового покрытия достаточно велико [2, 15, 16]. Эти типы могут быть основаны на исходном коде ПО, спецификации ПО, вариантах использования программной системы и их модельных представлениях. В качестве примера назовем несколько типов тестового покрытия:

- покрытие ветвлений модуля программы (*branch coverage for program unit*);
- покрытие всех вершин в графе потока управления программы (*nodes coverage of control flow graph*);
- покрытие тестами требований из спецификации (*requirements coverage*).

Рассматриваемая задача состоит в выборе и сочетании критериев оценки тестового покрытия разрабатываемого набора тестов для программной системы и оценки этого покрытия. Выбор критериев должен осуществляться на этапах подготовки тестирования и проектирования тестов. Таким образом, тестовое покрытие может использоваться на протяжении всего процесса тестирования.

Основу для выбора критериев могут представлять данные анализа рисков системы, выбранная стратегия тестирования, а также требования, предъявляемые к системе. В то же время основная сложность в решении данной задачи обусловлена тем, что рассматриваемые оценки и сила критериев являются относительными.

Для оценки тестового покрытия производится сбор необходимых данных. В случае применения критериев оценки, основанных на спецификации, собирается информация о планируемом числе тестов, их целевом назначении и проверяемой фукнциональности. Затем строится матрица контроля (*traceability matrix*), по которой определяется степень покрытия тестами характеристик системы.

При использовании критериев, основанных на исходном коде ПО, оценка может производиться автоматически с помощью специальных средств анализа тестового покрытия кода приложений [17]. Также получение оценки можно автоматизировать при использовании в процессе тестирования модели. При этом необходимая информация извлекается из модели автоматически.

Кроме того, что применение оценок тестового покрытия позволяет определять необходимое число тестов и данных для тестирования, существует другая сторона задачи тестового покрытия. Использование данных о тестовом покрытии может быть основой для решения вопроса о прекращении тестирования. Проблема завершения тестирования выделена среди основных актуальных проблем программной инженерии в работе [3]. В настоящее время существует потребность в разработке параметров и моделей, которые позволяли бы выявлять момент прекращения тестирования на основе специальных критериев.

Можно отметить также, что в инженерии надежности ПО (*Software Reliability Engineering*) тестовое покрытие является одним из механизмов предсказания надежности разрабатываемого ПО [18]. Информация о тестовом покрытии может быть использована наравне со знаниями о рисках тестируемой системы как инструмент для принятия решений о выборе тестов для повторного (регрессионного) тестирования. При этом критерии тестового покрытия могут применяться при выборе данных для тестирования вместе с критериями отбора тестовых данных, описанных в разд. 2.1.

2.5. Генерация тестовых сценариев

Тестовый сценарий – это основа любого теста. Сценарий содержит последовательность тестовых воздействий и соответствующие тестовые данные для тестирования системы в определенных состояниях. Помимо этого, тестовые сценарии должы быть оптимальны для тестируемой системы/программы – обладать следующими свойствами:

- должна отсутствовать избыточность и дублирование тестовых воздействий;
- должно быть обеспечено соответствие требованиям времени выполнения сценария;
- должно быть ограничено максимальное число тестовых воздействий.

Слишком длинная последовательность воздействий в тестовом сценарии может вызвать неоправданно длительный процесс выполнения теста. Это может помешать оптимальному выбору и исполнению тестов в дальнейшем. Слишком короткая последовательность может вызвать противоположную ситуацию – свести к минимуму эффективность применения сценарного тестирования из-за малого отличия от использования единичных тестовых воздействий.

Глубина и полнота тестирования непосредственно зависят от качества разработанных тестовых сценариев. Таким образом, необходимы формальные методы для создания оптимальных тестовых сценариев, а также способствующие автоматизации генерации тестовых сценариев.

Число воздействий в тестовых сценариях, как и число самих сценариев, может быть бесконечным. Поэтому появляется необходимость выбора тестовых сценариев. Так же, как и в задаче выбора тестовых данных (разд. 2.1), для этого применяются различные подходы и техники, целью которых является отбор сценариев, позволяющих провести проверки в значимых состояниях системы с меньшими затратами времени и других ресурсов.

Применение моделей тестирования может существенно облегчить процесс генерации тестовых сценариев. В настоящий момент известен ряд техник, методов и программных средств, позволяющих автоматически генерировать тестовые сценарии на основе разнообразных моделей [11].

В общем случае решение задачи с применением моделей тестирования сводится к созданию формального представления модели, которое описывает поведение системы, варианты использования системы или ограничения из спецификации. При этом модель

может базироваться на спецификации или исходном коде системы. Далее должен выполняться переход к представлению модели в виде графа, а затем осуществляться обход графа с использованием различных алгоритмов [19, 20] с последующей генерацией тестовых сценариев. В настоящее время существуют подходы для генерации тестов на основе спецификаций, в которых описанные действия производятся автоматически [5].

Существуют и применяются также такие виды тестирования, как *ad hoc/exploratory*. При этом тестовые сценарии не задаются заранее, а формируются инженером по тестированию непосредственно в процессе выполнения воздействий на тестируемую систему.

2.6. Выбор тестов для повторного (регрессионного) тестирования

Как и большинство задач тестирования, эта задача состоит в выборе альтернатив и в достижении компромисса между множеством условий и ограничений. В соответствии с работой [4], основная проблема регрессионного тестирования состоит в поиске компромисса между доступными ресурсами и необходимостью проведения повторных тестов по мере внесения каждого изменения в программную систему. В определенной степени рассматриваемая задача состоит в том, чтобы определить критерии «масштабов» изменений в ПО, с достижением которых необходимо проводить регрессионные тесты.

Таким образом, при регрессионном тестировании должны быть учтены критерии влияния различных факторов, таких как сложность редактируемого программного кода (*code complexity*) и архитектурные особенности ПО, включая информацию о компонентах, которые используют модифицируемую часть системы или используются ей.

Решения о выборе тестов принимаются на основе приоритетов тестов, назначенных на основе анализа рисков системы. Кроме этого, рассматривается матрица контроля (*traceability matrix*) для нахождения тестов, соответствующих требованиям спецификации или блокам кода ПО. После этого производится расчет необходимого времени выполнения тестов. Опеделяются операционные профили[1]. Собирается информация о функциональности системы, проверенной на предыдущих итерациях выполнения тестов, а также информация о том, сколько циклов повторений регрессионных тестов запланировано в процессе тестирования ПО. Кроме того, принимается во внимание готовность компонентов программного продукта к тестированию.

С учетом изложенного формируются различные подходы к решению задачи. В общем случае эти подходы выражаются в виде стратегий выбора тестов на протяжении всего процесса тестирования. Известны случаи создания моделей изменений ПО для регрессионного тестирования [21]. Эти модели позволяют выявлять зависимости в ПО, которые необходимо протестировать, а после этого автоматически генерировать тестовые наборы для регрессионного тестирования.

Можно отметить, что информация, полученная при решении вопросов остановки тестирования и оценки тестового покрытия (разд. 2.4), может быть непосредственно использована при подготовке и выборе тестов для регрессионного тестирования.

Заключение

В настоящей работе приведен обзор актуальных задач тестирования программного обеспечения. Рассмотрены вопросы автоматизации решения задач. Определена связь между задачами в рамках процесса тестирования ПО, и указаны пути решения рассмотренных задач.

[1] Операционный профиль (*operational profile*) – набор тестовых данных, отражающий реальные условия функционирования программы с учетом вероятности их возникновения.

Литература

1. Bertolino A. Software Testing Research: Achievements, Challenges, Dreams /Future of Software Engineering. Briand L. and Wolf A. (eds.). IEEE-CS Press. 2007, pp. 85–103.
2. Ammann P., Offutt J. Introduction to software testing // NY: Cambridge University Press. 2008. 344 pp.
3. Воас Д. Чертова дюжина проблем программной инженерии // Открытые системы. – 2007. – № 7. – С. 30–35. – Режим доступа: http://www.osp.ru/os/2007/07/4391815/
4. Орлик С. Программная инженерия. Тестирование программного обеспечения (Software Testing) /Введение в программную инженерию и управление жизненным циклом ПО. 2004. – Режим доступа: http://www.sorlik.ru/swebok/3-4-software_engineering_testing.pdf
5. Offutt J., Liu S., Abdurazik A., Ammann P. Generating Test Data From State-based Specifications /The Journal of Software Testing, Verification and Reliability. – 2003. – 13(1). – P. 25–53.
6. McMinn P. Search-based software test data generation: A survey /Software Testing, Verification and Reliability. – 2004. – 14(2). – P. 105–156.
7. DACS (Data and Analysis Center for Software) Gold Practices Website. Model-based testing. – Режим доступа: https://www.goldpractices.com/practices/mbt/
8. El-Far I.K., Whittaker J. Model-Based Software Testing. Encyclopedia on Software Engineering (edited by Marciniak J.). Wiley. 2001. 225 pp.
9. Al-Ghafees M., Whittaker J. Markov Chain-based Test Data Adequacy Criteria: a Complete Family // IS June 2002. – P. 13-37.
10. Acharya M., Xie T., Pei J., Xu J. Mining API patterns as partial orders from source code: from usage scenarios to specifications / In Proc. ESEC/FSE 2007. – Режим доступа: http://people.engr.ncsu.edu/txie/publications/esecfse07.pdf
11. Hartman A. AGEDIS model based test generation tools. – Режим доступа: http://www.agedis.de/documents/ModelBasedTestGenerationTools.pdf
12. Блэк Р. Ключевые процессы тестирования: планирование, подготовка, проведение, совершенствование. – М.: Лори, 2006. – 576 с.
13. Bach J. Exploratory Testing Explained. 2002. 10 pp. – Режим доступа: http://www.satisfice.com/articles/et-article.pdf
14. Hoffman D. Using Oracles in Test Automation /Software Quality Methods. – 2001.
15. Towhidnejad M. Types of Software Testing Coverage. – Режим доступа: http://faculty.erau.edu/towhid/coverage.html
16. Kaner C. Software Negligence and Testing Coverage. – Режим доступа: http://www.kaner.com/coverage.htm
17. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения. Внедрение, управление и эксплуатация. – М.: Лори, 2003. – 592 с.
18. Lyu, M. Software Reliability Engineering: A Roadmap /Future of Software Engineering. Briand L. and Wolf A. (eds.). – IEEE-CS Press, 2007. – P. 153–170.
19. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. – СПб: Питер, 2004. – 318 с.
20. Tonella P. Evolutionary Testing of Classes /Proceedings of the 2004 ACM SIGSOFT international symposium on Software testing and analysis ISSTA '04. ACM Press. – NY, 2004. – P. 119–128.
21. Chen Y., Probert R., Ural H. Model-based regression test suite generation using dependence analysis / In Proc. of the 3rd International workshop on Advances in model-based testing. – London, 2007. – P. 54–62.

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ОБРАЗОВАТЕЛЬНЫМ ПРОЦЕССОМ ВУЗА

А.Н. Зайцева (Нижегородский филиал государственного университета – Высшей школы экономики)
Научный руководитель – к.т.н., доцент Э.А. Бабкин (Нижегородский филиал государственного университета – Высшей школы экономики)

В работе рассмотрена проблема автоматизации образовательного процесса вуза. Были рассмотрены проблемы, с которыми сталкиваются участники образовательного процесса, и существующие варианты их решений. После проведенного анализа недостатков и уточнения пожеланий пользователей была разработана собственная система.

Введение

Любая организация, как и человек, для координации своих действий должна иметь механизм внутренних коммуникаций. Каждое образовательное учреждение, независимо от того, в какой области оно специализируется (будь то гуманитарный институт или технический вуз), имеет свои базовые рабочие процессы, которые обеспечивают функционирование системы в целом (прием студентов, составление расписания учебных занятий, проведение контролей усвоения материалов и пр.).

Требования эффективности и качества стимулируют широкое развитие автоматизации управления такими процессами. Каждое подразделение может использовать локальные приложения для выполнения своих функций, которые сами по себе могут работать стабильно, однако данные этих систем остаются изолированными друг от друга, и их интеграция является весьма трудной задачей. При таком подходе отсутствует столь важный компонент, как связи между разными видами информации.

Большое значение для любой системы имеют хорошие рефлексивные показатели, которые говорят о том, как быстро система может реагировать на любую неожиданность. Примером может служить своевременное информирование студентов о том, что преподаватель не сможет провести занятие в связи с внезапно возникшими трудностями. Поэтому в столь быстро меняющейся экономической и политической мировой ситуации информационная система вуза должна обеспечивать доступ к самым свежим, достоверным и аргументированным данным.

Скачок в развитии информационных технологий стал основой разработки новой федеральной целевой программы в России – «Развитие образования на 2006–2010 гг.» [5], в рамках которой планируется провести «совершенствование системы управления образованием на основе эффективного использования информационно-коммуникационных технологий в рамках единого образовательного пространства».

Хочется отметить, что и до внедрения вышеуказанной программы многие учебные и коммерческие заведения технического профиля занимались разработкой и внедрением автоматических систем управления учебным процессом. Среди них следует особо выделить: Naumen (система Naumen University), Якутский государственный инженерно-технический институт (Web-ориентированная система управления учебным процессом) и Российский государственный университет путей сообщения (Автоматизированная система управления учебным заведением – АСУ УЗ). Данные системы успешно прошли интеграцию в высших учебных заведениях и активно развиваются. Но вместе с тем, их характерной особенностью является узость решаемых задач (проект ЯГИТИ, АСУ УЗ) или же сложность программной реализации (Naumen University).

Поэтому в рамках данной работы мы стремились к созданию гибкой, легко адаптируемой к нуждам организации (с учетом системы рейтинговых оценок) системы, которая позволила бы не только автоматизировать часть «бумажной работы» сотрудников учебного заведения, снабжать всех заинтересованных лиц своевременной и достовер-

ной информацией, но и создавала бы благоприятные условия для наиболее полного взаимодействия между различными участниками учебного процесса.

В следующих главах работы содержатся требования к автоматизированным системам управления образовательным процессом, анализ уже существующих систем, описание использованных в процессе разработки нашей собственной системы технологий, программных продуктов, краткое описание архитектуры и функциональности нашей системы.

Требования к системе управления образовательным процессом

Эффективность управления образовательным процессом представляет собой:

- обеспечение быстрого и удобного доступа к необходимой информации;
- налаженную систему коммуникаций между участниками системы;
- оптимизацию использования информационных и временных ресурсов.

Для обеспечения эффективного управления учебным процессом необходимо внедрение и оптимальное использование новейших информационных технологий. При необходимости нужно адаптировать деятельность системы с учетом интересов руководства, студентов, инвесторов, государства и прочих заинтересованных субъектов. В связи с этим разрабатываемая система должна удовлетворять определенным требованиям, основными из которых [4] являются:

1. интегрированность,
2. адаптируемость,
3. распределенность,
4. масштабируемость,
5. качество.

В частности, система управления образовательным процессом должна обладать следующими ключевыми возможностями [3]:

- обеспечивать взаимодействие различных функциональных модулей, как по данным, так и по процедурам обработки;
- иметь развитые технологические методы интеграции с другими прикладными системами и базами данных.

Данные требования были полностью учтены при разработке нашей информационной системы, именуемой далее информационным порталом.

Анализ существующих программных решений

Прежде чем приступать к разработке собственного программного продукта, нами был изучен рынок существующих средств автоматизации учебного процесса. Среди наиболее крупных систем мы хотели бы выделить:

- Naumen University – разработка российской ИТ-компании Naumen;
- Автоматизированная система управления учебным заведением (АСУ УЗ) – разработка Российского государственного университета путей сообщения (РГУПС);
- Web-ориентированная система управления учебным процессом – разработка Якутского государственного инженерно-технический института (ЯГИТИ).

Анализируя функциональность данных приложений, мы заметили, что система ЯГИТИ имеет достаточно узкую область применения. Ориентированный на реализацию идеи дистанционного образования, данный продукт позволял пользователям, подключившись к web-серверу института, просмотреть расписание курсов, выполнить в режиме on-line практические задания и получить информацию о текущей успеваемости. Таким образом, данная система по набору предоставляемых информационных компонент не соответствовала заявленным выше требованиям [10].

В свою очередь, система, предложенная разработчиками РГУПС [7], имела сугубо прикладной характер, позволяя лишь автоматизировать документооборот внутри организации. Пользователями данной системы преимущественно являются секретари, преподавательский состав и финансовый отдел организации. Используя различные программные модули, они могут получить доступ к учебным планам, распределению нагрузок или стандартным шаблонам документов и т.п. Данная система, хоть и снижает существенно объем «бумажных работ», не способствует построению прямого канала связи между студентами и кафедрой и не способна обеспечить пользователей своевременной информацией.

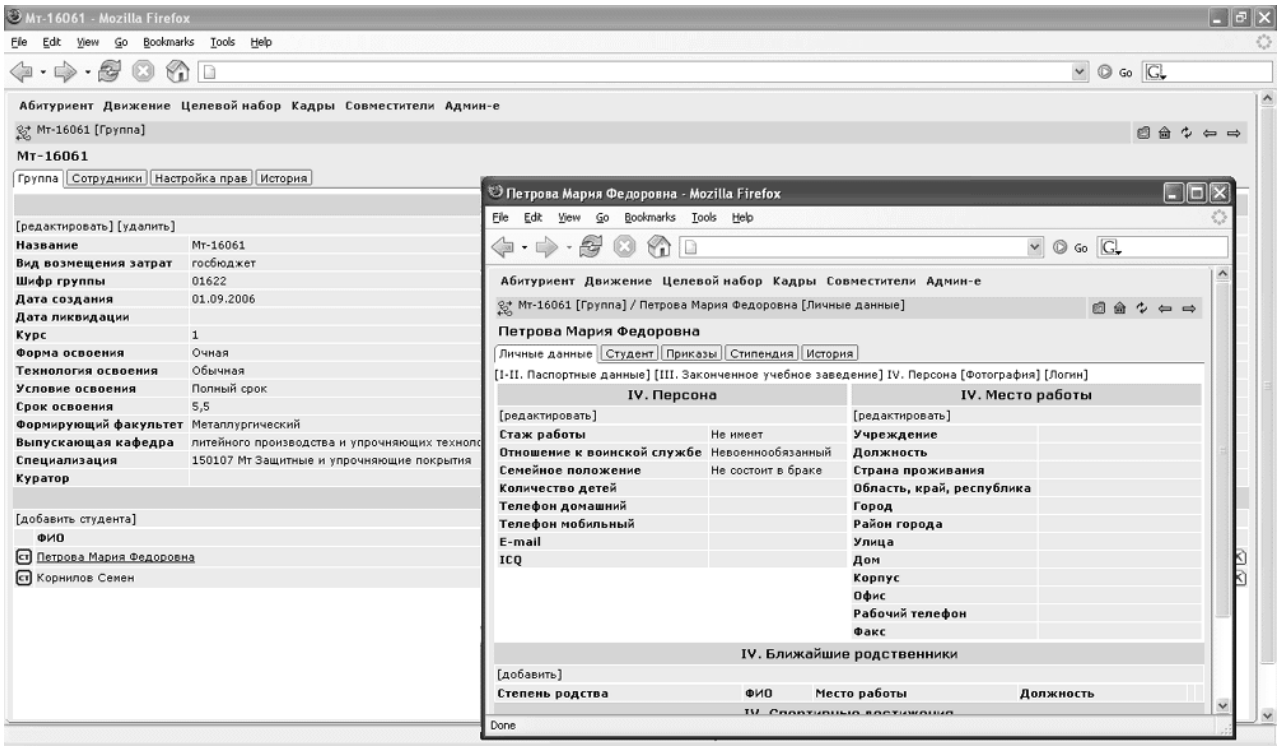


Рис. 1. Страница группы и личная страница пользователя в системе Naumen University

Naumen University [6], пожалуй, является одной из лучших систем автоматизации учебного процесса, которая существует на данный момент в России. Программный комплекс разработан с использованием технологии Java и стандартов Java-сервлетов и является совместимым с серверами Tomcat (начиная с версии 5.0). Система полностью отвечает всем предъявленным выше требованиям за исключением адаптируемости: данный программный продукт, как мы установили, не может быть перестроен в соответствии с рейтинговой системой, принятой в Государственном университете – Высшей школе экономики. Среди недостатков данной системы можно выделить сложный, недружественный интерфейс пользователя и отсутствие прямого канала связи между преподавателями и студентами (форум и т.п.).

Создавая свой информационный портал, мы принимали во внимание опыт разработчиков по созданию информационных систем для высших учебных заведений [2]. Вместе с тем мы стремились создать качественный, многофункциональный, уникальный в своем роде продукт, который мог бы стать не только площадкой для дальнейших исследований в области автоматизации учебных и бизнес-процессов (как это происходит со многими подобными системами). Мы предполагали внедрение разработанных технологических решений на практике, создавая тем самым единый источник информации для нашего вуза и прямой канал связи между различными участниками учебного процесса.

Выбор технологии и средств разработки программного обеспечения

Представленные выше требования во многом определяют выбор инструментов и методик, используемых в процессе создания системы.

Для реализации данной задачи было принято решение использовать свободно распространяемый продукт, так как временные и трудовые затраты в данной ситуации оказываются неоправданно велики, и использовать готовый программный продукт. В рамках данной технологии существует 2 лидирующих программных продукта, на базе которых возможна разработка информационного портала: OFBiz (Open for Business) и Pentaho. Для определения наилучшего решения был произведено сравнение систем.

Критерии сравнения	OFBiz [8]	Pentaho [9]
1. Наличие бизнес-ориентированных моделей данных, возможности по их созданию, расширению, связыванию.	Содержит широкий спектр готовых бизнес-ориентированных моделей данных (entity), позволяет легко создавать и использовать собственные модели. Возможно построение связей между моделями данных, их совместное использование и перестроение.	Не поддерживает бизнес-ориентированных моделей данных (преимущественно логические модели и метаданные). Механизм изменения моделей негибок, сама модель данных сложна в управлении. Отсутствует механизм одновременной обработки нескольких моделей данных.
2. Возможности по управлению бизнес-процессами в составе системы	Содержит особый компонент Workflow Engine, предназначенный для автоматизации выполнения бизнес-процессов в соответствие со спецификациями и стандартами WfMC и OMG. Не так давно был интегрирован с Shark.	Обладает превосходным набором средств для создания, управления, исполнения и контроля над бизнес-процессами. Среди них: язык описания потоков работы XPDL и распорядитель выполнения работ Shark. Поддерживает стандарт WfMC .
3. Поддержка аналитической обработки данных	Содержит все необходимые инструменты для анализа данных (в т.ч. статистические модели) и интерпретации полученных результатов.	Включает отдельный модуль, реализующий различные инструменты для анализа данных и интерпретации полученных результатов.
4. Уровень графического интерфейса и средства по его созданию	Представляет широкий спектр возможностей по созданию графического интерфейса, в т.ч. динамическая генерация интерфейсов из заранее созданных файлов на языках html, xml, PDF и др. Позволяет использовать шаблоны оформления для создания корпоративных сайтов (CCS).	Низкий уровень графического интерфейса, который не всегда является интуитивно понятным. Интерфейс является жестким и практически недоступным для изменения.
5. Возможности локализации интерфейсов в системе (русский / английский языки) для	Обладает набором средств локализации интерфейса создаваемого в системе приложения с учетом языковой принадлежности пользователя (lo-	Локализация недоступна.

Критерии сравнения	OFBiz [8]	Pentaho [9]
разных типов устройств: ПК – КПК - мобильный телефон	cale).	
6. Уровень защиты от несанкционированного доступа	Безопасность в рамках системы обеспечивается за счет следующих механизмов: использование свободно распространяемого программного обеспечения с открытыми исходными кодами, соответствующего промышленным стандартам; отсутствие записи на компьютер пользователя каких-либо компонентов, ведущих к потере пользователем контроля над своим компьютером; поддержкой механизмов аутентификации: парольная аутентификация и аутентификация при помощи сертификатов.	Система использует свободно распространяемое ПО, защищенное сертификатом доступа. Вместе с тем, модуль защиты компонентов и аутентификация пользователей еще не реализованы до конца (т.к. система развивается относительно недавно по сравнению с OFBiz). Поэтому разработчику приходится самому настраивать защиту доступа пользователей к различным компонентам системы в соответствии с его потребностями.
7. Активность разработчиков	Продукт активно развивается, как силами официальной команды разработчиков, так и просто энтузиастами.	Продукт привлекает разработчиков новизной. Активно развивается, преимущественно сторонними разработчиками.
8. Работа с базами данных (простота в подключении, наличие встроенных СУБД и их удобство)	Система содержит встроенную СУБД, доступную для эксплуатации. Предоставляет возможность подключения сторонних баз, но данное решение сопровождается существенными трудовыми и временными затратами на установку стабильного соединения.	Не содержит внутренних СУБД. Через компонент Mondrian осуществляет взаимодействие с SQL-базами. При этом пользователь не обязан создавать запросов на языке SQL. Система иногда ведет себя нестабильно.

Таблица 1. Сравнительные характеристики систем OFBiz и Pentaho

Из приведенной табл. 1 ясно видно, что наиболее подходящим программным продуктом для решения поставленной перед нами задачи является OFBiz.

1. Более устойчив и содержит большее число стабильно работающих встроенных компонентов.
2. Предоставляет широкие возможности для разработки пользовательского интерфейса и локализации, что немаловажно при создании Web-ориентированного приложения.
3. Достаточно легко интегрируется с другими системами, может быть установлен на мобильный телефон или КПК.
4. Предоставляет разработчикам развитую систему обеспечения безопасности, включая авторизацию пользователей, создание групп с различными правами и назначение ролей.

5. OFBiz содержит собственную СУБД, что упрощает задачу по сбору, хранению и управлению данными пользователей системы и информационным контентом.
6. Компонентная архитектура системы OFBiz гарантирует ее масштабируемость и гибкость.

Таким образом, выбор данного программного продукта гарантирует интегрируемость, адаптируемость, распределенность и масштабируемость информационной системы.

Описание архитектуры системы

В разработанной нами системе можно выделить 3 основные архитектурные компоненты: сервер баз данных, сервер приложений и клиентскую часть.

В качестве постоянного хранилища данных может использоваться собственная база системы OFBiz или сервер, который удовлетворяет определенным требованиям к функциональности (поддержка транзакций и др.) и для которого существует драйвер JDBC (Java Database Connectivity) уровня не ниже 2.

Сервер приложения реализован на основе программного продукта OFBiz, он создан на языке Java и использует серверные технологии J2EE. Сервер OFBiz чрезвычайно модулярен, что позволяет легко настраивать и расширять систему. Деление сервера на модули происходит в соответствии с принципами Stovepipe Architecture. Функциональность в системе реализована в виде служб или сервисов на языке Java. Для корректной работы системы требуется JDK версии не ниже 1.4 [1].

Основными функциональными составляющими системы являются собственно Интернет-портал и приложение, позволяющее составлять расписание в интерактивном режиме. Интернет-портал реализован в виде web-приложения в системе OFBiz, включающее приложение для составления расписания в качестве сервиса.

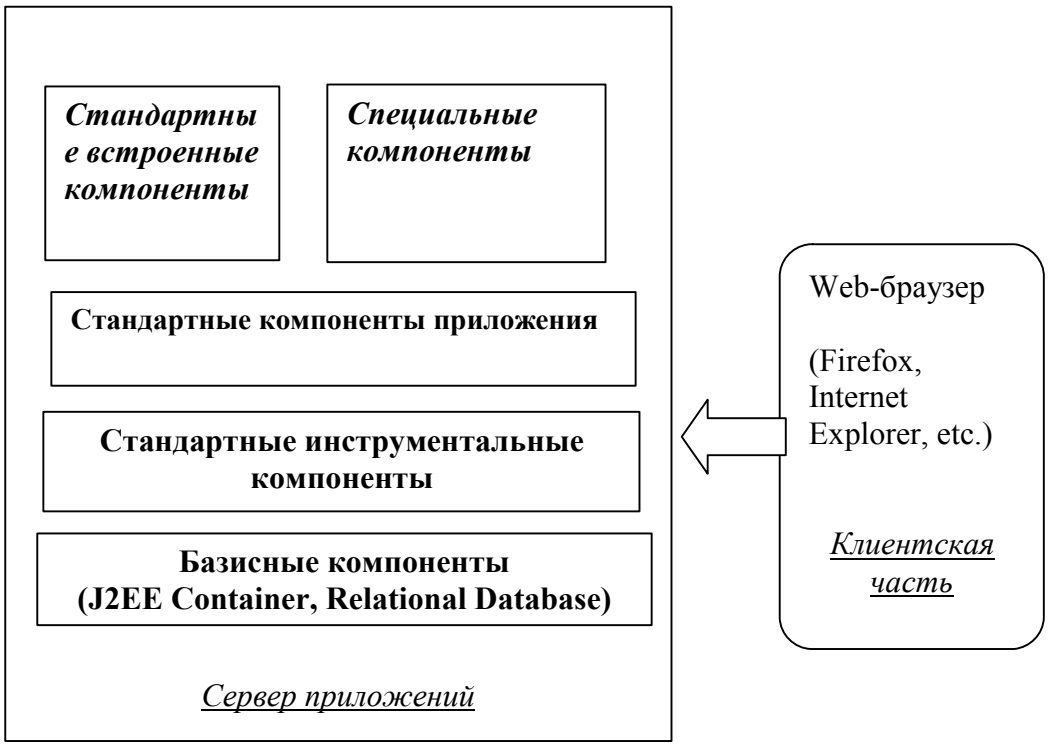


Рис. 2. Архитектура информационного портала

Описание функциональных возможностей информационной системы

Построенный на базе вышеуказанных систем информационный портал предоставляет доступ к различным службам через клиентский интерфейс следующим группам пользователей:

- Secretary (секретарь или администратор),
- Professor (преподаватель),
- Student (студент факультета),
- Entrant (гость, абитуриент).

Всем группам, особенно неавторизированным пользователям, предоставляется полная информация о вузе, преподавательском составе, кафедрах, специальностях, а также об организации внеучебного времени и различная литература, необходимая в учебных целях. В то же время доступ к информации и сервисам отличается у различных пользователей системы. Для контроля доступа к системе производится аутентификация пользователей по индивидуальному или групповому паролю.

В настоящее время пользователи портала имеют доступ к следующим функциям.

- Получение информации о различных информационных проектах факультета. Портал должен предоставляет возможность подключения других проектов, разработанных студентами в качестве научных работ на базе продукта OFBiz или с использованием технологии Java.
- Возможность интерактивной работы: форум, новости, опросы, составление и получение расписания. Студенты и преподаватели должны получать достоверную информацию в кратчайшие сроки. Для достижения этой цели создан раздел новостей. Функция по созданию новостей доступна секретарю. В целях повышения эффективности учебного процесса организован сбор различной информации посредством обновляющейся системы опросов.

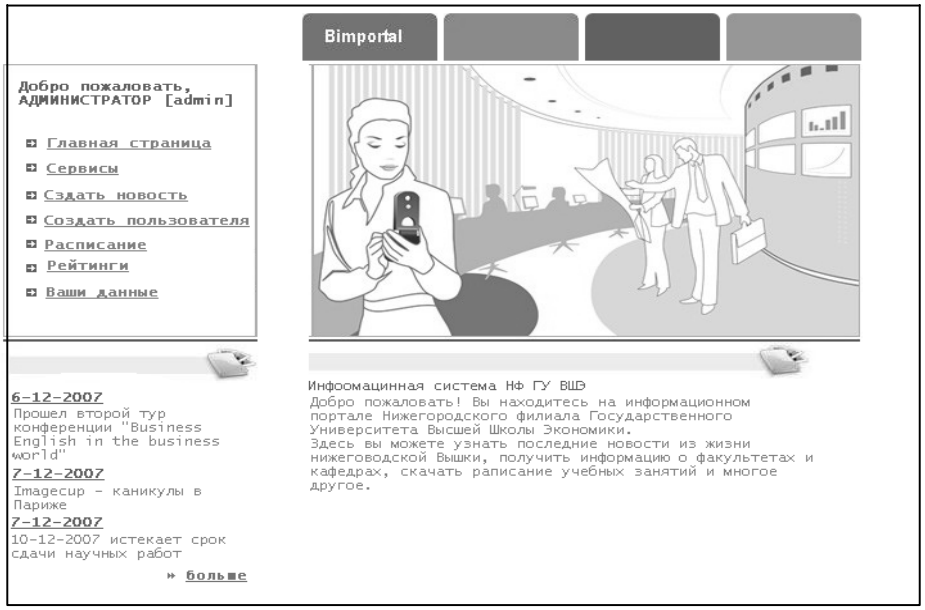


Рис. 3. Заглавная страница информационного портала с разделом новостей

- Получение информации о преподаваемых курсах и предоставление доступа к материалам курсов.
- Создание учетной ведомости предметов (для группы Professor) с механизмом учета домашних заданий (и формированием рейтинга). Студенты имеют возможность передавать домашние выполненные задания через сервис портала. Преподаватель, в свою

очередь, используя соответствующие приложения, ведет учет сданных (или несданных) вовремя работ, выставляет оценки, размещает тексты заданий. В конце курса преподаватель выставляет итоговые оценки по своему предмету, которые автоматически объединяются в таблицу рейтинга с учетом используемых в нашем вузе коэффициентов. В зависимости от позиции в рейтинге студенту начисляется стипендия. Таким образом, данный сервис позволяет ускорить обмен информации между пользователями.

Рис. 4. Экран ведомости успеваемости студентов по предмету «Математическая статистика»

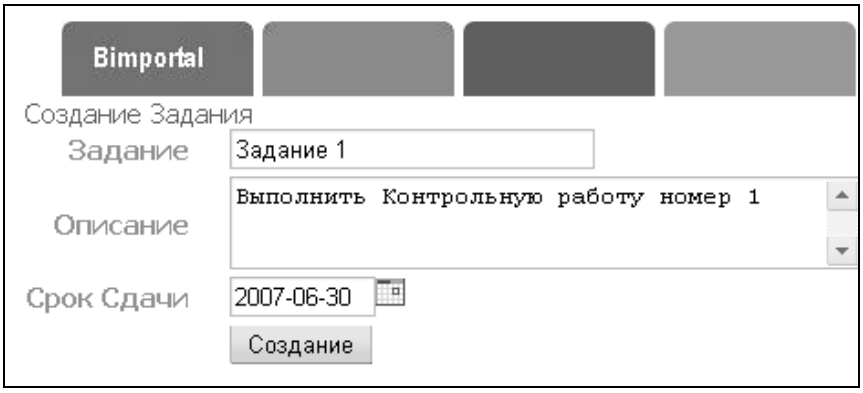


Рис. 5. Экран пользователя группы Professor с функцией «Создание Задания»

- Получение информации о подготовительных курсах, порядке и графике сдачи экзаменов, порядке и сроках подачи документов на зачисление, порядке и величине платежей для учащихся, поступающих на коммерческой основе (для группы Entrant).
- Поучение информации о тематике магистерских и бакалаврских работ, взаимодействие с научными руководителями, информация о действующих научно-исследовательских проектах.
- Подписка на рассылку (для групп Students и Professors). Пользователи данных групп могут быть в соответствие с их пожеланиями включены в базу рассылки системы, после чего им будут автоматически отправляться обзоры последних новостей, проектов и важнейших событий в жизни факультета.

Разработанная нами система удовлетворяет всем вышеуказанным требованиям, позволяет своевременно обеспечивать пользователей достоверной информацией и частично автоматизировать образовательный процесс. На сегодняшний момент продолжа-

ется работа над текущими компонентами с целью улучшения пользовательского интерфейса, а также рассматриваются возможности интеграции в систему новых модулей для расширения ее функциональности (например, использование DataWerehousing для контроля над порталом и выявления активно посещаемых приложений).

Перспективы развития системы

В дальнейшем планируется расширение системы и преобразование ее из информационно-учебного портала в полнофункциональную автоматизированную информационную систему, функционирующую в рамках вуза.

Рис. 6. План наращивания функциональности системы

Данная задача может быть разбита на 3 этапа:

- создание информационного портала;
- развитие системы автоматизированного документооборота;
- наращивание функциональности и создание полнофункциональной автоматизированной системы.

В добавление к описанной функциональности информационного портала система автоматизированного документооборота будет включать следующие модули.

- Party – личные страницы сотрудников вуза и студентов: расписания занятий, данные об успеваемости, важные встречи и события, деловая переписка, сведения о научной деятельности, рейтинги, закладки поиска, личные данные (дата рождения, контакты).
- Human Resources – управление персоналом (контроль должностей на кафедрах, время присутствия на кафедре и т.п.).

И, наконец, реализация последней компоненты подразумевает выполнение следующих этапов бизнес-процесса:

1) в конце каждого модуля заполняются необходимые экзаменационные и зачетные ведомости, и составляется рейтинг;
2) на основании общего учебного плана специальности автоматически формируется учебный план для группы на год. Система составления расписаний позволяет согласовать планируемые нагрузки с пожеланиями студентов;
3) Financials – бухгалтерия (организация бухгалтерского и управленческого учета);
4) Facility – контроль занятости помещений и изменение аудиторного фонда (анализ состояния аудиторий, их используемость);
5) Inventory – инвентарный учет (помощь при проведении инвентаризации).

Также возможно внедрение в систему модуля безопасности в случае введения в рамках учебного заведения автоматизированной охранной системы.

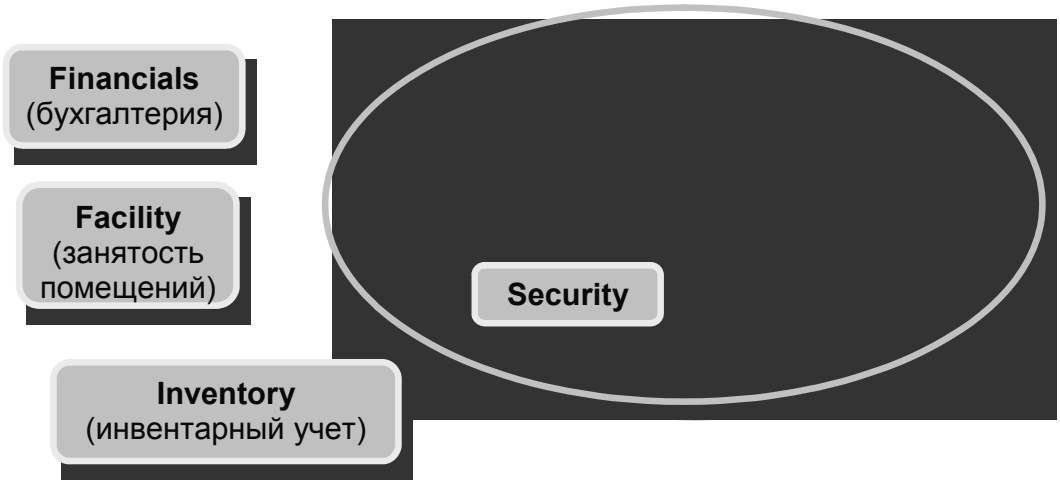


Рис. 7. Полнофункциональная автоматизированная информационная система

На конечном этапе своего развития система будет представлять собой полноценный автоматизированный комплекс полезных программ и приложений, который, мы думаем, при соответствующей технической поддержке сможет создать единую информационную систему, объединяющую не только сотрудников и студентов Высшей школы экономики, но и сторонних агентов. Это позволит людям своевременно получать точную информацию и решать возникающие проблемы на ранних стадиях развития.

Заключение

В заключение хочется отметить, что нами был создан уникальный программный продукт, который позволяет не только создать единое информационное пространство для такой крупной организации, как Нижегородский филиал Высшей школы экономики, но и обеспечить максимальное удобство пользователей при взаимодействии в его рамках. В отличие от уже существующих программных реализаций, наш портал не только выполняет информативную функцию, но и позволяет автоматизировать часть рутинных процессов, поддерживает прямой канал связи между преподавателями и студентами в виде форума и сервиса составления расписаний, имеет простой, интуитивно понятный интерфейс.

Следует отметить, что в настоящее время созданный нами информационный портал находится в стадии тестирования и интеграции. Планируется осуществить практическое внедрение системы в период 2008–2009 гг., так как, во-первых, данный процесс сопряжен с существенными временными затратами, вызванными необходимостью перенесения большого объема информации из бумажных источников в подходящие электронные формы; во-вторых, с развитием технологий многоагентного программирования появляются новые привлекательные возможности расширения системы on-line взаимодействия пользователей.

Сотрудники и студенты учебного заведения, участвовавшие в тестировании программного продукта, отмечали удобство и привлекательность пользовательских интерфейсов, доступность и простоту использования сервисов. Пользователи высоко оценили такие программные компоненты, как рассылка новостей и форум, обеспечивающие своевременный доступ к наиболее актуальной информации и создающие прямые каналы связи между студентами и преподавательским составом.

Литература

1. Бабкин Э., Козырев О., Полухина О. Разработка информационных систем ERP-класса. – Н. Новгород: Нижегородский гос. техн.ун-т, 2006. – 266 с.
2. Вендров А.М., Проектирование программного обеспечения экономических информационных систем: учебник / Вендров А.М. – М.: Финансы и статистика, 2000. – 352 с.
3. Голенищев Э. Информационное обеспечение систем управления / Голенищев Э., Клименко И. – М.: Феникс, 2002. – 352 с.
4. Смирнова Г. Проектирование экономических информационных систем / Смирнова Г., Сорокин Ю., Тельнов Ю. – М.: Финансы и статистика, 2001. – 512 с.
5. Официальный сайт Министерства Образования и Науки [Электронный ресурс]: – http://www.mon.gov.ru, свободный. – Загл. с экрана. – Яз. рус.
6. Официальный сайт компании Naumen [Электронный ресурс]: – http://www.naumen.ru/go/solution/naumen_unversity, свободный. – Загл. с экрана. – Яз. рус.
7. Официальный сайт Российского Государственного Университета Путей Сообщения [Электронный ресурс]: – http://www.rgups.ru, свободный. – Загл. с экрана. – Яз. рус.
8. Официальный сайт системы OFBiz [Электронный ресурс]: – http://www.apache.org, свободный. – Загл. с экрана. – Яз. англ.
9. Официальный сайт системы Pentaho [Электронный ресурс]: – http://www.pentaho.com, свободный. – Загл. с экрана. – Яз. рус., англ.
10. Официальный сайт Якутского государственного технологического института [Электронный ресурс]: – http://www.yseti.ru, свободный. – Загл. с экрана. – Яз. рус.
11. Российская государственная библиотека [Электронный ресурс] / Центр информ. технологий РГБ; ред. Власенко Т.В.; Web-мастер Козлова Н.В. – Электрон, дан. – М.: Рос. гос. б-ка, 1997. – Режим доступа: http://www.rsl.ru, свободный. – Загл. с экрана. – Яз. рус., англ.

ОРГАНИЗАЦИЯ УПРЕЖДАЮЩЕГО КЭШИРОВАНИЯ В СЕТЕВОЙ СРЕДЕ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

И.Е. Сахаров (Костромской государственный технологический университет)
Научный руководитель – д.т.н., профессор В.Н. Шведенко
(Костромской государственный технологический университет)

В работе описана предложенная автором организация сервиса упреждающего кэширования данных на основе метода разделения потоков и результаты тестирования разработанной системы.

Введение

В работе представлена организация сервиса упреждающего кэширования в сетевой среде распределенных вычислений (ССРВ), которая позволяет использовать разнородные средства как единый вычислительный ресурс. При запуске задачи на выполнение могут возникнуть проблемы, связанные с тем, что исходные данные, необходимые для счета, располагаются в других компонентах ССРВ. Из большого территориального расположения компонент ССРВ возникает задержка передачи данных по каналам связи и задержка обращения к дисковым накопителям.

Основная проблема для получения выигрыша от ССРВ – сложность получения точных и своевременных предвыборок. Предлагается новый подход для генерации точных и своевременных предвыборок без участия программиста. Разница между временем вычисления одной процессорной операции и временем доступа к единице дисковой памяти постоянно увеличивается. Эта тенденция служит причиной того, что приложения должны осуществлять дополнительную выборку данных из дисковой памяти, до момента обращения к этим данным. Необходимость ускорения доступа к распределенным данным обуславливает актуальность задачи разработки упреждающего кэширования.

Существующие методы доступа к данным, основанные на примерах, истории и статистическом анализе, не позволяют решить задачу эффективного доступа к данным. Предлагаемый метод упреждающего кэширования, основанный на методе разделения потоков, позволяет решить эту задачу для большого множества различный приложений с интенсивным обменом данными. Основная идея заключается в разложении приложения на два потока: вычислительный поток, содержащий неизмененный код оригинальной программы, включающий все вычислительные операции и все операции ввода/вывода, и упреждающий поток, содержащий все оригинальные инструкции, которые имеют отношение к вводу/выводу.

Архитектура упреждающего кэширования

Упреждающий поток исполняется быстрее вычислительного потока, поэтому данные, к которым обращается вычислительный поток, всегда уже находятся в кэш-памяти, так как к ним уже было обращение при выполнении упреждающего потока. Данные помещаются в буферизированный файл упреждающего кэширования. Разработан транслятор, который осуществляет разделение на два потока и создание между ними канала, через который осуществляется взаимодействие и синхронизация выполнения обоих потоков.

Для многих приложений обычное кэширование является неэффективным, поскольку рабочая область данных, с которыми работает приложение, значительно превышает размеры кэш-файлов, используемых операционными системами. Всеми известный подход осуществления предвыборки данных до момента их реального использования используется во множестве проектов под операционными системами типа UNIX и

Linux [1]. В большинстве случаев предполагается, что требуемые данные располагаются последовательно друг за другом. Для большинства обычных приложений этот подход является эффективным, но для приложений с интенсивным обменом данными подобный подход неэффективен. Как правило, подобные приложения обращаются одновременно к большому количеству файлов, распределенных произвольным образом между множеством ВС. В работе предлагается использовать механизм упреждающего кэширования, который учитывает реальную последовательность обращений к файлам. Команды предвыборки получают требуемую информацию в процессе дополнительного запуска кода оригинальной программы.

Существуют ситуации, когда упреждающий поток не может сформировать упреждающие выборки. Например, когда имя требуемого файла является входящей переменной, или когда существуют внутри циклов зависимости по данным. В этой ситуации оба эти потока необходимо синхронизировать. Главная особенность упреждающего кэширования с применением двух потоков заключается в том, что исключается необходимость вручную изменять код программы. Все современные разработки принято делить на три направления:

Предсказуемое упреждающее кэширование. Самые ранние разработки систем упреждающего кэширования основывались на методах, основанных на примерах и методах основанных на истории обращений [2].

Специальные приложения, контролирующие упреждающие выборки и кэширование. Подобными системами занимался Patterson [3]. Такой подход повысил эффективность исполнения приложения от 13 до 30%. Также в этом направлении работал Cao [4]. Он первым предложил интегрировать упреждающее кэширование с обычным кэшированием.

Специальное компилирование приложений. Основная идея этого направления заключается в исключении роли разработчика приложения для использования упреждающего кэширования. T. Mowry разработал компилятор, который осуществлял специальные вставки в код приложения [5]. Chang и Gibson [6] разработали средство SpecHint, которое специальным образом изменяло откомпилированные файлы, что позволяло использовать так называемое спекулятивное выполнение задачи в моменты блокирующих операций ввода/вывода.

Системная архитектура

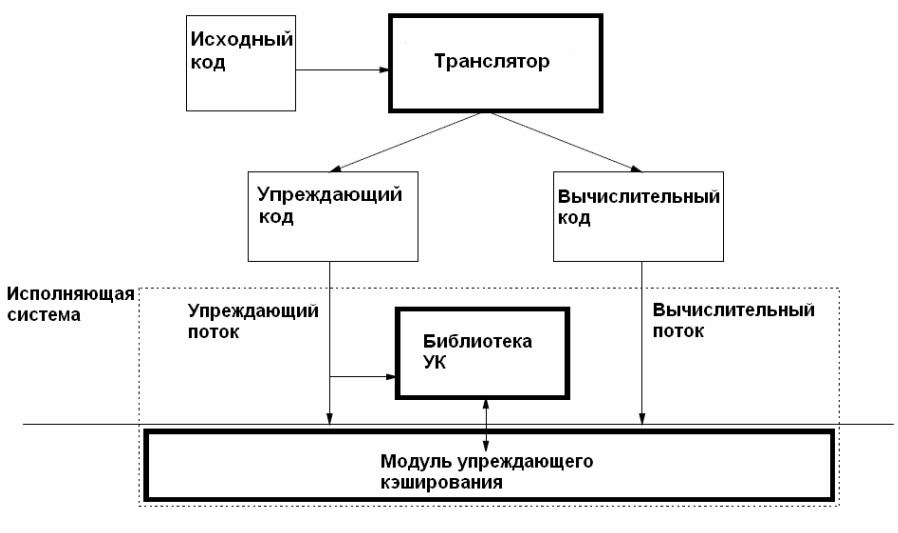


Рис. 1. Системная архитектура

Упреждающее кэширование основывается на четырех компонентах: транслятор исходного кода, библиотека упреждающего кэширования, загрузочном модуле и модуле упреждающего кэширования.

Транслятор кода создает упреждающий поток из приложения путем выборки тех частей кода, которые относятся к вводу/выводу и урезанию всех вычислительных частей. Все вызовы ввода/вывода заменяются вызовами из библиотеки упреждающего кэширования. Само исходное приложение формирует вычислительный поток. Имеется однозначное соответствие между упреждающими вызовами в упреждающем потоке и вызовами ввода/вывода из оригинального приложения. Упреждающий поток выполняется как Linux поток, используя стандартную библиотеку phtread. Каждый упреждающий вызов обслуживается загрузочным модулем. Каждому упреждающему вызову ставиться в соответствие логический адрес требуемого блока данных. Множество упреждающих вызовов заноситься в пользовательскую очередь упреждающих вызовов. Когда пользовательская очередь упреждающих вызовов заполняется, модуль упреждающего кэширования осуществляет системный вызов, который передает эту очередь на выполнение модулю упреждающего кэширования. Естественно эти вызовы могут и не выполниться, если например, не будет ресурсов для выполнения их, тогда модуль запустит их через некоторую паузу.

Основная сложность заключается в реализации синхронизации вычислительного и упреждающего потока. Синхронизация позволяет избежать отставания упреждающего потока от вычислительного потока. Необходимо осуществлять постоянную проверку файлов на предвыборку. Нужно точно знать, что подкачиваемые файлы будут использоваться в ближайшее время, а не использовались уже. Если условие не выполняется, то получается дополнительная работа и сильное уменьшение производительности выполнения реального приложения. Основная идея синхронизации строится на метках. Каждому упреждающему запросу ставится некоторый идентификационный номер ID соответствующего упреждающего вызова, исполняющего этот запрос. Модуль упреждающего кэширования также содержит номер ID текущей операции ввода/вывода вычислительного потока. Если номер записи в упреждающей очереди меньше, чем номер самого последнего вызова, то получается, что вычислительный поток обгоняет упреждающий, и тогда модулю упреждающего кэширования указывается пропустить все команды, поставленные для выполнения. Чтобы не получилось ситуации, когда упреждающий поток слишком опережает вычислительный поток, предусматривается некоторое расстояние N, на которое вычислительный поток может отставать от упреждающего потока. Это расстояние, например, можно привязать к среднему времени выполнения диском операции ввода/вывода, которую можно измерить заранее или просто к некоторому фиксированному числу.

Упреждающий поток

Рассмотрим основные операции упреждающего потока и их взаимодействия с вычислительным потоком, и связь с библиотекой упреждающего кэширования. На рис. 2 показано, как строится вычислительный и упреждающий поток.

Программный интерфейс упреждающего потока включает в себя четыре функции библиотеки:

(1) create_prefetch_thread(prefetch_function): эта функция позволяет приложению создать упреждающий поток и выполнить prefetch_fuction. Prefetch_function является входным параметром;

(2) prefetch_xxx(): это множество функций, которые заменяют в упреждающем потоке все стандартные функции ввода/вывода. Эти функции имеют такой же синтак-

сис, как и обычные функции ввода/вывода. Например, функция read(stream, ptr, size) заменяется функцией prefetch_read(stream, size);

(3) inform_open(file_pointer) и inform_close(file_pointer): эти функции необходимы, чтобы информировать упреждающий поток об открытии или закрытии файлов вычислительным потоком. Эти функции является необходимыми, поскольку модуль упреждающего кэширования составляет таблицу, содержащую две записи {file pointer, current offset} для всех файлов, использующихся в упреждающем потоке;

(4) synchronize(synchronization_point, type): эта функция синхронизации двух потоков. Аргумент type может принимать два значения: signal или wait.

	int fp;		
	Вычислительный поток		**Упреждающий поток (УП)**
	void main(void){		Void prefetch_fuction(void){
	int i; int data[100];		int I;
	/*создание упреждающего потока*/		
	create_prefetch_thread(		
	(void*)prefetch_function);		
##C1	/*открытие файла данных*/		
	fp=open("mydata.dat", O_RDONLY);		
	/*сигнал упреждающему потоку*/		
C2	synchronize(1, signal);		/*ожидание открытия файла*/
			synchronize(1, wait);
##C3			/*указание об открытии файла*/
		$$P1	inform_open(fp);
	for(i=99; i>=0; i--){		/*предвыборка*/
	/*ввод-вывод*/	$$P2	for(i=99; i>=0; i--){
C4	lseek(fp, i*4, SEEK_SET);		
	read(fp, &data[99-i], 4);	P3	/*только ввод-вывод*/
C5	/*вычисления*/		prefetch_lseek(fp,i*4,SEEK_SET);
C6	data[99-i] = data[99-i]*i;		prefetch_read(fp, 4);
	}	P4	
C7	/*закрытие файла*/	P5	
	close(fp);		}
	}		/*указание о закрытие файла*/
C8			inform_close(fp);
			}
		$$P6	

Рис. 2. Слева – вычислительный поток, справа – упреждающий поток. Внутри функции main строчки без ## являются оригинальным кодом приложения. В этот код были вставлены изменения по созданию упреждающего потока. Строчки без символа $$ были взяты с оригинального кода приложения. В упреждающем потоке не присутствуют вычисления

Синхронизация

Вычислительный поток и упреждающий поток функционируют независимо друг от друга до точки синхронизации. Все три следующих вида представляют собой точки синхронизации.

(1) *Открытие файла*. Упреждающий поток должен ожидать, пока вычислительный поток откроет файл. Потоки синхронизируются вызовом synchronize() с указанием номера синхронизационной точки. Вычислительный поток открывает файл и вызывает функцию synchronize для указания, что файл был открыт. Упреждающий поток вызывает функцию synchronize и ожидает получение сигнала от вычислительного потока. Линии C3 и P1 на рис.2 представляют собой точки синхронизации для двух потоков. Функция синхронизации использует функции pthread_cond_wait и pthread_cond_signal

из библиотеки pthread. Точки синхронизации обеспечивают запуск механизма упреждающего кэширования только для требуемых файлов.

(2) *Ввод информации пользователем*. Упреждающий поток вынужден ожидать, пока вычислительный поток вычислит адрес на основе входной информации(stdin) или считает его из специального файла. Если адрес следующих данных зависит от данных файла, для которого уже осуществляется предвыборка, тогда в каждый из потоков вставляются точки синхронизации. В этом случае упреждающий поток продолжит выполнение только после того, как вычислительный поток считает требуемую информацию из буфера кэш-памяти. Если входные данные считываются из конфигурационных файлов, тогда также добавляются точки синхронизации.

(3) *Чтение после записи*. Если программа содержит только вызовы чтения из одного файла и запись в разные файлы или операции чтения и записи не пересекаются для некоторого файла, тогда во всех этих случаях синхронизация не нужна, в остальных случаях синхронизация необходима. Предположим, что упреждающий поток выполняется правильным образом, т.е. идет быстрее вычислительного. В какой-то момент УП определил, что чтение некоторых данных зависит от предыдущих операций записи, то УП должен остановиться и подождать пока вычислительный поток не завершит требуемые операции записи. В такой ситуации необходима синхронизация.

Обмен данными между потоками

Оба потока могут обмениваться данными через глобальные переменные, используемые в приложении. Также необходимо, чтобы потоки могли самостоятельно обмениваться информацией. Для этого между потоками создается коммуникационный канал или просто канал pipe.

	Вычислительный поток (ВП)		**Упреждающий поток (УП)**
	void main(void){		Void prefetch_fuction(void){
	int fp; int data, int index;		int fp, int index;
	/* FILE *config_fp */		FILE *config_fp;
XX	create_prefetch_thread(		
##C1	(void*)prefetch_function);		
	fp=open(“mydata.dat”, O_RDONLY);		
C2	/*посылка дескпритора для УП*/		/*получение дескриптора*/
	send_fileptr(fp);		receive_fileptr(&fp);
##C3		$$P1	inform_open(fp);
	/*config_fp=fopen(“config.dat”,“r”);*/	$$P2	config_fp = fopen(“config.dat”, “r”);
XXC4		P3	/*ввод-вывод*/
	/*flseek(config_fp,10,SEEK_SET);*/		flseek(config_fp,10,SEEK_SET);
XXC5		P4	
			/*чтение и посылка ВП*/
	/*fscanf(config_fp, “%d”, &index);*/		send_fscanf(config_fp,“%d”, &index);
XXC6	/*ожидание данных от УП*/	P5	
	receive_fscanf(“%d”, &index);		
##C7	lseek(fp, index*4, SEEK_SET);		prefetch_lseek(fp,index*4,
C8		P6	SEEK_SET);
	read(fp, &data, 4);	P7	prefetch_read(fp, 4);
C9	data = data + 10;		
C10			
	close(fp);	$$P8	inform_close(fp);
C11	}		}

Рис. 3. Обмен данными между потоками. Локальные переменные fp и index должны быть доступны обоим потокам. Строки, помеченные XX, были изъяты из вычислительного потока и помещены в упреждающий поток. Строки, не помеченные $$, были взяты из оригинального кода приложения

Существует общий буфер для этих каналов, создается он в момент создания упреждающего потока. Обычно в качестве обменной информации выступают файловые дескрипторы и входные параметры. Существуют функции, обеспечивающие обмен данными между потоками.

(1) send_fileptp(file_pointer), receive_fileptr(file_pointer): эти две функции осуществляют пересылки файловых указателей между потоками. В эти функции включена синхронизация и не нуждаются в запуске дополнительной синхронизации (линии C3 и P1 на рис. 3).

(2) send_xxx() и receive_xxx(): это множество функций использующихся для передачи информации между потоками. Например, для функции fscanf(fp, "%d", &value) предусмотрены две функции send_fscanf(fp, "%d", &value) и receive_fscanf("%d", &value). Функция посылки send_fscanf считывает значение и посылает его в канал. Переменная fp для функции receive_fscanf не нужна, поскольку она считывается из канала. Для распознавания файлов конфигурации предлагается проверять имена файлов на наличие фраз conf, configure и config. На рис. 3 это показано крестиками. Вместо scanf в упреждающий поток вставляется функция send_fscanf, а в вычислительный поток вставляется функция receive_fscanf.

Внутренний анализ зависимостей функции

Цель анализа заключается в построении зависимостей внутри функции. В результате анализа определяются все переменные и выражения, относящиеся к дисковому вводу/выводу. Пусть P(x) – это некоторое множество выражений включающих в себя прямо или косвенно переменную x внутри конкретной функции. Используется достаточно простой подход.

(1) Все выражения, которые изменяют переменную x, включая определение этой переменной, включаются в множество P(x).

(2) Вычислим некоторое множество A, содержащее все переменные, используемые в выражениях из множества P(x).

(3) Для каждой a $\in$ A включим P(a) в P(x).

Для блочной структуры языка Си при выполнении шага 1 (построение множества P(x)) необходимо ограничиваться областью, где определена переменная x для функции. Например, рассмотрим фрагмент кода. Здесь 4 и 5 строки не должны включаться в множество P(x).

```
1.    void main(void){
2.        int i;
3.        i = 5;
4.        { int i;
5.          i = 6;
6.        }
7.        lseek(fp, i, SEEK_SET);
8.    }
```

Представленный алгоритм является простым и очень быстрым, но у него есть ряд недостатков. Алгоритм может включать много лишних выражений, которые никогда не заканчиваются операциями ввода/вывода или не влияют на операции ввода/вывода. Существующие компиляторы способны осуществлять проверку и убирать лишние части кода, которые не влияют на функционирования программы. Любой компилятор обладает мощным анализатором кода, который может найти и убрать неиспользуемые фрагменты. Поэтому предположим, что компилятор уберет лишнее. Тогда наш простой алгоритм становится достаточно эффективным.

Представленный алгоритм для определения множества P используется для анализа функции и построения упреждающего потока.

1. Все операции ввода/вывода в коде программы, которые относятся к файлам, к которым применимо упреждение, должны быть включены во множество операций, влияющих на ввод/вывод. Для каждой переменной x, встречающейся в выражениях дискового доступа к данным, строится множество P(x), и результат включается в УП.

2. Если в цикловой структуре, т.е. for, while, do-while, if-then-else или case, встречаются выражения, относящиеся к УП, то все используемые внутри переменные помечаются как влияющие на дисковые операции (помечаются с помощью массива булевых переменных). Для каждой такой переменной a строится множество P(a) и включается в УП. Эту операцию необходимо выполнить до конца.

3. Вставить в УП определения всех выражения и переменных, помеченных как влияющие на дисковые операции ввода/вывода.

4. Вставить в УП вызовы синхронизации(synchronize), посылки(send_xxx) и приема(receive_xxx) для передачи данных между потоками.

5. Если функция не содержит операций дискового ввода вывода после того, как пройден весь алгоритм, то необходимо убрать эту функцию из УП.

Этот алгоритм запускается итеративно для каждой процедуры до тех пор, пока не дойдет до конца приложения.

Внешний анализ зависимостей функций

Для создания правильного упреждающего потока, кроме внутреннего анализа функций, необходимо произвести межфункциональный анализ зависимостей. Анализ построен на поиске информации о входных переменных функций, возвращаемых значениях и глобальных переменных.

1. Для каждой функции T пусть $Q(x_1, x_2, \ldots, x_n)$ – это одна из процедур, которые вызывает T с параметрами $(y_1, y_2, \ldots, y_n)$. Если любой из параметров y_i является зависимым для дискового ввода/вывода в функции T(т.е. значение переменной может быть изменено в функции Q), то это указывает, что переменная x_i внутри процедуры Q, также влияет на дисковый ввод/вывод. Если функция T сохраняет возвращаемое значение функцией Q в некоторую переменную a, то a также влияет на дисковый ввод/вывод в функции T.

2. Для каждой процедуры $T(y_1, y_2, \ldots, y_n)$ пусть R – это функция, которая вызывает T с параметрами $T(z_1, z_2, \ldots, z_n)$. Если параметр y_i влияет на дисковый ввод/вывод в функции T, то это означает, что параметр z_i также влияет на дисковый ввод/вывод в R.

3. Все глобальные переменные, влияющие на ввод/вывод во всех функциях, также влияют на ввод/вывод.

Выше представленный алгоритм применяется для каждой функции лишь один раз, если нет рекурсивного вызова. Если такие вызовы есть, то алгоритм применяется до тех пор, пока не закончится выполнение.

Ограничения

Существуют ограничения на построение кода УП. Для многопотоковых приложений возникают существенные сложности, так как в одной функции заложено выполнение для множества узлов. Также необходимо учитывать передаваемую информацию между потоками. Реализовать это можно посредством усложнения процесса анализа кода. Предлагается из кода многопотокового приложения строить код для однопоточного приложения и граф зависимостей между этими потоками. Сейчас автор занимается разработкой анализатора для многопотокового приложения. Главная задача – построение правильного графа зависимостей этих потоков.

Понятно, что основная задача исследования состоит в создании упреждающего кэширования для ССРВ. Разработанный модуль работает только для конкретного узла, а стоит задача осуществить запуск упреждающего кэширования для многопотокового приложения, в частности, для приложений MPI. Ставится задача построения анализатора, который создавал бы упреждающий поток, сохраняя всю структуру взаимодействия MPI. Основной проблемой является организации правильного взаимодействия между множеством вычислительных потоков и множеством упреждающих.

Архитектура исполняющей системы

Библиотека упреждающих вызовов

Все вызовы упреждения, которые использует упреждающий поток, содержатся в библиотеке упреждающих вызовов. На самом деле это загрузочный модуль для работы упреждающего кэширования, который использует библиотеку упреждающего кэширования, а также размещает и хранит в памяти очередь команд на упреждение и адреса блоков, которые должны быть упреждены. Структура очереди команд на упреждение имеет следующий вид: {идентификатор файла, номер блока, идентификатор вызова}. Идентификатор файла и блока определяют реальный физический блок данных, который необходимо предвыбрать. Идентификатор вызова определяет вызов, который инициировал запуск механизма упреждения для данного блока. Начальное значение идентификаторов вызовов начинается с 0. Файловая таблица для каждого файла дескриптора содержит смещение. Используя эту информацию, всегда можно вычислить номер блока для вызовов prefetch_read и prefetch_lseek.

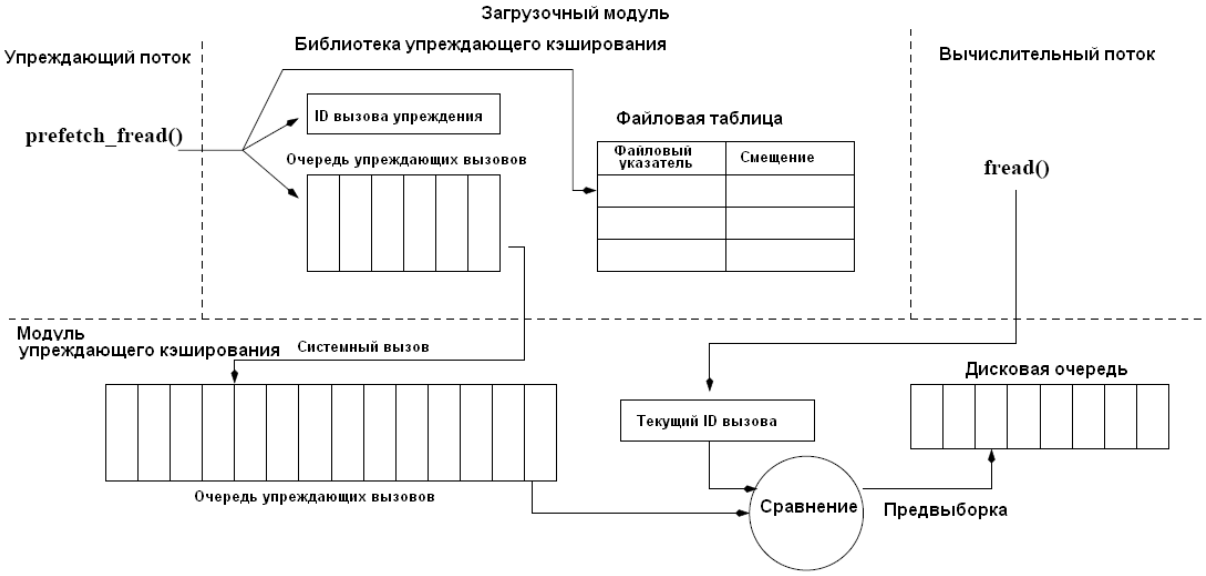


Рис. 4. Архитектура исполняющей системы

Каждому вызову ставится в соответствие идентификатор этого вызова. Потом счетчик идентификаторов вызовов увеличивается на 1 (рис. 4). Далее ID вызова помещается в очередь на предвыборку. В конце происходит обновление текущего смещения в файле. Как только очередь упреждающих вызовов заполняется, она перемещается в очередь модуля упреждающего кэширования на исполнение. Когда приложение выполняет функцию create_prefetch_thread(), то загрузочный модуль, используя вызов fork() запускает поток упреждения и передает модулю упреждающего кэширования идентификаторы вычислительного и упреждающего потоков. Загрузочный модуль так-

же регистрирует указатели на файлы, которые требуют упреждения, и передает информацию модулю упреждающего кэширования.

Модуль упреждающего кэширования (ядро системы упреждения)

Основная задача модуля состоит в запуске процессов копирования и постоянной проверке: упреждающий поток должен быть всегда впереди вычислительного на определенное количество команд. Когда загрузочный модуль определяет, что очередь запросов на упреждение заполнилась, то она передается модулю упреждающего кэширования на исполнение. Если очередь модуля упреждающего кэширования заполнена, то загрузочный модуль делает паузу в передаче своей очереди. Как только модуль упреждающего кэширования освободит свою очередь, он передает сигнал загрузочному модулю, и тот передает остановленные ранее вызовы. Передача осуществляется, когда очередь в загрузочном модуле полностью заполнится. Размер очереди модуля упреждающего кэширования больше, чем размер очереди загрузочного модуля.

Получается следующая модель исполнения приложения: в момент, когда вычислительный поток производит обращение к некоторым данным, эти данные уже находятся в файловом буфере, так как к ним уже было осуществлено обращение при исполнении упреждающего потока. Таким образом, уменьшается количество блокирующих операций ввода/вывода, и тем самым увеличивается эффективность выполнения приложения. Подобная схема эффективно работает для приложений с периодическими постоянными вызовами доступа к данным. Для упреждающего потока выставляет более высокий приоритет по сравнению с вычислительным потоком. Перед тем, как исполнить любой запрос из очереди модуля упреждающего кэширования, происходит сравнение идентификаторов вызовов упреждающего и вычислительного потока. Если идентификатор упреждающего потока меньше, чем идентификатор вычислительного, то модуль просто удаляет этот вызов из очереди и переходит к выполнению следующего.

Пусть K – текущий размер очереди модуля упреждающего кэширования, T – среднее время исполнения одного запроса из очереди, а среднее время вычислений между двумя дисковыми вызовами пусть будет C. Предположим, что t – текущее время, текущий идентификатор вызова вычислительного потока будет i, а упреждающего потока будет j. Тогда время, до момента, когда приложение обратится к некоторому блоку, будет равняться C*(j – i). Упреждающий вызов, который посылается в дисковую очередь на выполнение во время t, будет исполнен ко времени t+(K+1)*T. Модуль упреждающего кэширования должен выполнить операцию упреждения, только если выполняются следующие условия:

(1) $(C*(j - i)) - ((K+1)*T) \leq$ временного предельного значения;

(2) $(j - i) \leq$ предельного ограничения дисковой очереди.

Первое неравенство определяет, что подкачиваемый дисковый блок еще будет использоваться. Второе равенство определяет, что в буферной кэш-памяти есть еще место под следующие блоки. Обе эти предельные величины определяются текущей аппаратурой и пользователем и должны быть заданы заранее.

Тестирование

Использовались следующие тестовые приложения.

(1) Подсчет некоторой характеристики для двух матриц размером 512×512×512, где каждый элемент равняется байту. Общий размер каждой матрицы составлял 130 Гбайт. Каждая матрица была поделена на 4 файла по 33 Гбайт и размещена на двух диска по 2 файла на каждом. Исполнялась задача расчета характеристики.

(2) Получение новой матрицы из имеющихся по следующему принципу: пусть A и B – исходные матрицы, A(x,y,z) и B(x,y,z) – элементы исходных матриц, тогда элемент новой матрицы C рассчитывается следующим образом:

C(x,y,z) = B(x,y,z)×A(511-x,511-y,511-z) ×B(x+a,y+a,z+a)/ A(x+b,y+b,z+b).

Размер блока был равен 4 Кб. По результатам тестирования были получены следующие результаты. Для первого случая уменьшение времени выполнения приложения с применением упреждающего кэширования составило порядка 41,23%, для второго случая – порядка 23,99%.

Заключение

В статье представлена организация упреждающего кэширования на основе метода разделения потоков. Для доказательства эффективности введения упреждающего кэширования была разработана имитационная модель упреждающего кэширования для ССРВ в пакете GPSS[7]. Разработанный метод упреждающего кэширования, основанный на методе разделения потоков, позволяет решить задачу определения последовательности будущих обращений к файлам для большого множества различный приложений с интенсивным обменом данными. Основная идея заключается в разложении приложения на два потока: вычислительный поток, содержащий неизмененный код оригинальной программы, включающий все вычислительные операции и все операции ввода/вывода, и упреждающий поток, содержащий все оригинальные инструкции, которые имеют отношение к вводу/выводу.

Дальнейшими направлениями работы являются разработка системы упреждающего кэширования для многопроцессорных приложений и внедрение разработанной системы в различные ССРВ. Также необходимо, чтобы система упреждающего кэширования учитывала реальное расположение файлов по всей ССРВ. Планируется внедрение представленного материала в учебный процесс.

Литература

1. M.K. McKusick. A Fast File System for UNIX // ACM Transaction on Computer System, 2(3), August 1984.
2. K. Curewitz. Practical Prefetching via Data Compressing // ACM Conference on Management of Data, May 1993.
3. R. Hugo Patterson, Garth A. Gibson, M. Satyanarayanan. A Status Report on Research in Transparent Informed Prefetching // ACM Operating Systems Review, 1993.
4. Kieran Harty, David R. Cheriton. Application-controlled physical memory using external page-cache management / Proceedings of the fifth international conference on Architectural support for programming languages and operating systems, 1992.
5. Todd C. Mowry, Angela K. Demke, Orran Krieger. Automatic Compiler-Inserted I/O Prefetching for Out-of-Core Applications / Proceedings of the 1996 Symposium on Operating Systems Design and Implementation.
6. Fay Chang, Garth A. Gibson. Automatic I/O Hint Generation through Speculative Execution / Operating Systems Design and Implementation, 2001.
7. Киселев А., Корнеев В., Семенов Д., Сахаров И. Управление метакомпьютерными системами // Открытые системы. – 2005. – №2.

СЕМАНТИЧЕСКИЙ АНАЛИЗ ПРОЕКТНОЙ ДОКУМЕНТАЦИИ

С.В. Клименков, А.Н. Максимов, А.Е. Харитонова
Научный руководитель – к.т.н., доцент А.А. Приблуда

Рассматриваются вопросы применения семантического анализа текстовой информации в задачах автоматизации проектирования информационных систем (ИС) с использованием компонентов уже готовых ИС.

Введение

Одной из важнейших частей сегодняшнего мира являются различные ИС – сложные программно-аппаратные комплексы, без которых функционирование многих предприятий было бы невозможно. Создание таких комплексов является нетривиальной и дорогостоящей задачей. Одним из способов ускорить разработку и сократить расходы на создание новых ИС является повторное использование архитектурных решений и компонент уже существующих ИС.

Как создание новой ИС, так и анализ функционирующих ИС с целью повторного использования их отдельных компонент требуют многомесячной работы группы специалистов, если делать это без использования средств автоматизации. Сокращение стоимости и сроков разработки может быть достигнуто за счет программно выполняемого анализа текстовой документации создаваемых ИС и сопоставления частей этой документации объектным структурам существующих проектов ИС.

Постановка задачи

В качестве базовой информации для работы с текстовыми данными ИС предполагается использовать семантическое описание технического задания (ТЗ). Семантические описания можно получить несколькими способами. Один из них – создать семантическое описание самостоятельно, используя широко распространенные средства простого лингвистического анализа.

Другим способом является использование уже имеющихся средств лингвистической направленности, которые позволяют получить готовые семантические описания. Эти семантические описания, в свою очередь, должны быть адаптированы для задач автоматизации проектирования информационных систем.

На определенном этапе исследований встал вопрос выбора способа получения семантических описаний проектной документации. Была поставлена задача, рассмотреть различные способы и исследовать возможность их применения для получения семантических описаний ТЗ. Приведем краткую характеристику основных возможностей, предоставляемых рассматриваемыми способами построения семантических описаний.

Использование простых лингвистических программ

Этот способ предполагает построение семантического описания текста на основе анализа результатов обработки текста простыми лингвистическими программами.

В рамках исследования данного способа был создан прототип лингвистического процессора для системы автоматизированного проектирования. Основной проблемой при его создании являлось получение семантического описания, в котором единичный элемент должен представлять собой группу слов-существительных, имеющих сходное значение. Поэтому в прототипе для реализации ряда функций лингвистического процессора использовались упрощенные алгоритмы. Алгоритмы были упрощены как с математической, так и с лингвистической точки зрения. Общая структура алгоритма лингвистического процессора такова.

1. Синтаксическо-статистическая обработка текста. На этом этапе составляется распределение существительных в обрабатываемом тексте.
2. Синтаксическая обработка. С помощью функций библиотек программы «Linkparser» [1] в тексте определяются подлежащие, сказуемые и дополнения. Полученные результаты дают возможность сделать предположение о том, какие слова несут ключевую информацию о тексте. На этом же этапе происходит формирование простейшей семантической структуры – концепта с единичным содержанием.
3. Семантическая обработка. Происходит расчет взаимосвязей концептов между собой, т.е. строится матрица связей. Степень сходства слов рассчитывается на основе установления сходства хотя бы 2-мя из 4-х различных методов [2]. На основе этой матрицы концепты объединяются.

В упрощенном алгоритме в недостаточной степени учитывалась корреляция всех используемых слов из текста. На начальных этапах разработки алгоритма сходные слова группировались вокруг какого-либо одного слова, выбираемого случайно. В результате тестирования оказалось, что в группах, сформированных таким образом, возможны противоречия. При наличии связи с основным словом концепта рассматриваемое слово могло быть недостаточно связано с остальными присоединенными или вообще противоречить им, т.е. не иметь никакой связи с основным словом концепта.

Основным достоинством данного способа является то, что получаемое семантическое описание полностью соответствует специфическим требованиям решаемой задачи.

В качестве недостатка данного способа следует отметить то, что получение семантического описания довольно трудоемко без специальных лингвистических знаний и поэтому может недостаточно полно соответствовать реальной семантической модели текста. Кроме того, существующий прототип лингвистического процессора работает только с текстами на английском языке.

Использование готовых средств семантического анализа

Данный способ позволяет избежать ошибок в формировании семантического описания, вызванных недостатком лингвистических знаний. Среди всего многообразия существующих средств, позволяющих производить семантический анализ на русском языке, два средства выглядят наиболее перспективными для использования их в задачах автоматизации проектирования. Это системы «Диалинг» [3] и «Fact Extractor» [4].

Пакет «RCO Fact Extractor SDK» – это комплексный инструментарий для разработки информационно-поисковых и аналитических систем, требующих лингвистического анализа текста на русском языке. Ядро пакета представляет библиотека RCO FX Ru, при помощи которой осуществляется полный синтактико-семантический разбор русского текста. Существует возможность автоматически анализировать содержание текстовых документов, представляя его в форме ассоциативной семантической сети.

Ассоциативная семантическая сеть представляет собой ориентированный граф, вершинами которого служат значимые темы, выделенные в анализируемом тексте, а дугами – связи между ними. С каждой вершиной связаны вес (значимость) и частота упоминания темы, а с каждой дугой – вес (сила) связи и частота появления связи в тексте. Одной из его основных возможностей является построение полного семантического описания текстового документа.

В системе «Диалинг» для получения семантического описания исследуемый текст обрабатывается последовательно несколькими модулями досемантического анализа.

После обработки текста в системе «Диалинг» получаем семантические описания каждого отдельного предложения. Например, для предложения «Дом Пашкова стоял на пригорке» семантическая структура представлена на рис. 1.

ПРИНАДЛ (Пашков,дом)

П-АКТ(дом,стоял)

ЛОК (пригорке,стоял)

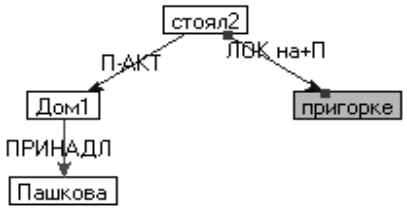


Рис. 1. Пример семантической структуры предложения

Описание строится в виде графа, вершинами которого являются семантические узлы, а дугами – семантические отношения. Семантические узлы, помимо слов, могут включать знаки препинания, устойчивые обороты, жесткие синтаксические группы. Тип семантического отношения выбирается из фиксированного набора отношений, используемых в системе, всего может быть использовано около 30 типов отношений.

К основным достоинствам системы «Диалинг» можно отнести открытость программного интерфейса, на основе которого можно создавать собственные алгоритмы семантического анализа.

Система «Диалинг» позволяет получить семантическое описание только одного предложения. В связи с этим остается нерешенной задача непротиворечивого семантического описания всего текста, предполагаемым способом решения которой может быть построение гиперграфа, отражающего семантическую структуру всего текста, с последующим его анализом.

Практическое использование системы «Диалинг»

Система автоматического анализа текста «Диалинг» была выбрана в качестве основного инструмента для исследования возможности построения семантического описания. С помощью этой системы предполагается строить семантические описания технических заданий и проектной документации для дальнейших исследований.

В ходе исследований был обработан ряд документов с целью изучения возможности объединения семантических представлений предложений, получаемых с помощью системы Диалинг, в единую семантическую структуру всего документа. В качестве исходного материала была использована техническая документация по проектам, выполняемых студентами в рамках дисциплины «Проектирование информационных систем».

В результате первичного анализа выяснилось, что только часть предложений из документа несет в себе непосредственную информацию об архитектуре описываемой системы. Предложения, не несущие в себе информации, во всех рассматриваемых проектах оказались практически идентичными, в связи, с чем появилась возможность составить словарь «избыточных данных проектной документации». В этот словарь будут входить словосочетания, устойчивые выражения и предложения, которые не отражают архитектуру создаваемого проекта. Данный словарь будет создаваться и пополняться в процессе анализа набора технических заданий.

При дальнейшем анализе полученных семантических структур и сопоставлении их с фрагментами архитектурных решений реализованных проектов удалось однозначно сопоставить требования к ИС, описанные в ТЗ на естественном языке, с реализацией этих требований. В частотности, четко прослеживалось соответствие между семантическими структурами ТЗ и такими элементами реализованного проекта, как диаграммы

use-case, инфологическая модель использованной базы данных и отдельные классы объектного представления.

В качестве простого примера можно рассмотреть описываемую ниже модель. В техническом задании к информационной системе для тренинг-центра существует требование – «Подача заявки на обучение». Предложения, которые описывают общий механизм «движения заявок», представлены так: «Заинтересованный клиент подает заявку на обучение» и «Тренер получает заявку на проведение занятий».

Семантические описания этих предложений представлены на рис. 2.

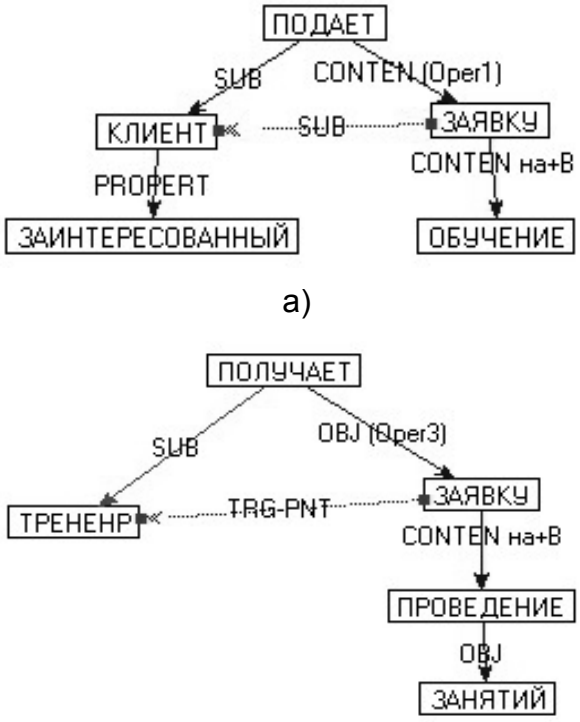


Рис. 2. Семантические описания требований ТЗ: а) «Заинтересованный клиент подает заявку на обучение»; б) «Тренер получает заявку на проведение занятий»

Непосредственные действующие лица – «тренер» и «клиент» имеют прямые связи с понятием «заявка». В рассматриваемой системе класс «Заявка» выделен разработчиком в абстрактный класс, которой является супер-классом всех видов заявок, фигурирующим в системе. Таким образом, можно предположить, что связи SUB и TRG-PNT, применимые к одному и тому же понятию (в данном случае «заявка»), могут служить предпосылкой для создания абстрактного класса или его шаблона, отображающего это понятие.

Подобная зависимость прослеживается при появлении таких абстрактных классов, как «оценка», «программа курса» и т.п.

Заключение

Из нескольких возможных способов построения семантических моделей проектной документации ИС для последующих исследований была выбрана система автоматического анализа текста «Диалинг».

В результате исследований удалось получить однозначные соответствия между отдельными семантическими структурами ТЗ и элементами реализованных ИС. Полученные результаты свидетельствуют о возможности и перспективности применения

системы «Диалинг» для семантического анализа проектной документации в системах автоматизированного проектирования ИС.

Литература

1. LinkGrammar parser [Электронный ресурс]. Режим доступа: http://www.link.cs.cmu.edu/link/, свободный. – Загл. с экрана. – Яз. англ.
2. Кириченко К.М., Герасимов М.Б. Обзор методов кластеризации текстовой информации, конференция по компьютерной лингвистике в России – Москва, 2001.
3. Сокирко А.В. Семантические словари в автоматической обработке текста: Дис. … канд. техн. наук. – М., 2003. – 56 с.
4. Ермаков А.Е. Поиск фактов в тексте // Мир ПК. – 2005. – № 2.

ОРГАНИЗАЦИЯ ТЕЛЕМЕДИЦИНСКОГО ЦЕНТРА НА БАЗЕ ТАМБОВСКОГО ГОСУДАРСТВЕННОГО ТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

М.А. Лядов, Д.Н. Труфанов, С.Вл. Фролов, М.С. Фролова
(Тамбовский государственный технический университет)
Научный руководитель – д.т.н., профессор С.В. Фролов
(Тамбовский государственный технический университет)

В Тамбовском государственном техническом университете на кафедре «Биомедицинская техника» ведется разработка новой организационно-технической структуры региональной телемедицинской сети и широкополосной, симметричной системы приема-передачи медицинской информации по IP сетям.

Введение

Термин *телемедицина* может быть определен как способ обеспечения врачей и организаторов здравоохранения современными информационно-телекоммуникационными средствами там, где расстояние является критическим фактором. Российская Федерация, как никакая другая страна мира, нуждается во врачебных телемедицинских консультациях. Несомненно, что экономически выгоднее внедрять технологию телеконсультаций, чем развивать традиционную инфраструктуру здравоохранения в отдаленных, малонаселенных районах. К тому же продолжает оставаться неудовлетворенность населения работой поликлиник – отсутствием врачей нужной специальности, низкой квалификацией медицинского персонала, слабо развитой сетью специализированных больничных коек, крайне медленным решением проблемы приближения медицинской помощи, оказываемой сельскому населению к уровню города, возможностей доступа городского и сельского населения к крупнейшим федеральным медицинским центрам.

Сейчас ситуация с телемедициной в России выглядит так, что более чем в 30 областях и республиках созданы и функционируют телемедицинские центры: региональные или республиканские. На их базе проводятся телеконсультации больных и телеобучение врачей, которых консультируют и обучают специалисты из медицинских центров Москвы и Санкт-Петербурга. Телемедицинские технологии активно применяются в Архангельской, Ленинградской, Нижегородской, Пензенской, Воронежской областях, в Якутии, Хабаровском крае, Татарии, Башкирии и Чувашии.

Однако до настоящего времени в России не создана эффективная организационно-техническая структура телемедицинской сети. Существующие системы телемедицины основаны на технологии передачи данных ISDN, которые построены на существующих телефонных сетях. Каналы связи ISDN требуют высоких затрат. В системах ISDN используется дорогостоящее оборудование. Организацию телемедицинских центров на основе технологии ISDN могут позволить себе только крупные лечебно-профилактические учреждения (ЛПУ). Например, в Тамбовском регионе только на базе Тамбовской областной и Тамбовской областной детской больниц работают телемедицинские центры, в которых установлены комплексы на основе ISDN. Здесь связь может быть установлена только с крупными федеральными медицинскими центрами.

Развернуть телемедицинскую сеть в районах области на базе этого оборудования из-за высокой стоимости каналов связи или их отсутствия невозможно. В мировой практике современные телемедицинские технологии также основаны на технологии ISDN, что затрудняет использование существующих зарубежных систем в системе региональной телемедицины с имеющимися ограниченными ресурсами. С другой стороны, имеющиеся региональные телемедицинские сети, например, система телемедицины Пензенской области, основана на передаче медицинской цифровой информации с использованием обычных телефонных модемов в режиме off-line.

Низкая скорость модемной связи не вызволяет проводить сеансы видеоконференцсвязи и передавать медицинские изображения в цифровой форме. Таким образом, актуальной задачей в региональной телемедицине является переход к технологиям IP (Интернет).

В проекте предлагается разработка системы региональной телемедицины с использованием современных технических возможностей информационно-телекоммуникационных систем на основе комбинирования технологий оптоволоконных сетей, технологии ADSL и спутникового Интернета, которая предоставляет доступ к телемедицине врачам практически любого медицинского учреждения в Тамбовском регионе в условиях ограниченности финансовых и технических ресурсов.

Постановка задачи

В настоящее время житель глубинки не всегда может в полной мере реализовывать свое право на получение квалифицированной медицинской помощи, так как уровень диагностики и лечения в районных больницах значительно ниже, чем в крупных медицинских центрах города, городское и сельское население имеет трудности доступа к крупным федеральным медицинским центрам. Решением таких проблем является развитие региональной телемедицины.

Насущной необходимостью для развития региональной медицины является внедрение телемедицинских технологий на уровне районных, городских и даже поселковых больниц. Здесь телемедицина принесет максимальный социально-экономический эффект. Предлагается проект телемедицинской сети Тамбовского региона, внедрение которой должно обеспечить на постоянной основе проведение телемедицинского консультирования как медицинских работников районных ЛПУ силами врачебного персонала областных клиник, так и врачей областных больниц и поликлиник специалистами федеральных медицинских центров.

Телемедицинская сеть может также быть использована для дистанционных курсов повышения квалификации медицинских работников с выдачей соответствующих сертификатов. Большой проблемой для здравоохранения Тамбовской области является отсутствие в районах квалифицированных рентгенологов. Поэтому актуальной является задача оперативной передачи по каналам связи оцифрованных рентгенограмм. Такую задачу будет успешно решать региональная телемедицинская сеть.

Таким образом, создание региональной телемедицинской сети решает следующие задачи:

- приближение квалифицированной медицинской помощи к ЛПУ к больному;
- повышение качества врачебных консультаций в ЛПУ Тамбовского региона;
- повышение качества здравоохранения в Тамбовском регионе;
- повышение охвата и доступности последипломного обучения врачей Тамбовского региона.

В предлагаемом проекте телемедицинской сети Тамбовского региона научно-технической новизной является разработка для целей телемедицины новой широкополосной, симметричной системы приема-передачи медицинской информации по IP сетям на основе комбинирования технологий оптоволоконных сетей, технологии ADSL и спутникового Интернета. Все существующие в России телемедицинские сети используют либо проводные технологии передачи данных (например, система телемедицины Пензенской области), либо спутниковые каналы связи. Данные системы связи имеют недостатки: высокую стоимость или низкую скорость передачи данных. Предлагаемые комбинированные системы телекоммуникаций связи обеспечивает оптимальное соотношение цены и качества. Этот инновационный и актуальный способ связи будет реализован в Тамбовской региональной телемедицинской сети.

Пилотным проектом региональной телемедицинской сети станет реализация консультационного центра в Тамбовском областном онкологическом диспансере. Консультирование будет осуществляться по направлению онкология. Отсутствие онкологов в районных больницах Тамбовского региона делает внедрение такого консультационного центра актуальным. При отложенных телеконсультациях история болезни пациента, включая текстовое и графическое описание диагностических исследований в цифровом виде, отправляется врачу консультационного ТМЦ Тамбовского областного онкологического диспансера по электронной почте для консультации. Связь в таком случае будет устанавливаться с ЛПУ крупнейших городов области – Мичуринской центральной городской больницей (ЦГБ), Моршанской центральной районной больницей (ЦРБ), а также с ЛПУ районных центров, например, с Петровской ЦРБ.

В процессе лечения пациент, живущий в районе, и его лечащий районный врач могут систематически получать консультацию у врачей Тамбовского областного онкологического диспансера. Центральный ТМЦ будет проводить консультации по онкологии в режиме реального времени. Сеансы видеоконференцсвязи ТМЦ ТГТУ будет устанавливать с крупными федеральными и зарубежными медицинскими центрами.

В проекте предполагается создание на базе ГУЗ «Тамбовское областное патологоанатомическое бюро» информационной системы для хранения и анализа результатов морфологических исследований операционного и биопсийного материала. Кроме проведения патологоанатомических вскрытий, в обязанности патологоанатомической службы входит морфологические исследование (микроскопия) операционного и биопсийного материала поступающего практически изо всех ЛПУ как города, так и области. Количество исследований и количество больных, которые подвергаются такому виду исследований, ежегодно увеличивается и составляет более 30 тысяч больных в год. В бюро существует локальная компьютерная сеть с базой данных на выделенном сервере, куда входят данные более 400 тысяч больных, которым проводились те или иные морфологические исследования с 1992 г.

В морфологических исследованиях (микроскопии) операционного и биопсийного материала зачастую встает вопрос о консультациях микропрепаратов у более квалифицированных специалистов, работающих в крупных научных медицинских учреждениях. Обеспечение данных консультаций ложится на плечи больных или родственников, которые должны с микропрепаратами ехать в другой город, чтобы уточнить диагноз. Поэтому в бюро возможно создание банка данных фотоизображений, полученных в результате микроскопирования препаратов, для дальнейшего их использования в работе, и в частности – консультаций данных изображений с использованием телемедицинских технологий.

Для реализации данных идей требуется:

- создание сервера для хранения фотоархива;
- приобретение микроскопа с цифровой фотокамерой и программным обеспечением для нее;
- реализация канала быстрого доступа к сети Интернет для соединения с сетями консультирующих учреждений.

Важным элементом сети является унифицированная телемедицинская станция для региональной телемедицины. Станция должна обеспечивать возможность хранить и передавать по каналам связи в цифровом виде: историю болезни, текстовое описание лабораторных исследований, рентгеновские снимки, результаты УЗИ и другую графическую информацию, фотографии пациента и поврежденных участков. При всем многообразии возможностей станция должна быть проста в освоении и иметь интуитивно понятный интерфейс. Необходимо учесть возможность подключения дополнительного диагностического оборудования (кардиографа, спирографа и т.п.). Кроме того, телеме-

дицинская станция должна иметь низкую стоимость. В рамках проекта предлагается разработка такой станции.

В предлагаемом проекте будет отлажена система связи телемедицинской станции, установленной в районном ЛПУ, с ТМЦ города Тамбова по IP сетям на основе комбинирования технологий оптоволоконных сетей, технологии ADSL и спутникового Интернета. Будут предложены новые методы организации комбинированной связи по IP сетям в условиях ограниченности возможностей региональных каналов связи. На базе ТМЦ ТГТУ в рамках пилотного проекта планируется также организация дистанционных курсов повышения квалификации врачей. Основной вариант применения технологий дистанционного образования в ТМЦ ТГТУ – телелекции, во время которых преподаватель использует преподавательскую систему, чтобы с помощью сети Интернет и ученических систем дать ученикам, находящимся в ТМЦ ТГТУ, требуемый материал и проверить, насколько хорошо он усвоен. Очевидно, что дистанционные образовательные технологии следует преимущественно применять на теоретических циклах повышения квалификации медицинских специалистов, однако сдачу экзаменов целесообразно проводить традиционным способом: тестирование, проверка навыков, собеседование.

Развитие в ТМЦ ТГТУ образовательного направления (дистанционных обучающих технологий) необычайно важно, так как оно органично вписывается в общую концепцию повышения качества оказания медицинской помощи. Несомненно, что дистанционное образование в медицине позволяет существенно повысить уровень профессиональных знаний врачей без отрыва от рабочих мест или дает возможность сократить сроки очной части обучения без потери его качества на фоне повышения охвата и доступности последипломного обучения специалистов. Таким образом, предлагаемый проект предусматривает создание новой организационно-технической структуры региональной телемедицинской сети в условиях ограниченных ресурсов и разработку для целей телемедицины новой широкополосной, симметричной системы приема-передачи медицинской информации по IP сетям на основе комбинирования технологий оптоволоконных сетей, технологии ADSL и спутникового Интернета.

Методы исследований

Телемедицинскую сеть Тамбовской области предлагается создать по трехуровневой схеме. На первом уровне будут работать телемедицинские пункты, развернутые в районных ЛПУ. На втором уровне – консультационные телемедицинские центры (ТМЦ) на базе ЛПУ Тамбова. На третьем – центральный ТМЦ, который обеспечит работу всей телемедицинской сети и связь с федеральными и зарубежными медицинскими центрами, а также с центральными ЛПУ других регионов. ТМЦ ТГТУ будет являться организационно-правовым центром в этой информационной системе (рис. 1).

Все участники телемедицинской сети входят в единую информационную систему. Это подразумевает наличие:

- единой БД;
- единого файлового хранилища;
- единую систему доступа к данным – Интернет-портала и специального ПО;
- четкого разграничения доступа;

Основными каналами связи в региональной телемедицинской сети станет IP-коммутация (Интернет). В центральный ТМЦ будут приглашаться специалисты из ЛПУ г. Тамбова, не имеющие своего консультационного ТМЦ, по специально разработанной организационной системе для проведения телеконсультаций как с медицинскими работниками районных ЛПУ, так и с федеральными медицинскими центрами. Центральный ТМЦ создан на территории ЛПУ (поликлиника № 6 и санаторий-

профилакторий «Тонус») Тамбовского государственного технического университета (ТГТУ). ТМЦ ТГТУ обеспечит техническую, организационную и финансовую составляющие работы региональной телемедицинской сети.

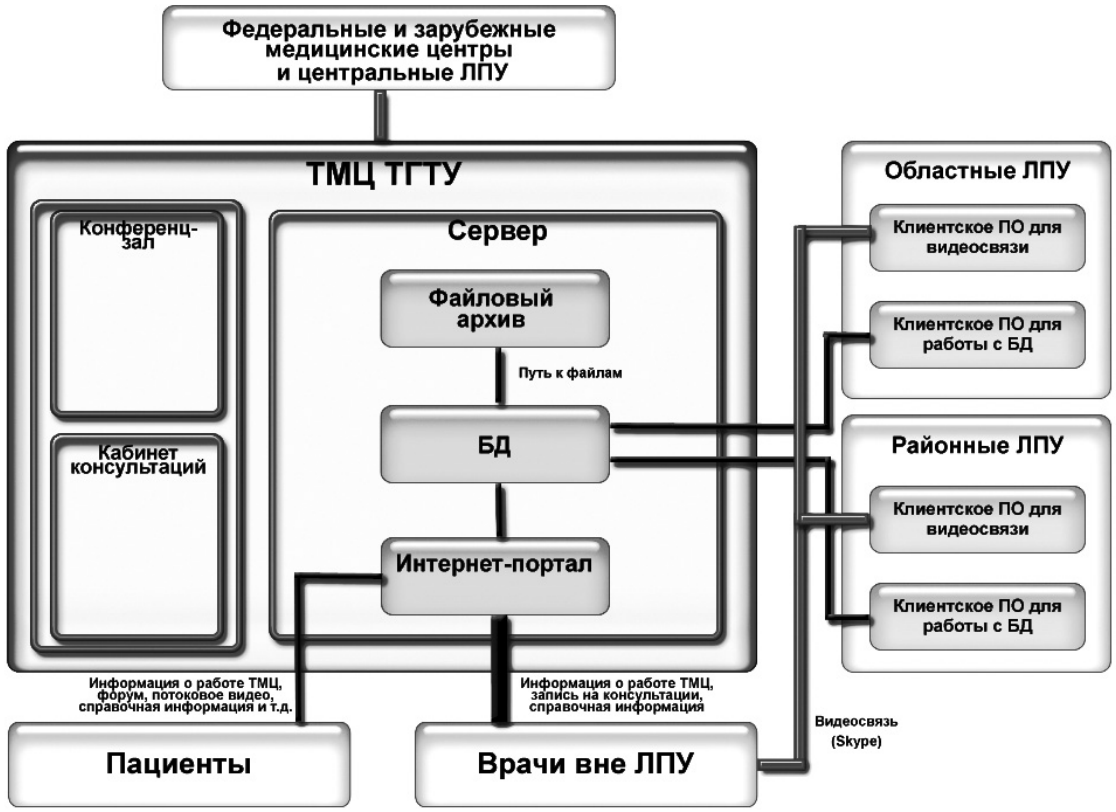


Рис. 1. Трехуровневая телемедицинская сеть

Так как ТГТУ является Интернет-провайдером, он имеет возможность в рамках телемедицинского проекта оптимизировать Интернет-связь между городом и районными центрами внутри сети-провайдера и обеспечить максимально возможную в настоящее время скорость передачи информации от 300 Кбит/с до 2 Мбит/с для связи с федеральными медицинскими центрами. Связь между консультационными ТМЦ в Тамбове и области будет осуществляться на основе комбинирования технологий оптоволоконных сетей, технологии ADSL и спутникового Интернета. Ядром всей региональной телемедицинской сети будет являться ТГТУ.

Интернет-портал создается на платформе PHP, системой управления базами данных является MySQL, а клиентские приложения разрабатываются в среде Borland Development Studio.

На рабочем компьютере ТМЦ установлена программа Skype. На этом же компьютере, а также на компьютерах ЛПУ, будет установлена разрабатываемая программа для работы с файлами и БД, а также для оформления всей необходимой документации (рис. 2). В БД на сервере будут созданы типы пользователя с различными привилегиями.

1. Администратор – все привилегии.

2. Оператор ТМЦ – Полное чтение и запись в БД. Запрещается изменение структуры.

3. Врач – доступ ко всей информации о проведенных им сеансов телемедицины.

4. Гость – доступ к справочной информации (ЛПУ, аптеки …)

На рис. 2 изображены информационные подсистемы телемедицинского сервера и рабочей станции ТМЦ.

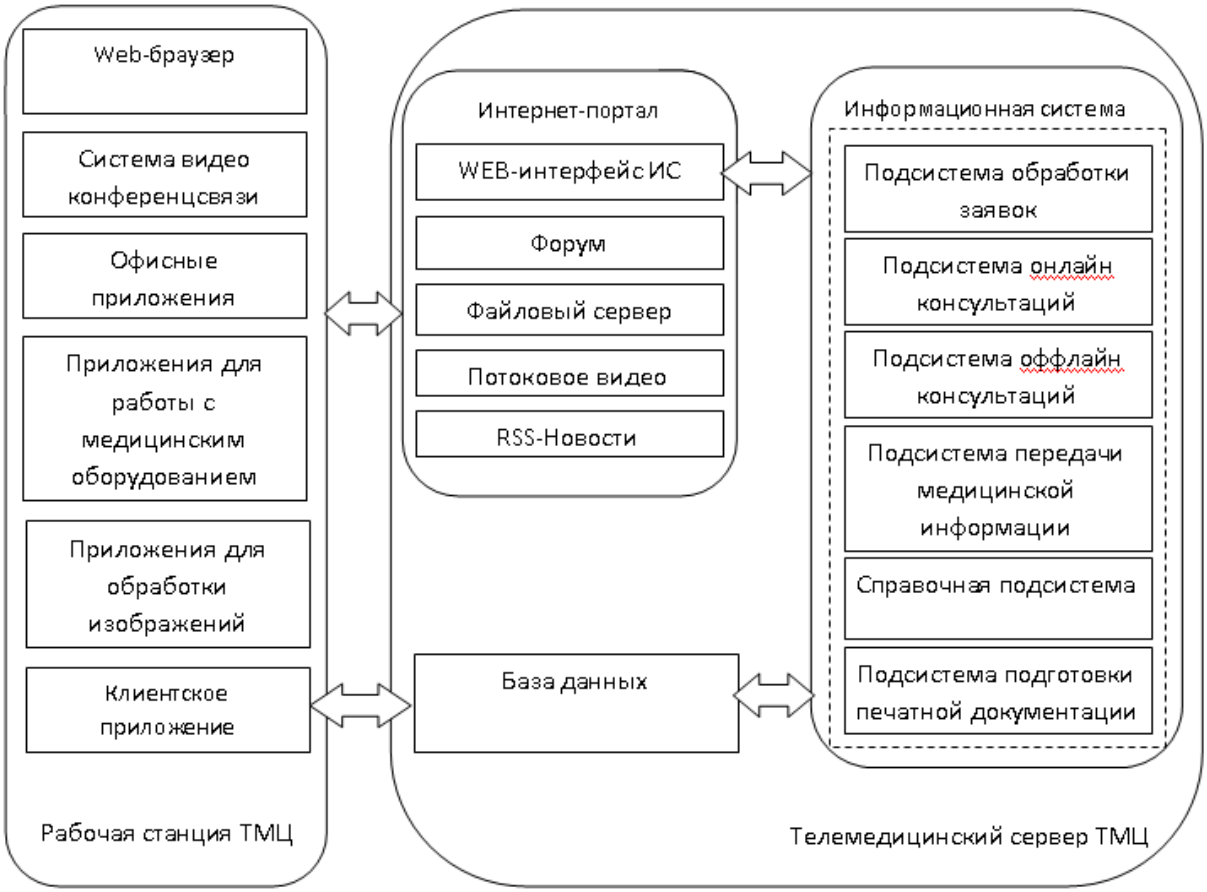


Рис. 2. Информационные подсистемы ТМЦ ТГТУ

Опишем процесс проведения телемедицинской консультации.

1. Оформляется заявка на проведение сеанса телемедицины через разрабатываемое ПО или через Интернет-портал. К этой заявке прикрепляется медицинская запись пациента, в которую лечащий врач заносит информацию о проведенных диагностических исследованиях, пройденном лечении и подобного рода информацию. Данные заносятся в БД на сервере. В случае необходимости дополнительно прилагается медицинская графическая информация – рентгеновские снимки, ЭКГ и т.п. Эта информация в виде отдельных файлов заносится на FTP-сервер. Местоположение файлов записывается в БД.

2. Консультирующий врач при помощи разрабатываемого ПО под своей учетной записью согласовывает время проведения сеанса.

3. В назначенное время будет проводиться сеанс телемедицины с помощью программы Skype. В случае необходимости будет использоваться имеющиеся медицинское оборудование (например, для передачи ЭКГ во время телесеанса).

4. По завершению сеанса телемедицины формируется отчет по утвержденной форме, в соответствии с отчетами будет проводиться оплата участников телемедицинской сети. Информация о названии и местоположении всех файлов записывается в БД на сервере. Все это будет производить разрабатываемое ПО.

Результаты и их обсуждение

Было закуплено оборудование, необходимое для проведения врачебных телеконсультаций и дистанционных курсов повышения квалификации медицинских работников. Был также оборудован телемедицинский кабинет.

Был проведен ряд врачебных телемедицинских консультаций. Основной канал – Интернет на основе бесплатной программы Skype (www.skype.com).

Телеконсультации проводились с телемедицинским комплексом Российской детской клинической больницы (www.rdkb.ru). Также проводился дистанционный консилиум с Российским научным центром хирургии, во время которого обсуждалась тактика лечения больного с дефектом межжелудочковой перегородки и возможного хирургического вмешательства.

В результате проведенных консультаций поставленные ранее диагнозы были уточнены и получены рекомендации по дальнейшему лечению.

Полученные результаты актуальны, консультация с ведущими московскими специалистами проходила в Тамбове, что сократило время и финансовые затраты больных на посещение московских клиник.

Были проведены пробные занятия в рамках дистанционного обучения медицинских работников (хирургов и анестезиологов).

В ТМЦ ТГТУ проводился ряд телеконсультаций с врачами Российской детской клинической больницы и Российского центра хирургии РАМН. Телеконсультирование проводилось в режиме видеоконференцсвязи по программе Skype. В результате проведения телеконсультаций были уточнены диагнозы и даны рекомендации по дальнейшему лечению пациентов.

Заключение

На базе Тамбовского государственного технического университета, на территории лечебно-профилактического учреждения (поликлиника № 6 и санаторий-профилакторий «Тонус») создан телемедицинский центр ТГТУ (ТМЦ ТГТУ).

Главной целью данного проекта является демонстрация улучшенного доступа к медицинскому обслуживанию и информации на уровне базового здравоохранения в сельских регионах посредством создания программы телемедицины. Проект будет сосредоточен в 20 районах Тамбовской области, оснащенных новейшим оборудованием и обеспеченных медицинским консультированием. Проект будет сфокусирован на повышении квалификации, установке оборудования и сессиях телемедицины между областным консалтинговым центром телемедицины в городе Тамбов и 6 городскими больницами (г. Котовска, Мичуринска, Рассказово, Кирсанова, Моршанска, Уварово).

Телемедицина является одной из областей современной медицины, которая может предложить эффективное решение в улучшении медицинского обслуживания. На сегодняшний день телемедицина является одним из самых быстро развивающихся способов лечения пациентов и получения информации. Возможность передачи анализов и данных пациентов посредством телекоммуникации на самом деле можно назвать революцией во многих аспектах медицинского обслуживания. Это может стать в перспективе основной системой, обеспечивающей взаимосвязанное медицинское обслуживание сельским районам Тамбова, уникальной возможностью значительно улучшить медицинское обслуживание и снизить затраты.

Таким образом, разрабатываемая система региональной телемедицины позволит обеспечить доступ к мировым телемедицинским ресурсам для любого врача Тамбовского региона с учетом ограниченности финансовых ресурсов и технических возможностей.

Литература

1. Фролова М.С. Реализация телемедицинских технологий // Медицинские компьютерные технологии: Материалы Всероссийского научного форума. Инновационные технологии медицины XXI века. – М.: МЕДИ Экспо, 2005. – С.539–541.
2. Фролов С.В., Фролова М.С. Дистанционное образование в медицине // Открытое образование. – 2005. – №4. – С.77–80.
3. Фролов С.В., Фролова М.С. Врачебные телемедицинские консультации// Новые медицинские технологии // Новое медицинское оборудование. – 2005. – №12. – С.37–40.
4. Фролов С.В., Фролова М.С. Современные направления телемедицинских технологий // Врач и информационные технологии. – 2006. – №1. – С.45–52.

ВЛИЯНИЕ КОЛЛИЗИЙ И ОШИБОК НА ПРОИЗВОДИТЕЛЬНОСТЬ СЕТЕВЫХ ПРИЛОЖЕНИЙ

С.М. Платунова
Научный руководитель – к.т.н., доцент В.А. Костеж

Описывается постановка задачи исследования влияния коллизий и ошибок на производительность гетерогенных сетей.

Введение

Объектом исследования являются гетерогенные сети. Элементы неоднородности и особенности применяемых протоколов обуславливают вероятностный характер появление кадров в общей среде. Несмотря на наличие в сетевых операционных системах встроенных средств межсетевого взаимодействия, невозможно устранить вероятностный характер величины трафика и его временных показателей. В некоторых ситуациях время решения задачи может оказаться неприемлемым.

Исследование производительности гетерогенных сетей связано с идентификацией ошибок и определением их вероятностного влияния на работу сети. Вне зависимости от причин возникновения ошибок, в условиях проведения экспериментов, существует возможность получения интегральной оценки вероятности выполнения задачи в определенное время.

Особенности влияния коллизий и ошибок на производительность гетерогенных сетей

В работе под гетерогенной сетью понимается информационная сеть, в которой физически реализованы и взаимодействуют разнородные программно-аппаратные средства, такие как:

- операционные системы – среды (UNIX подобные, производства Microsoft и др.), утилиты командной строки, виртуальные машины, операционные системы на Live CD, тонкий клиент;
- различные коммутационные средства (коммутаторы 2, 3 уровней, роутеры, шлюзы);
- сочетание сетей различных типов (ЛВС, корпоративные, Интернет);
- сочетания различных вариантов администрирования (одноранговые сети, двухранговые сети);
- различные протоколы – TCP/IP, NetBIOS, SMB…;
- различные прикладные программы, в широком смысле не согласующие между собой порядок использования общей среды;
- протоколы создания высокоскоростных широкополосных беспроводных сетей.

Принципы разделения среды (CSMA/CD – Carrier Sense Multiple Access With Collision Detection (CSMA/CD)) – и другие неоднородности обусловливают вероятностный характер появления кадров в общей среде и некоторые элементы статичности, например, решение конкретной задачи, первоначально порождает определенное количество кадров (N) впоследствии увеличивающееся из-за процедур разрешения коллизий и исправления ошибок (n).

Под производительностью сети в данной работе будем понимать показатель качества сети, который оценивается в количестве полезного трафика ($N+n$) за установленный промежуток времени ΔT.

В некоторых ситуациях время решения задачи может оказаться неприемлемым. Например, решения задачи отчета 1С при достаточно больших размерах исходных дан-

ных может привести к необходимости изменения физической структуры сети и изменения аппаратного состава.

Наличие дополнительного программного обеспечения может внести удобства в более глубокое изучение особенностей гетерогенных сред. Это может быть изучение трафика по объему и времени возникновения изменений, изучение чувствительности сети к критичности нагрузки, изменения во времени возникновения трафика (время начала работы той или иной программы).

Существует много причин появления дополнительного трафика (n), часть которых обусловлена коллизиями и ошибками. Например, под «ошибкой CRC» понимается пакет с неверной контрольной суммой или поврежденный пакет. Вариантов, из-за чего может произойти повреждение в гетерогенной среде, много, но для нашего исследования представляет интерес только результаты потери пакета и механизм его повтора, соответственно, влияние на производительность гетерогенной сети.

С точки зрения проводимых исследований, коллизия – это тоже кадр (фрейм) с неверной контрольной суммой, но коллизия не является, в общем случае, ошибкой работы сети. Таким образом, если первоначально для решения задачи необходимо было N кадров, то в результате в сети необходимое для решения задачи количество кадров будет (N+n), где n – случайная величина дополнительных кадров, определяемая процессом возникновения коллизий и ошибок и необходимостью повтора отдельных кадров.

Задача в сети должна быть решена за определенное время ΔT, которое, в общем случае, является случайной величиной, зависящей от множества факторов, например, оперативной срочности решения задачи, важности решения задачи, условий решения задачи т.д. На первоначальном этапе изучения будем рассматривать эту величину как статическую.

Вне зависимости от причин возникновения дополнительного трафика существует возможность при проведении тестирования или имитационного моделирования выявить интегральную оценку вероятности выполнения задачи в определенное время ΔT. В соответствии с исследованиями С. Поповского [1] и в соответствии со стандартами:

- в случае возникновения коллизии сетевой интерфейс (например, сетевая карта) повторяет передачу пакета с задержкой от 0 до 0,5 мсек;
- повтором уничтоженных пакетов занимается по тайм-ауту транспортный уровень протокола передачи, и, естественно, он не может быть столь малым, как в случае коллизии;
- начальное значение тайм-аута может составлять, по результатам исследований, 0,2 сек;
- начальные и последующие значения тайм-аута могут меняться для различных протоколов и сред передачи;
- учитывая время передачи уничтоженного пакета и обработки ошибки, суммарное время повтора составляет, примерно 0,5 сек;
- в определенных ситуациях коллизии могут мало влиять на показатели производительности гетерогенной сети (например, в случае низкого сетевого фона), но и в этом случае изменения трафика (n) будет обусловлено сетевыми ошибками. Автор исследований обратил внимание на то, что при кажущемся малом количестве ошибок CRC сеть становится практически неработоспособной.

В результате исследований В.М. Вишневского, А.И. Ляхова (Институт проблем передачи информации РАН, Москва, Россия), Хр. Даскаловой (Институт информационных технологий БАН, София, Болгария [2]) разработан вероятностный метод исследования, оценки производительности и оптимизации гетерогенных сетей. В соответствии с их исследованиями последовательные попытки передачи каждой станции сети разделены интервалом задержки, а также случайным временем отсрочки (backoff time), отсчет которого ведется только при свободном канале.

Время отсрочки b измеряется в интервалах длительностью σ и равновероятно выбирается из множества (0, …, *W*–1), где $W = W_i$ – окно конкуренции, зависящее от стадии отсрочки, определяемой количеством i неудачных попыток передач пакета. После каждой неудачной передачи *W* удваивается, т.е. $W_i = 2iW_0$, и так до достижения максимальной величины W_m.

В настоящее время ведутся научные исследования, направленные на повышение эффективности сетей и выбор оптимальных параметров.

Если поведение отдельных станций предполагать независимым, то процесс можно описывать цепью Маркова, изображенной на рис. 1 (*p* – вероятность коллизии, предполагаемая независимой от числа *i* сделанных попыток передачи). Здесь состояние станций описывается парой (*ik*), где *k* – значение счетчика отсрочки.

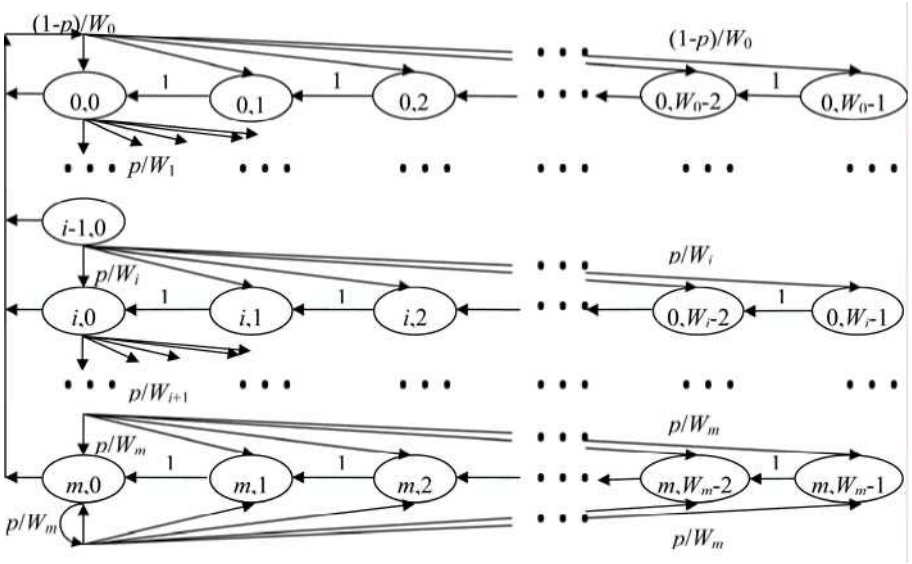


Рис. 1. Марковская модель изменения состояния станции

Данный вариант модели рационален в предположении идеального канала передачи, что приводит к завышению оценок производительности гетерогенной сети, так как уровень помех и ошибок в сети существенен. В целом процессы в гетерогенной среде могут быть описаны выражением

$$\mathrm{P}(\mathrm{A}_{y}) = \begin{Bmatrix} 1, если \Delta t(\mathrm{N}+n) \lhd \Delta\mathrm{T}, \mathrm{M}=0, m=0 \\ 1-\mathrm{P}_{z} \\ 0, если \Delta t(\mathrm{N}+n) \rhd \Delta\mathrm{T} \end{Bmatrix}.$$

Здесь $P(Ay)$ – вероятность успешного завершения передачи трафика $(N+n)$ в заданный период времени ΔT; Δt – стандартное время (длительность) кадра; $(M+m)$ – дополнительный фоновый трафик, вызванный коллизиями и ошибками; P_z – вероятность занятости общей среды трафиком или задержками в соответствии с результатами вышеописанных исследований.

В дополнение к сказанному процесс загрузки сети можно охарактеризовать так:

- в сети существует период Δt_1 до наступления первой коллизии или ошибки. За этот период в сети успевает пройти $N_{\Delta t1}$ кадров трафика основной задачи, и $M_{\Delta t1}$ кадров фонового трафика;
- при возникновении первой коллизии количество необходимого трафика для дальнейшей передачи будет равняться

$$\left(\left(\mathrm{N} - \mathrm{N}_{\Delta t1}\right) + 1\right) + \left(\left(\mathrm{M} - \mathrm{M}_{\Delta t1}\right) + 1\right);$$

- далее существует период Δt_2 до наступления второй коллизии или ошибки. За этот период в сети успевает пройти $N_{\Delta t2}$ кадров трафика основной задачи и $M_{\Delta t2}$ кадров фонового трафика;
- при возникновении второй коллизии количество необходимого трафика для дальнейшей передачи будет равняться:
$$\left(\left(\left(\mathrm{N}-\mathrm{N}_{\Delta t1}\right)+1\right)-\mathrm{N}_{\Delta t2}+1\right)+\left(\left(\left(\mathrm{M}-\mathrm{M}_{\Delta t1}\right)+1\right)-\mathrm{M}_{\Delta t2}+1\right).$$

Процесс повторяется до момента исчерпания трафика основной (N+*n*) и/или фоновой (M+*m)* задачи.

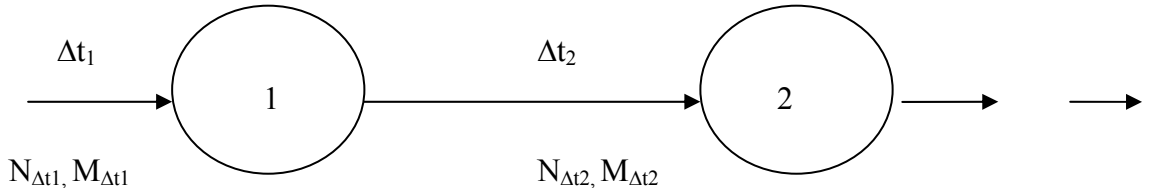


Рис. 2. Процесс загрузки сети

Если фоновый трафик намного меньше, чем основной ($M+m$)<<($N+n$), то вероятность наступления первой коллизии равняется

$$\mathrm{P}_{k1}=\Delta t\left(\mathrm{N}+n\right)/\Delta\mathrm{T}/k,$$

для второй коллизии:

$$\mathrm{P}_{k2}=\Delta t\left(\left(\mathrm{N}-\mathrm{N}_{\Delta t1}+1\right)+n\right)/\left(\Delta\mathrm{T}/k-\Delta t_1\right)$$

и т.д. до момента прохождения основного трафика $N+n$.

Если моменты возникновения основного ($N+n$) и фонового ($M+m$) трафиков близки и находятся в промежутке: $[0, \Delta t_1] << \Delta T/k$, то вероятность наступления первой коллизии вычисляется на основе следующих предположений. В общую шину поступают кадры основного трафика и фонового трафика, причем поступление каждого кадра равновозможно в любой момент времени длительностью $[0, \Delta t_1]$. Моменты поступления кадров независимы один от другого. Коллизия возникает, если разность между моментами поступления кадров меньше Δt (Δt намного меньше ΔT). Необходимо найти вероятность возникновения 1-й коллизии.

Обозначим моменты поступления кадров основного трафика и фонового трафика соответственно через t_n и t_m. В силу условия задачи должны выполняться двойные неравенства: $0 \le t_n \le \Delta\mathrm{T}/k; 0 \le t_m \le \Delta\mathrm{T}/k$. Введем в рассмотрение прямоугольную систему координат xOy. В этой системе двойным неравенствам удовлетворяют координаты любой точки квадрата OTAT. Таким образом, этот квадрат можно рассматривать как фигуру, координаты точек которой представляют все возможные значения моментов поступления кадров в общую шину. Коллизия возникает, если разность между моментами поступления кадров меньше Δt, т.е. если:

- $t_m - t_n \lhd \Delta t$ для $t_m \rhd t_n$ или, что то же самое, $t_m \lhd t_n + \Delta t$ для $t_n \rhd t_m$. Неравенство выполняется для тех точек фигуры, которые лежат выше прямой $t_m = t_n$ и ниже прямой $t_m = t_n + \Delta t$;
- $t_n - t_m \lhd \Delta t$ для $t_n \rhd t_m$, или, что то же самое, $t_m \rhd t_n - \Delta t$ для $t_m \lhd t_n$. Неравенство имеет место для точек, расположенных ниже прямой $t_m = t_n$ и выше прямой $t_m = t_n - \Delta t$.

Все точки, координаты которых удовлетворяют неравенствам, принадлежат шестиугольнику. Таким образом, этот шестиугольник можно рассматривать как фигуру, координаты точек которой являются благоприятными моментами времени t_m и t_n. Искомая вероятность 1-ой коллизии равняется:

$$P_{k1} = \left((\Delta T / k)^2 - (\Delta T / k - \Delta t)^2 \right) / (\Delta T / k)^2 = (\Delta t (2\Delta T / k - \Delta t)) / (\Delta T / k)^2 .$$

В дальнейшем предполагается в процессе исследований провести математическое моделирование с учетом вышеприведенных отношений.

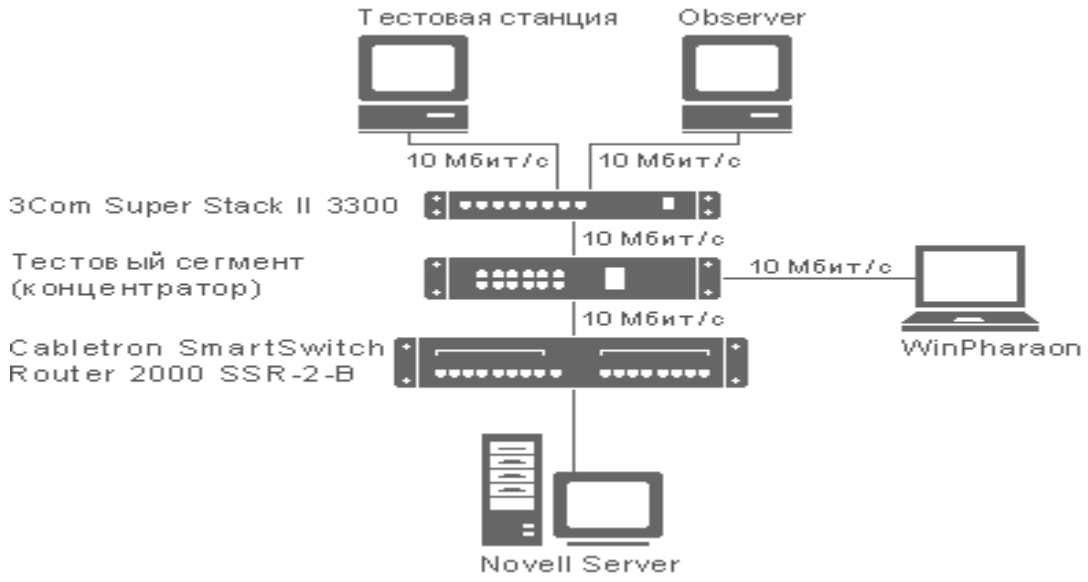


Рис. 3. Схема диагностики сетевых процессов

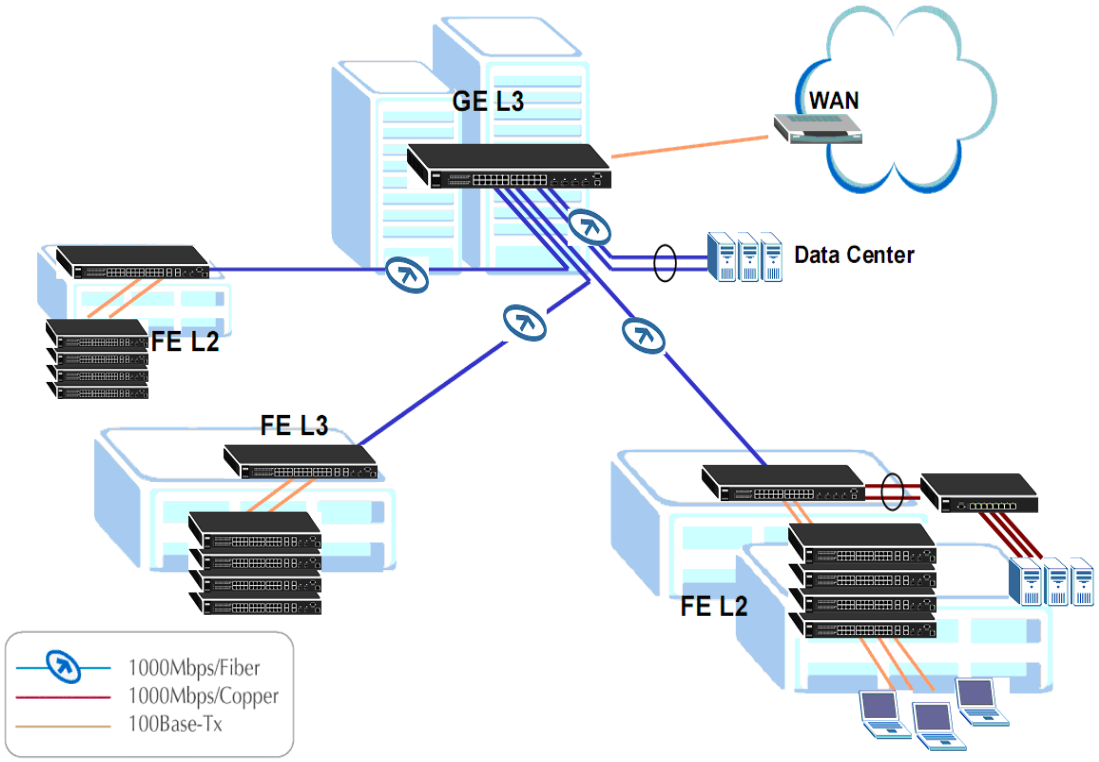


Рис. 4. Типовая структура сети уровня корпорации

Причина ошибок	Локальные коллизии	Удаленные коллизии	Поздние коллизии е	Короткий кадр	Длинный кадр	Jabber	Ошибка CRC
Дефектная сетевая плата	>5% при U<30%	>5% при U<30%	Есть	Есть	Есть	Есть	Есть
Дефектный драйвер платы				Есть	Есть	Есть	Есть
Дефектный коммутатор	>5% при U<30%	>5% при U<30%	Есть			Есть	Есть
Неправильное подключение активного оборудования	>5% при U<30%	>5% при U<30%	Есть			Есть	
Слишком длинный кабель			Есть				Есть
Более 4 коммутаторов			Есть				
Неправильное заземление компьютеров	>5% при U<30%	>5% при U<30%	Есть			Есть	Есть
Дефекты кабельной системы и пассивного оборудования	>5% при U<30%	>5% при U<30%	Есть			Есть	Есть
Источник шума рядом с кабельной системой	>5% при U<30%	>5% при U<30%	Есть			Есть	Есть

Таблица. Типы ошибок и коллизий, фиксируемые измерительным средством

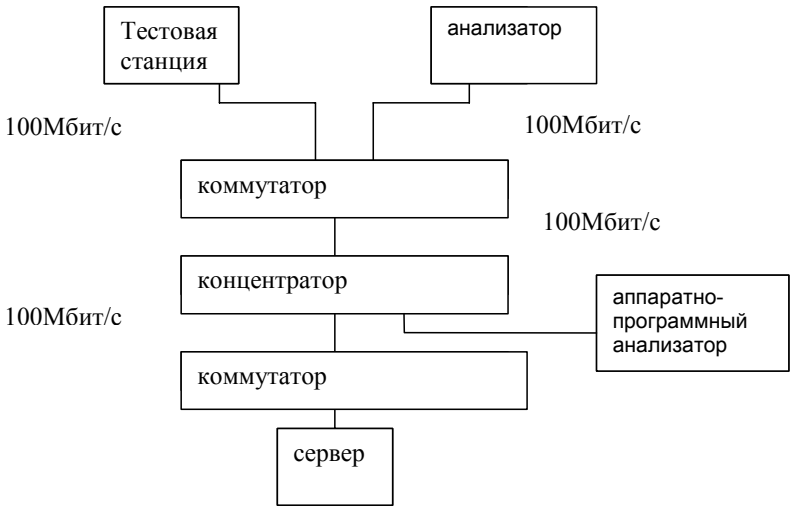


Рис. 5. Схема имитационного моделирования

Математическое моделирование [mathematical modeling] – процесс построения и исследования в динамике поведения математических моделей сети с использованием дополнительно разработанных программ. Кроме математического моделирования, предполагается имитационное моделирование. С. Поповским [1] была проведена диагностика сетевых процессов. В последнее время в сетях используется современные сетевые элементы. Типы ошибок и коллизий, фиксируемые измерительным средством в эксперименте С. Поповского, приведены в таблице.

В связи с изменениями скорости и аппаратного состава рационально провести имитационное моделирование на базе других сетевых элементов. При имитационном моделировании будут использоваться реальные аппаратные средства, а имитироваться будут условия решения задачи.

Заключение

В статье рассмотрен ряд аналитических моделей влияния коллизий и ошибок на производительность гетерогенных сетей. Ранее подобные исследования проводились В.М. Вишневским, А.И. Ляховым (Институт проблем передачи информации РАН), Москва, Россия, Хр. Даскаловой (Институт информационных технологий БАН, София, Болгария) для беспроводных сетей. Также предложены изменения проведения имитационного моделирования, ранее проведенные в исследованиях С. Поповского.

Литература

1. Российская государственная библиотека [Электронный ресурс] / Центр информ. технологий РГБ; ред. Власенко Т.В.; Web-мастер Козлова Н.В. – Электрон, дан. – М.: Рос. гос. б-ка, 1997. – Режим доступа: http://www.rsl.ru, свободный. – Загл. с экрана. – Яз. рус., англ.
2. Поповский С. Анализ влияния коллизий и ошибок на производительность сетевых приложений // База знаний компании ProLAN, Дефекты сетей [Электронный ресурс] – Режим доступа: http://www.prolan.ru/company/article/kb/56.html#h1_01, свободный – Загл. с экрана. – Яз. рус.
3. Вишневский В.М., Ляхов А.И., Даскалова Хр. Вероятностные методы исследования широкополосных беспроводных сетей // Proc. Int. Workshop «Distributed Computer and Communication Networks» (DCCN-2005). Sofia, Bulgaria, April 23–29, pp. 9-18, 2005. – Режим доступа: www.lyakhov.iitp.ru/dccn05_1.pdf
4. Вишневский В.М., Ляхов А.И. и др. Широкополосные беспроводные сети передачи информации. – М.: Техносфера, 2005. – 592 с.
5. Олифер В.Г., Олифер Н.А. Компьютерные сети: Принципы, технологии, протоколы: Учеб. пособие для студ. вузов. – СПб: Питер, 2004. – 672 стр.
6. Таненбаум Э. Компьютерные сети. – СПб: Питер, 2008. – 992 с.
7. Семенов Ю.А Алгоритмы телекоммуникационных сетей. В 3-х частях. Часть 1. Алгоритмы и протоколы каналов и сетей передачи данных. – М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2007 – 637 с.
8. Семенов Ю.А. Алгоритмы телекоммуникационных сетей. В 3-х частях. Ч.2 Протоколы и алгоритмы маршрутизации в INTERNET Ч.2 – М.: Интернет-Университет 2007. – 832 с.
9. Скотт Хогдал Дж. Анализ и диагностика компьютерных сетей. – М.: Лори, 2007. – 364 с.
10. Лоу А.М., Кельтон В.Д. Имитационное моделирование. – СПб: Питер, Издательская группа BHV, 2004. – 848 стр.
11. Рыжиков Ю.И. Имитационное моделирование: Теория и технологии. – СПб: КОРОНА-принт, 2004. – 384 с.

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ IP-СЕТЕЙ ПЕРЕДАЧИ ДАННЫХ НА ОСНОВЕ УРАВНЕНИЙ СПЛОШНОЙ СРЕДЫ

Д.С. Северов, С.В. Трифонов, М.И. Миненко
(Московский физико-технический институт (государственный университет))
Научный руководитель – к.ф.-м.н., доцент Я.А. Холодов
(Московский физико-технический институт (государственный университет))

Настоящая работа посвящена проблеме снижения вычислительной сложности компьютерного моделирования пакетных сетей передачи данных. Авторы предлагают замену традиционного дискретно-событийного моделирования эволюции пакетов численным моделированием потоков данных с использованием интегро-дифференциальных уравнений сплошной среды в частных производных.

Введение

Современные сети обмена информацией демонстрируют устойчивую тенденцию к переходу на пакетный принцип передачи данных. По сравнению с традиционными телефонными сетями пакетные сети передачи данных используют более сложные и менее предсказуемые протоколы взаимодействия. В условиях такой непредсказуемости решение практически значимых задач требует адекватного моделирования процессов протекающих в сети.

Разработан и активно применяется целый спектр различных специализированных средств моделирования. На одном конце спектра находятся средства подробного дискретно-событийного моделирования [1], позволяющие проследить судьбу каждого пакета и предназначенные, в первую очередь, для детального исследования и разработки протоколов взаимодействия. Такие средства обладают сложностью, неудовлетворительной для моделирования коллективных явлений и сети в целом. На другом конце спектра находятся средства имитационного моделирования сети в целом как системы массового обслуживания [2]. Такие средства игнорируют детали взаимодействия, существенные для исследования протоколов, и воспроизведения коллективных явлений.

Известны попытки моделирования пакетных сетей передачи данных с использованием численного решения обыкновенных дифференциальных уравнений для математических ожиданий основных моделируемых величин, изменяющихся во времени [3, 4]. Данный подход создает определенные трудности для прямого моделирования протоколов взаимодействия, не позволяет адекватно отражать современные сложные алгоритмы управления очередями и различать механизмы потерь данных в очередях и в каналах связи.

В основу предлагаемого решения заложено представление переноса данных и информации о потерях данных функциями, изменяющимися как во времени, так и вдоль очереди. Типовые устройства сети моделируются в зависимости от выполняемой функции дифференциальными уравнениями, как в частных производных, так и обыкновенными, а также явными кусочно-непрерывными функциональными зависимостями, включая логические. Такой подход представляет собой удачную комбинацию возможности прямого детального моделирования всех существенных особенностей протоколов взаимодействия, и умеренной вычислительной сложности для громоздких конфигураций сети.

Теоретическая модель

Рассмотрим модель сети как совокупность маршрутизаторов, отправителей, получателей, и соединяющих их однонаправленных каналов связи. Естественным образом каждый двунаправленный канал реальной сети представляется в модели двумя однонаправленными, а каждый узел реальной сети – необходимой комбинацией отправителей

и получателей. В модели в каждом маршрутизаторе и в каждом отправителе присутствуют только выходные очереди, по одной на каждый интерфейс. Входные очереди в маршрутизаторах и получателях данных, а также множественные очереди, конкурирующие за пропускную способность одного интерфейса, в текущей постановке не рассмотрены.

Модельные каналы относятся к типу «точка – точка». Реальные каналы типа «точка – многоточка» могут быть представлены комбинацией нескольких модельных каналов получателя и отправителей. Всевозможные канальные устройства, применяемые при создании реальной сети связи, учитываются либо изменением параметров канала, либо введением маршрутизатора. К первой группе устройств относятся, например, повторители и/или регенераторы сигнала, а ко второй – коммутаторы.

Базовым понятием настоящей модели является *сессия*, которая определяется как пара узлов сети, претендующих на передачу данных от одного (отправителя) к другому (получателю)[1]. В каждый момент модельного времени в сети могут присутствовать отправленные пакеты нескольких сессий.

Из-за достижения максимальной пропускной способности отдельных каналов не все пакеты могут быть переданы в очередной канал немедленно и накапливаются в очередях для последующей передачи. Если длина очереди превышает определенные пороги, то часть пакетов отбрасываются с определенной вероятностью.

Принято, что все данные, поступающие на модельный маршрутизатор из всех входящих каналов, мгновенно перераспределяются в соответствии с заданными правилами маршрутизации, после чего помещаются в очередь, соответствующую каждому исходящему каналу связи и, после прохождения очереди, поступают в канал связи. В модели отсутствуют входные очереди, в которых пакеты могли бы находиться после поступления на маршрутизатор и до перераспределения.

Моделируемые величины и структура модели

В каждой точке модели различаются потоки данных нескольких активных соединений. Для каждого активного соединения в каждой рассматриваемой точке модели определяются: (1) размер пакета, характерный для этого соединения; (2) та доля общего потока пакетов в рассматриваемой точке, которая приходится на пакеты данного активного соединения; (3) информация о потерянных пакетах данного активного соединения. Поток данных в целом в определенной точке модели представляется как перенос модельного «вещества» с определенными «плотностью» (количеством бит в пакете) и «скоростью» (количеством пакетов проходящих в единицу времени).

Важно подчеркнуть, что модель, предлагаемая в настоящей работе, в значительной мере определяется пакетной организацией трафика, и поведение сети описывается в терминах движения именно пакетов. Например, в случае протоколов обмена подобных TCP динамика поведения отправителя определяется, как правило, количеством подтверждений об успешном приеме пакетов, и не зависит напрямую от количества бит в этих пакетах. Аналогично, сброс данных из очереди маршрутизатора производится пакетами, одновременно для всех битов данных входящих в состав пакета.

Таким образом, в отличие от эталонной модели, где рассматриваются события цикла жизни определенных дискретных пакетов, в настоящей модели рассматриваются изменения во времени непрерывных характеристик, соотнесенных с общим количеством переданных и/или потерянных пакетов.

Структура модели представляется направленным графом, в котором различаются входы, выходы, а также узлы и ребра. Узлу соответствует функция перераспределения

[1] *Сессия* может не быть сессией в смысле того или иного коммуникационного протокола, Например, поток пакетов UDP/IP также является сессией в контексте данной модели.

потоков данных в маршрутизаторе. Вход представляет отправителя данных, а выход – потребителя. Ребро отражает очередь и примыкающий к ней канал связи.

В узлах графа (маршрутизаторах) задаются соотношения (правила маршрутизации) характеризующие перераспределение потоков данных. На входах графа (отправителях) и выходах графа (получателях) задаются соотношения, описывающие изменения потоков данных при возникновении в источниках и достижении получателей. На ребре задаются соотношения отражающие продвижение и сброс пакетов в очереди, и канале связи с учетом характеристик интерфейса.

Обозначения

Скалярные величины, имеющие один и тот же смысл в терминах модели, для различных фрагментов модели обозначаются одинаковыми символами с различными индексами $<\text{величина}>_{s_{<\text{индекс сессии}>}}^{<\text{элемент}>_{<\text{индекс элемента}>}.<\text{терминал}>_{<\text{индекс терминала}>}}$.

$<\text{величина}>$ символ, обозначающий моделируемую величину: ρ (плотность), d (потери в пути), u (скорость), ψ (сброс), x (координата), t (момент времени).

$<\text{элемент}>$ символ, обозначающий тип структурного элемента модели: q (очередь), c (канал), r (узел, маршрутизатор), h (хост, отправитель или получатель).

$<\text{индекс элемента}>$ – целочисленный индекс элемента структуры среди однотипных элементов.

$<\text{терминал}>$ обозначает вход (in) или выход (out) структурного элемента.

$<\text{индекс терминала}>$ – порядковый номер входа или выхода. Для величин моделирующих структурный элемент, имеющий не более одного входа и одного выхода, порядковый номер может не указываться.

s обозначает элемент полезной нагрузки сети, сессию.

$<\text{индекс сессии}>$ – целочисленный индекс данной сессии среди прочих сессий.

Обозначения величин, которые соотносятся только со структурным элементом и не соотносятся с сессией, не содержат $s_{<\text{индекс сессии}>}$. Обозначение величин, которые соотносятся только с сессией и не соотносятся со структурными элементами, не содержат фрагментов $<\text{элемент}>_{<\text{индекс элемента}>}$, $<\text{терминал}>_{<\text{индекс терминала}>}$.

В тех обозначениях, где не указан фрагмент $<\text{индекс элемента}>$, подразумевается то значение, которое соответствует текущему по тексту элементу структуры.

В тех обозначениях, где не указан фрагмент $<\text{индекс сессии}>$, подразумевается то значение, которое соответствует текущей тексту сессии.

В рамках данной работы, если явно не оговаривается иное, ограничимся следующими вариантами фрагментов $<\text{элемент}>_{<\text{индекс элемента}>}$: s_n, q_m, c_m, h_l, r_k, out_j, in_i. В квадратных скобках после обозначения величины может указываться ее размерность, например $\rho_{s_n}^{q_m}$ [бит/пакет].

Сессия

Для каждой сессии s_n, $n = 1,...,N$ в целом определяется характерный размер пакета ρ_{s_n} [бит]. Далее для каждой пары [сессия; элемент структуры] в каждый момент времени определяется количество пакетов данной сессии, ожидаемых в данном элементе структуры к данному моменту, но потерянных (сброшенных) на пути сессии $d_{s_n}^{<\text{элемент}>}$ [пакет].

Ребро: очередь и канал связи

Для моделирования движения пакетов вдоль каждой очереди q_m, $m = 1,...,M$ определяющими являются потери данных. Эти потери являются результатом работы алгоритмов управления очередью. Действие подобных алгоритмов сводится к мгновенному изъятию пакетов из очереди. При таком изъятии между пакетами, примыкающими с двух сторон к изъятому пакету, не образуется никакого «зазора». Изъятие пакета может наступить, в общем случае, из любого места очереди.

Модель каждой очереди q_m содержит непрерывную «пространственную» координату x^{q_m}, имеющую смысл положения пакета в очереди, измеренного в пакетах от конца очереди, примыкающего к передающему интерфейсу. Такой подход позволяет моделировать алгоритмы управления очередью, изымающие пакеты из любого требуемого места очереди.

Скорость изменения координаты пакета в очереди q_m принимается одинаковой для всех сессий проходящих через данную очередь и обозначается u^{q_m}.

В каждый момент времени t в каждой точке x^{q_m} очереди q_m для каждой сессии s_n определяется несколько переменных: $\rho_{s_n}^{q_m}$ [бит/пакет] – текущая плотность пакетов, $d_{s_n}^{q_m}$ [1] – плотность потерь пакетов, $\psi_{s_n}^{q_m}$ [1] – плотность сброса пакетов, $\tau_{s_n}^{q_m}$ [с] – время в пути. Указанные переменные могут быть соотнесены с дискретным наполнением очереди q_m пакетами различных сессий следующим образом. Предположим, что единичный интервал $[x^{q_m}, x^{q_m}+1]$ очереди взаимно однозначно соответствует пакету сессии s_n. Тогда $\int_{x^{q_m}}^{x^{q_m}+1} \rho_{s_n}^{q_m} dx = \rho_{s_n}$ – количество бит в этом пакете, $\int_{x^{q_m}}^{x^{q_m}+1} d_{s_n}^{q_m} dx$ – количество пакетов сессии s_n, потеря которых должна быть учтена после приема данного пакета, а $\int_{x^{q_m}}^{x^{q_m}+1} \psi_{s_n}^{q_m} dx$ –- вероятность сброса данного пакета. Определим для очереди действующую плотность

$$\rho^{q_m} = \sum_n \rho_{s_n}^{q_m} . \qquad (1)$$

Отметим, что в случае взаимно однозначного соответствия некоторого единичного интервала очереди пакету сессии s_n, в каждой точке этого интервала $\rho^{q_m} = \rho_{s_n}^{q_m}$. Скорость продвижения пакетов в очереди

$$u^{q_m} = \frac{Q^{c_m}}{(\rho^{q_m \cdot out} + \tilde{\rho}^{c_m})} + \sum_n \int_{x^{q_m}}^{X^{q_m}} \psi_{s_n}^{q_m} d\tilde{x}^{q_m}, \qquad (2)$$

где Q^{c_m} [бит/с] – номинальная пропускная способность выходного интерфейса, $\rho^{q_m \cdot out} \square \rho^{q_m}\Big|_{x^{q_m}=X^{q_m}}$, $\tilde{\rho}^{c_m}$ [бит/пакет] – количество бит, добавляемых к пакету сетевого уровня, при образовании последовательности кадров канального уровня, X^{q_m} [пакет] – длина очереди q_m.

$$\frac{dX^{q_m}}{dt} = \sum_n u_{s_n}^{r_k \cdot out_j} - u^{q_m \cdot in}, \qquad (3)$$

где $r_k.out_j$ выход того узла-маршрутизатора r_k, к которому примыкает очередь q_m. $u^{q_m.in} = u^{q_m}\Big|_{x^{q_m}=0}$. Важно подчеркнуть, что $X^{q_m} \geq 0$.

Эволюция плотности данных $\rho_{s_n}^{q_m}$ и потерь пакетов $d_{s_n}^{q_m}$ сессии s_n в очереди q_m описывается следующими уравнениями:

$$\begin{cases} \dfrac{\partial}{\partial t}\rho_{s_n}^{q_m} + \dfrac{\partial}{\partial x^{q_m}}\left(\rho_{s_n}^{q_m} u^{q_m}\right) = -\rho_{s_n}^{q_m} u_{s_n}^{q_m} \psi_{s_n}^{q_m} \\ \dfrac{\partial}{\partial t} d_{s_n}^{q_m} + \dfrac{\partial}{\partial x^{q_m}}\left(d_{s_n}^{q_m} u^{q_m}\right) = u_{s_n}^{q_m} \psi_{s_n}^{q_m} \end{cases}. \quad (4)$$

Исходя из топологии графа, на входной границе $x^{q_m} = 0$ очереди для уравнений (4) имеем следующие граничные условия:

$$\begin{cases} \rho_{s_n}^{q_m}(t,0) = \rho_{s_n}^{q_m.in} = \rho_{s_n}^{r_k.out_j} \\ d_{s_n}^{q_m}(t,0) = d_{s_n}^{q_m.in} = d_{s_n}^{r_k.out_j} \end{cases}, n = 1,\ldots,N \quad (5)$$

На выходной границе $x^{q_m} = X^{q_m}$ граничные условия не требуются, а значения $\rho_{s_n}^{q_m}(t, X^{q_m})$, $d_{s_n}^{q_m}(t, X^{q_m})$, однозначно определяются из уравнений (4), благодаря тому, что скорость движения пакетов в очереди $u^{q_m}(t, x^{q_m})$ всегда неотрицательна.

В наиболее распространенном частном случае, когда в качестве политики активного управления очередью (AQM – active queue management) выбирается алгоритм случайного раннего обнаружения [5] (RED – Random Early Detection), а сброс происходит в хвосте очереди имеем:

$$\int_0^1 \psi_{s_n}^{q_m} d\tilde{x}^{q_m} = \begin{cases} 0, rnd_{s_n}^{q_m} \geq P^{q_m}(t) \\ 1, rnd_{s_n}^{q_m} < P^{q_m}(t) \end{cases}, \quad (6)$$

где $rnd_{s_n}^{q_m}$ – случайная величина с равномерным распределением на отрезке [0;1], $P^{q_m}(t)$ – вероятность сброса пакета, которая в терминах RED AQM определяется как (см. [2–4])

$$P^{q_m}(t) = \begin{cases} 0, & D^{q_m}(t)\tilde{P}^{q_m}(t) < 1 \\ \min\left(1; \dfrac{\tilde{P}^{q_m}(t)}{2 - D^{q_m}(t)\tilde{P}^{q_m}(t)}\right), & 1 \leq D^{q_m}(t)\tilde{P}^{q_m}(t) \leq 2 \\ 1, & 2 \leq D^{q_m}(t)\tilde{P}^{q_m}(t) \end{cases}, \quad (7)$$

$D^{q_m}(t) = \int_{t_{drop}^{q_m}}^{t} u^{q_m.in}(\tau)d\tau$, $t_{drop}^{q_m}$ – момент последнего сброса, (8)

$$P^{q_m}(t) = \begin{cases} 0, & \tilde{X}^{q_m} < X_{min}^{q_m} \\ \dfrac{\tilde{X}^{q_m} - X_{min}^{q_m}}{X_{max}^{q_m} - X_{min}^{q_m}} P_{max}^{q_m}, & X_{min}^{q_m} \leq \tilde{X}^{q_m} \leq X_{max}^{q_m} \\ 1, & \tilde{X}^{q_m} > X_{max}^{q_m} \end{cases}, \quad (9)$$

где $\tilde{X}^{q_m}(t)$ – ожидаемая (прогнозируемая) длина очереди, а $X_{min}^{q_m}$, $X_{max}^{q_m}$, $P_{max}^{q_m}$ постоянные параметры RED AQM.

$$\frac{d\tilde{X}^{q_m}}{dt} = \frac{\ln \alpha^{q_m}}{\delta^{q_m}} \left(\tilde{X}^{q_m}(t) - X^{q_m}(t) \right). \quad (10)$$

Последнее уравнение получено из выражения скользящего среднего $\tilde{X}^{q_m}(t+\delta^{q_m}) = (1-\alpha^{q_m})\tilde{X}^{q_m}(t) + \alpha^{q_m} X^{q_m}(t)$. Безразмерный постоянный параметр $\alpha^{q_m} \in [1,10]$ определяет степень зависимости ожидаемой длины очереди от кратковременных колебаний длины реальной очереди. Имеющий размерность времени параметр δ^{q_m} равен интервалу времени между замерами реальной длины очереди.

Исчерпывающим представляется следующий набор параметров канала: Q^{c_m}, $\tilde{\rho}^{c_m}$, $\tilde{\tau}^{c_m}$ [c] (задержка распространения данных в данном канале), B^{c_m} (безразмерная относительная частота ошибочной передачи бита данных, Bit Error Rate – BER). В канале, также как и в очереди, происходит потеря целостности пакетов, причинами которой являются искажения сигнала и/или внешние помехи.

Определим вероятность того, что кадр канального уровня минует канал связи неповрежденным:

$$P^{c_m} = (1 - B^{c_m})^{\rho^{c_m.in} + \tilde{\rho}^{c_m}}. \quad (11)$$

Важные для данной модели отличия сводятся к тому, что «потерянные», т.е. искаженные и/или зашумленные данные, продолжают перемещаться по каналу и занимать его ресурсы, пока ошибка не будет обнаружена в конце канала и пакет не будет отброшен. Такое поведение может интерпретироваться как уменьшение скорости выхода из канала моделируемых **целостных** пакетов. Такая особенность позволяет моделировать канал следующими соотношениями на выходе

$$\begin{cases} \rho_{s_n}^{c_m.out}(t) = \rho_{s_n}^{q_m.out}(t - \tilde{\tau}^{c_m}) \\ \tau_{s_n}^{c_m.out} = \tilde{\tau}^{c_m} + \tau_{s_n}^{q_m.out}(t - \tilde{\tau}^{c_m}) \\ d_{s_n}^{c_m.out}(t) = (1 - P^{c_m})\tilde{\tau}^{c_m} p_{s_n}^{q_m.out} u^{q_m.out}(t - \tilde{\tau}^{c_m}) + d_{s_n}^{l.in}(t - \tilde{\tau}^{c_m}) \end{cases} \quad (12)$$

$$u^{c_m.out}(t) = P^{c_m} u^{q_m.out}(t - \tilde{\tau}^{c_m}). \quad (13)$$

Узел – маршрутизатор

Ядро маршрутизации моделируется соотнесением потоков данных со стороны входящих ребер – и потоков данных в сторону выходящих ребер. Исходя из топологии графа на стыке входа маршрутизатора $r_k.in_i$ и выхода канала c_m,

$$\begin{cases} \rho_{s_n}^{r_k.in_i} = \rho_{s_n}^{c_m.out} \\ d_{s_n}^{r_k.in_i} = d_{s_n}^{c_m.out} \\ \tau_{s_n}^{r_k.in_i} = \tau_{s_n}^{c_m.out} \\ u_{s_n}^{r_k.in_i} = u^{c_m.out} p_{s_n}^{q_m.out} \end{cases}. \quad (14)$$

Предположим, что данные любой сессии поступают в любой узел не более чем из одного ребра, примыкающего к этому узлу, и покидают этот узел не более чем через одно ребро. Значения на выходе узла определяются в соответствии с таблицей маршрутизации.

$$\begin{cases} \rho_{s_n}^{r_k.out_j} = \rho_{s_n}^{r_k.in_i} \\ d_{s_n}^{r_k.out_j} = d_{s_n}^{r_k.in_i} \\ \tau_{s_n}^{r_k.out_j} = \tau_{s_n}^{r_k.in_i} \\ u_{s_n}^{r_k.out_j} = u_{s_n}^{r_k.in_i} \end{cases}. \quad (15)$$

Входы и выходы графа – источники и получатели

Прежде чем описывать граничные условия модели, задаваемые на входах и выходах графа, имеет смысл напомнить, что ребра графа, моделирующего сеть, являются однонаправленными и любой двунаправленный канал моделируется двумя ребрами.

Примем, что каждой сессии соответствует один и только один вход графа, а также один и только один выход графа. На ребре, примыкающему к **входу** h, графа исходя из топологии графа, следует записать:

$$\begin{cases} \rho_{s_n}^{q_m.in} = \rho_{s_n}^{h_I} \\ d_{s_n}^{q_m.in} = d_{s_n}^{h_I} \\ \tau_{s_n}^{q_m.in} = \tau_{s_n}^{h_I} \\ u_{s_n}^{q_m.in} = u_{s_n}^{h_I} \end{cases}. \quad (16)$$

Значения $\rho_{s_n}^{h}, u_{s_n}^{h}$ определяются тем активным соединением s_n, которое берет начало в источнике h.

На ребре, примыкающем к выходу h графа, значения $\rho^{l.out}$, $d^{l.out}$, $\tau^{l.out}$, $u^{l.out}$ однозначно определяются из уравнений (12), вследствие того, что скорость движения пакетов в канале $u^{l} > 0$ всегда положительна и ее направление совпадает с направлением соответствующей ей ветви графа. Иными словами, на выходе графа ничто не влияет непосредственно на динамику трафика ребра, примыкающего к данному выходу. Исходя из топологии графа на стыке **выхода** графа h и выхода примыкающего канала *l*,

$$\begin{cases} \rho_{s_n}^{h_O} = \rho_{s_n}^{c_m.out} \\ d_{s_n}^{h_O} = d_{s_n}^{c_m.out} \\ \tau_{s_n}^{h_O} = \tau_{s_n}^{c_m.out} \\ u_{s_n}^{h_O} = u^{c_m.out} \end{cases}. \quad (17)$$

В некоторых случаях (таких, например, как телевизионная или радиотрансляция в сети передачи данных средствами протокола UDP) параметры в начале сессии, на хосте-источнике, являются постоянными, и параметры в конце сессии на хосте-получателе больше никак не влияют на поведение сети. В общем случае, такие соотношения для пары конец одной сессии – начало другой сессии можно представить следующим образом.

$$\begin{cases} \rho_{s_{n_2}}^{h.out} = P^{n_2,n_1}\left(\rho_{s_{n1}}^{h.in}, d_{s_{n1}}^{h.in}, \tau_{s_{n1}}^{h.in}, u_{s_{n1}}^{h.in}\right) \\ d_{s_{n_2}}^{h.out} = D^{n_2,n_1}\left(\rho_{s_{n1}}^{h.in}, d_{s_{n1}}^{h.in}, \tau_{s_{n1}}^{h.in}, u_{s_{n1}}^{h.in}\right) \\ \tau_{s_{n_2}}^{h.out} = T^{n_2,n_1}\left(\rho_{s_{n1}}^{h.in}, d_{s_{n1}}^{h.in}, \tau_{s_{n1}}^{h.in}, u_{s_{n1}}^{h.in}\right) \\ u_{s_{n_2}}^{h.out} = U^{n_2,n_1}\left(\rho_{s_{n1}}^{h.in}, d_{s_{n1}}^{h.in}, \tau_{s_{n1}}^{h.in}, u_{s_{n1}}^{h.in}\right) \end{cases}, \quad n_1 \neq n_2 \quad (18)$$

Зависимости P^{n_2,n_1}, D^{n_2,n_1}, T^{n_2,n_1}, U^{n_2,n_1} определяются применяемыми протоколами передачи данных и количественными характеристиками сетевой нагрузки.

В случае традиционной передачи данных с использованием протокола TCP значения в конце одного активного соединения на определенном выходе графа определяют значения в начале другого, ответного активного соединения (вдоль которого переносятся ACK пакеты) на другом определенном входе графа.

Пусть сессия s моделирует однонаправленный поток данных некоторого TCP-соединения и начинается на входе h_I, а заканчивается на выходе h_O. Пусть сессия $\tilde{s}$ аналогично моделирует поток подтверждений приема (ACK) для данного TCP соединения начинается на входе $\tilde{h}_I$ и заканчивается на выходе $\tilde{h}_O$. Тогда на приемном конце TCP-соединения (выход h_O, вход $\tilde{h}_I$) верно следующее:

$$\begin{cases} \rho_{\tilde{s}}^{\tilde{h}_I} = \rho_{\tilde{s}} \\ d_{\tilde{s}}^{\tilde{h}_I} = d_s^{h_O} \\ \tau_{\tilde{s}}^{\tilde{h}_I} = \tau_s^{h_O} \\ u_{\tilde{s}}^{\tilde{h}_I} = u_s^{h_O} \end{cases}, \tag{19}$$

где $\rho_{\tilde{s}}$ – плотность, соответствующая размеру пакета подтверждения приема (ACK). На передающем конце TCP-соединения (вход h_I, выход $\tilde{h}_O$)

$$\begin{cases} \rho_s^{h_I} = \rho_s \\ d_s^{h_I} = 0 \\ \tau_s^{h_I} = 0 \\ u_s^{h_I} = u_s \end{cases}, \tag{20}$$

где ρ_s – плотность, соответствующая размеру пакета передаваемых данных, а u_s определяется протоколом TCP. В данной работе моделируется версия Reno.

$$u_s = \begin{cases} u_{\tilde{s}}^{\tilde{h}_O}\left(1 + 1/W_s\right) & \text{, если режим - "Congesion avoidance"} \\ \tilde{W}_s / \tau^{\tilde{h}_O} & \text{, если режим - "Fast Recovery"} \\ W_s / \tau^{\tilde{h}_O} & \text{, если режим - "Slow Start"} \end{cases}. \tag{21}$$

W_s – текущий размер окна данного TCP-соединения, $\tilde{W}_s$ – пороговый размер окна данного TCP-соединения (threshold). Рост W_s определяется решением следующего дифференциального уравнения:

$$\frac{dW_s}{dt} = \begin{cases} u_{\tilde{s}}^{\tilde{h}_O} / W_s & \text{, если режим - "Congesion avoidance"} \\ u_{\tilde{s}}^{\tilde{h}_O} / W_s & \text{, если режим - "Fast Recovery"} \\ u_{\tilde{s}}^{\tilde{h}_O} & \text{, если режим - "Slow Start"} \end{cases}. \tag{22}$$

Определим далее несколько контрольных переменных, определяющих переключение режимов и связанные с этим изменения W_s и $\tilde{W}_s$. $I_{\tilde{s}}^{loss}$ – количество потерянных пакетов. $t_{\tilde{s}}^{loss}$ – момент обнаружения потери целого пакета. $T_{\tilde{s}}^{loss}$ – время, прошедшее с

момента $t_{\tilde{s}}^{loss}$ – количество пакетов, принятых с момента. $\tilde{T}_{\tilde{s}}$ – предельное время ожидания подтверждения о приеме (timeout).

Размер окна данного TCP-соединения в режиме «congestion avoidance» определяется решением следующего уравнения

$$\frac{dW_s}{dt} = \frac{u_{\tilde{s}}^{\tilde{h}_O}}{W_s} - \frac{W_s}{2\tau_{\tilde{s}}^{\tilde{h}_O}} \min(1; d^{\tilde{h}_O}) \cdot \qquad (23)$$

Сравнительные вычисления

Для сравнения поведения сети в целом по гистограммам усредненной производительности и усредненного времени оборота соединений (RTT) рассмотрен случай относительно сложной топологии сети (рис. 1), сходный с использованным в [4].

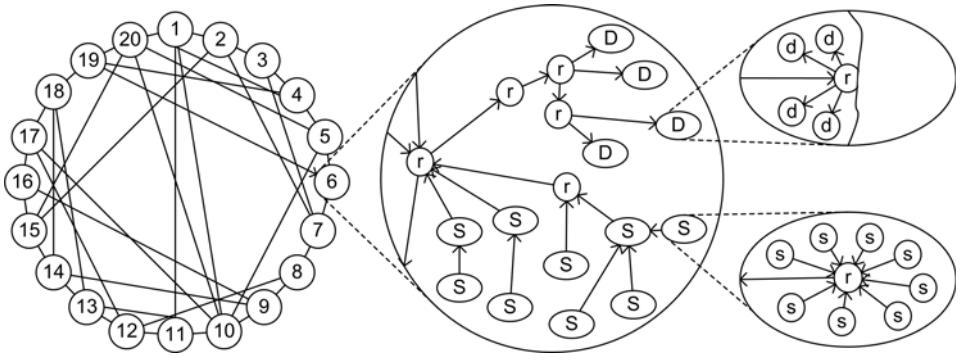


Рис. 1. Топология сети для сравнения поведения модели в целом

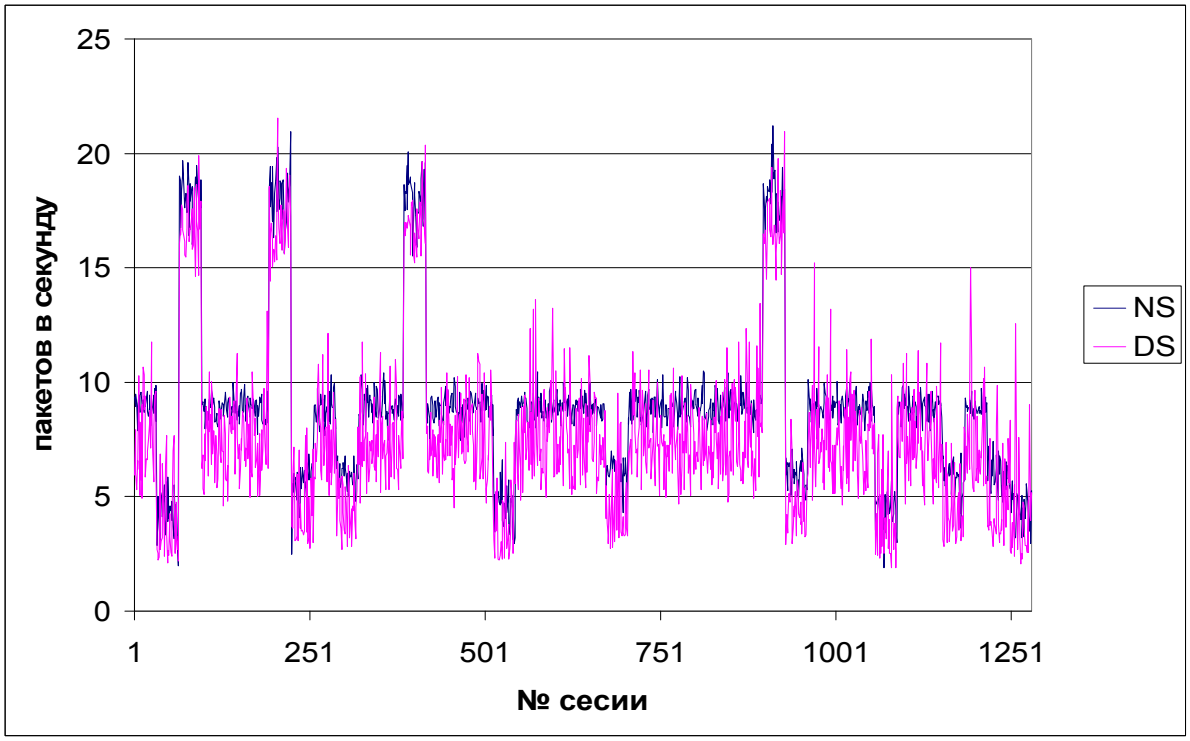


Рис. 2. Усредненные результаты производительности сети

Всего моделировалось 1280 сессий, при этом имела место одна загруженная очередь на пути каждой сессии. Использовались следующие характеристики сети: пропу-

скная способность магистрали – 20 Мбит/с, хорды подсети – 5 Мбит/с, размер пакетов – 8 Кбит/пакет. Трафик был организован так, что половина пакетов отправлялась по магистрали в соседнюю подсеть, а вторая половина – по хорде подсети. Соответствующие усредненные результаты производительности сети в зависимости от сессии показаны на рис. 2. Усреднение результатов делалось за период времени в 200 с. Красным цветом показаны результаты предложенной модели (DS), синим цветом, показаны результаты, полученные с использованием network simulator [1] (NS).

Заключение

В настоящей работе предложен новый подход к моделированию сетей передачи данных с использованием функций двух переменных в виде дифференциальных уравнений в частных производных, обладающей меньшей сложностью, чем традиционные средства дискретно-событийного моделирования. Выполнены сравнительные вычисления с использованием как традиционных средств [1], так и предлагаемого подхода. Результаты вычислений демонстрируют хорошую степень подобия, как на простейших, так и достаточно сложных конфигурациях сети

Литература

1. http://www.isi.edu/nsnam/ns/
2. http://www.opnet.com/solutions/network_rd/modeler.html
3. V. Misra, W.-B. Gong and D. Towsley. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED / Proceedings of ACM/SIGCOMM, 2000.
4. Yong Liu, Francesco Lo Presti, Vishal Misra, Don Towsley and Yu Gu. Fluid Models and Solutions for Large-Scale IP Networks / Proceedings of ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS), 2003.
5. S. Floyd and V. Jacobson. Random Early Detection gateways for congestion avoidance. IEEE / ACM Transactions on Networking. – August 1993. – 1(4). – P. 397–413.

УПРАВЛЯЮЩАЯ ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА ВЫСОКОЙ НАДЕЖНОСТИ С РЕКОНФИГУРАЦИЕЙ

Е.А. Дзбоев (Санкт-Петербургский государственный университет сервиса и экономики)
Научный руководитель – д.т.н., профессор В.А. Богатырев (Санкт-Петербургский государственный университет сервиса и экономики)

Рассматривается структура многопроцессорной вычислительной системы управления, обладающей свойствами повышенной надежности в результате применения метода резервирования отдельных узлов системы, а также повышения достоверности при приеме, обработке и передаче информации. Рассматриваемая система обладает свойствами программной реконфигурации в зависимости от решаемой функциональной задачи. Рассматриваемое техническое решение защищено авторским свидетельством (А.С. СССР № 1798946 G 06 F 11/20).

Введение

Требования, предъявляемые к комплексу бортового оборудования, могут быть реализованы только при наличии мощной и гибкой коммуникационной и вычислительной инфраструктуры. Для поддержки широкого набора различных операций вычислительные мощности должны обладать возможностью адаптации к динамически изменяющимся условиям функционирования и поставленным задачам. Бортовой комплекс должен обеспечивать высокую степень автономного функционирования и в то же время, в качестве задачи наивысшего приоритета, он должен обеспечивать живучесть.

Структура бортовой вычислительной системы разрабатывается как аппаратная реализация ее архитектуры, выполненная с ориентацией на конкретно решаемую в данный момент задачу. Одним из обязательных свойств бортовой вычислительной системы должна быть способность к реконфигурации. Бортовая вычислительная система должна быть реализована как распределенная вычислительная система, состоящая из многих вычислителей, способная выполнять все функции управления, телеметрии и мониторинга. Объединение различных вычислительных функций на борту в единую систему с высокой степенью резервирования позволяет осуществлять тесное взаимодействие между различными процессами и оптимизировать гибкое использование резервированных вычислительных ресурсов для выполнения различных заданий в зависимости от требований к функционированию и от необходимого уровня устойчивости к отдельным сбоям.

Основная часть

Рассматриваемая многопроцессорная вычислительная система обладает свойствами реконфигурации структуры, повышенной надежностью за счет применения метода резервирования отдельных узлов системы, а также повышения достоверности при приеме, обработке и передаче информации. Данная схема представлена на рисунке.

Управляющая вычислительная система содержит подсистему управления резервированием, содержащую в своем составе устройство реконфигурации и блок прерывания, и предназначенную для реализации принципа гибкого резервирования, т.е. замены вышедших из строя компонентов системы на исправные. Система содержит подсистему внешней памяти, доступную для всех входящих в систему резервируемых каналов, каждый из которых содержит процессор, блок локальной памяти, блок контроля, коммутатор обмена (мультиплексор), группу коммутаторов каналов, устройство ввода-вывода. Все устройства, входящие в систему, объединены соответствующим образом интерфейсными шинами и шинами контрольных и управляющих сигналов [1].

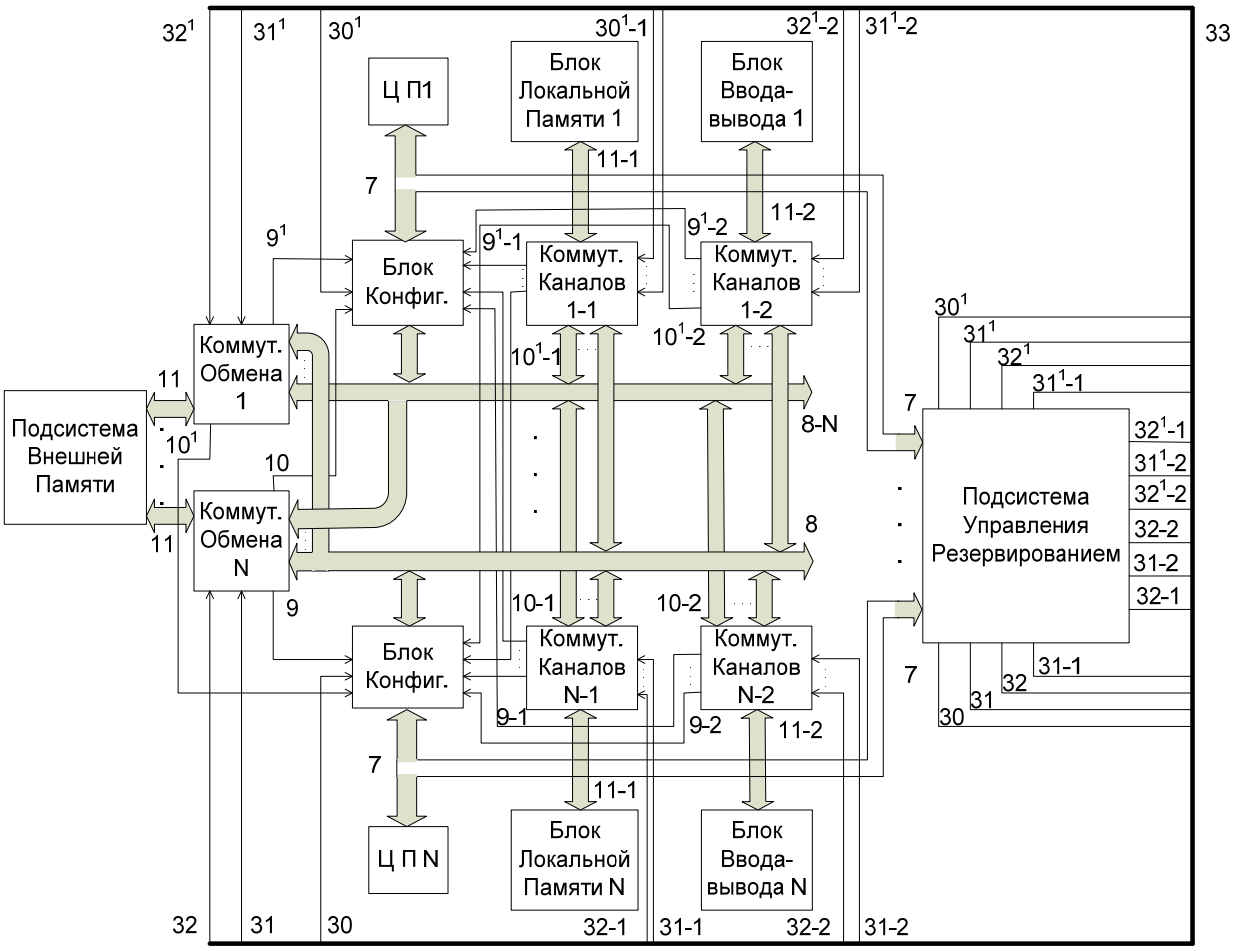


Рисунок. Структурная схема управляющей вычислительной системы с реконфигурацией

В исходном состоянии на управляющих выходах подсистемы управления резервированием устанавливаются сигналы управления, открывающие магистральные приемопередатчики всех блоков контроля. В коммутаторах каналов открываются магистральные приемопередатчики, подключающие выходные шины коммутаторов каналов к магистрали собственного канала. Магистральные приемопередатчики, подключающие выходные шины коммутаторов каналов к магистралям резервных каналов, блокируются.

При нормальной работе системы каждый процессорный блок через определенные, заданные интервалы времени осуществляет тестовый самоконтроль, результаты которого записываются в блок сравнения каждого канала подсистемы управления резервированием. Интервал времени самотестирования в каждом канале определяется таймером подсистемы управления резервированием. В момент записи результатов самотестирования в блок сравнения таймер устанавливается в исходное состояние. В случае записи неправильного кода самотестирования или отсутствия записи кода на выходе блока сравнения появляется сигнал, поступающий на один из входов элемента инициализации реконфигурации подсистемы управления резервированием. На второй вход этого элемента поступает сигнал с выхода таймера, что свидетельствует о полном выходе из строя блока процессора. Сигнал с выхода элемента инициализации реконфигурации поступает на соответствующий вход устройства управления реконфигурацией, а также на вход блока прерывания. На управляющих выходах устройства управ-

ления реконфигурацией формируются сигналы, блокирующие магистральные приемопередатчики блока контроля и коммутатора каналов отказавшего процессорного блока. Блок прерывания организует режим прерывания программы во всех резервируемых каналах. В режиме прерывания программы все исправные процессорные блоки читают содержимое регистров управления каждого канала подсистемы управления резервированием. Информация в регистры управления каналов поступает из блока управления реконфигурацией и определяет номер процессорного блока, который берет на себя функции вышедшего из строя. В канале неисправного процессорного блока магистральные приемопередатчики коммутаторов каналов, соединенные с каналом процессорного блока, взявшего на себя функции вышедшего из строя, разблокируются. Осуществляется доступ к ресурсам канала вышедшего из строя процессорного блока.

В случае исправной работы процессорного блока его доступ к ресурсам канала осуществляется через блок контроля и коммутаторы каналов.

Прохождение информации в блоке контроля и в коммутаторах каналов осуществляется через магистральные приемопередатчики. В случае искажения информации они формируют сигналы «ошибка», которые поступают в блок контроля по цепям готовности. В блоке контроля сигналы «ошибка» от магистральных приемопередатчиков поступают на информационные входы регистра состояния и на вход элемента инициализации блока прерывания. Процессорный блок переходит на подпрограмму обработки прерывания, которой, читая содержимое регистра состояния, определяет участок, на котором произошло нарушение в процессе приемопередачи информации.

В этом случае процессорный блок записывает в регистр реконфигурации подсистемы управления резервированием код реконфигурации. В соответствии с этим кодом блок управления реконфигурацией формирует на своих управляющих выходах сигналы управления, в соответствии с которыми неисправный участок приемопередачи информации исключается путем блокировки магистральных приемопередатчиков коммутаторов каналов, а доступ к общему ресурсу осуществляется через блоки коммутаторов резервных каналов.

Режим искажения прохождения информации через магистральные приемопередатчики блока контроля соответствует режиму отказа процессорного блока.

Реконфигурация структуры системы по условиям функциональной задачи осуществляется программным способом, методом обращения процессорного блока к блоку реконфигурации и записи в регистр реконфигурации соответствующего кода, определяющего необходимую конфигурацию системы.

Подсистема внешней памяти представляет собой совокупность многовходового перепрограммируемого и оперативного запоминающих устройство с определенной степенью избыточности.

Подсистема управления резервированием представляет собой многоканальное устройство, число каналов которого соответствует числу резервируемых каналов самой системы. Инициализация подсистемы управления резервированием осуществляется результирующими сигналами тестового самоконтроля каждого канала, сигналами блоков контроля и коммутаторов каналов. В системе осуществляется распределенный внутренний самоконтроль, что исключает использование централизованной системы внутреннего контроля.

В рассмотренной структуре резервирование осуществляется методом замещения таким образом, что при полном или частичном выходе из строя одного из резервируемых каналов исправно работающий канал, взявший функции отказавшего, имеет полный доступ к внутренним ресурсам и внешним устройствам отказавшего канала, что позволяет осуществлять процесс восстановления системы без обращения к общей внешней памяти.

Заключение

Рассмотренная структура управляющей вычислительной системы представляет собой многопроцессорную систему, состоящую из идентичных узловых вычислителей, которые соединены резервированной системой магистральной шины. В принципе каждый из узлов способен выполнять любые процессы и иметь доступ ко всем каналам ввода-вывода. Общая структура аппаратных средств обеспечивает отсутствие «изолированных мест повреждения» в системе. Узловые вычислители являются полностью обособленными устройствами, они не имеют общих компонентов, за исключением устойчивой к сбоям коммуникационной системы и каналов ввода-вывода, соединенных с подсистемами. Физическая реализация коммуникационных шин обеспечивает то, что неисправное устройство не будет мешать передаче информации между другими устройствами. Благодаря использованию специальной схемы логики интерфейса гарантируется, что отключенное или неисправное устройство будет полностью изолированно.

Можно сделать вывод, что рассмотренная система может быть охарактеризована как реконфигурируемая управляющая система высокой надежности.

Литература

1. Международный Научно-Технический Центр. TECH-DB.ISTC.RU/Проекты.
2. А.С. СССР. № 1798946 H 05 K 10/00, G 06 F11/20. Резервированная вычислительная система. / В.Ф. Беззубов.

ПОВЫШЕНИЕ ТОЧНОСТИ АЛГОРИТМОВ РАСПОЗНАВАНИЯ РЕЧИ НА ОСНОВЕ СКРЫТЫХ МАРКОВСКИХ МОДЕЛЕЙ

П.В. Балакшин

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

В статье представлен краткий обзор существующих алгоритмов распознавания речи. Дано сравнение алгоритмов на основе скрытых марковских моделей. Показана схема работы алгоритма Витерби. Предложена модификация данного алгоритма за счет введения дополнительной информации о длительности состояний.

Введение

Процесс взаимодействия человека с ЭВМ стал одним из наиболее важных вопросов развития компьютерной техники с момента появления самих ЭВМ. Сначала технологи «общались» с ней через посредника-программиста. Затем был диалоговый интерфейс, после графический. Но человечество всегда искало более простое, удобное, естественное решение. Поэтому нетрудно понять, что голосовой интерфейс – это тема, которая на протяжениии последних пятидесяти лет привлекает внимание ученых и инженеров всего мира. Голосовой интерфейс на языке пользователя – это наилучшее решение. Действительно, ведь речь – это наиболее натуральная, удобная, эффективная и экономичная форма человеческого взаимодействия.

Речевой ввод объединяет множество различных технологий и приложений. В некоторых случаях важно не понимание лингвистического содержания, а идентификация говорящего или языка, на котором происходит общение. Однако в основе речевого ввода лежит процесс распознавания речи. Первые конкретные шаги в данной области были предприняты в 1971 г. по заказу министерства обороны США [1].

В связи с быстрым и постоянным развитием, как науки, так и самой жизни человека, понятия скорости и времени становятся все более значимыми. Бурный рост информационных технологий позволил несколько снизить время, затрачиваемое на передачу и обработку информации. И на первый план стали выходить понятия точности, достоверности и надежности этой информации.

В системах автоматического распознавания речи важнейшим показателем является точность распознавания слитной речи, которая определяется отношением количества верно распознанных слов к сумме всех распознаваемых слов, пропущенных слов и лишних (неверно распознанных) слов [2]. В настоящее время этот показатель стремиться к значению 95% и выше. Повышение указанного показателя является важной и актуальной задачей. Подтверждением этого могут служить:

- множество научно-исследовательских центров, объединенных в Международную Ассоциации Речевого Взаимодействия (International Speech Communication Association);
- научно-технические конференции, крупнейшей из которых является ежегодная Interspeech;
- различные программно-технические разработки (Dragon Naturally Speaking, IBM Via Voice, встроенное речевое управление в ОС Vista).

Следует выделить 2 аспекта, в которых ведутся исследования, связанные с повышением точности распознавания: аппаратный и программный. В данной работе нами уделено внимание лишь алгоритмам распознавания, которые являются частью программного аспекта. Таким образом, задачей нашего исследования является модификация существующих алгоритмов распознавания слитной речи для повышения точности распознавания. Базовые принципы алгоритмов распознавания речи были сформулированы начале 80-х гг. прошлого столетия Л. Рабинером и Р. Шафером в книге «Цифро-

вая обработка речевых сигналов» [1]. В качестве опубликованных исследований о трудоемкости алгоритмов распознавания речи можно выделить только статью сотрудников компании SPIRIT [3].

Существующие алгоритмы

Все существующие ныне основные алгоритмы могут быть группированны следующим образом:

- распознавание на основе скрытых марковских моделей (СММ), где речевой сигнал представляется набором состояний с некоторыми вероятностями перехода между ними (рис. 1);
- распознавание на основе нейросети;
- гибридные модели.

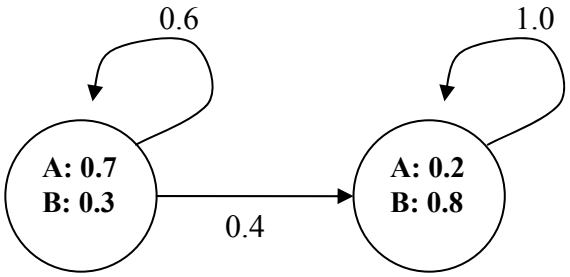


Рис. 1. Пример модели на основе СММ [2]

Каждая из них имеет свои достоинства и недостатки. Прежде всего, это обусловлено конкретными задачами и местом применения. Кроме того, распознавание речи осложняется тем, что существует множество источников вариативности, связанных с речевым сигналом.

Модели на основе СММ чаще применяются для небольших приложений (например, системы команд) и несколько лучше работают с раздельной речью. Перечислим их недостатки.

- Вероятности зависят только от текущего состояния;
- Отсутствует корреляции между смежными входными фреймами;
- Необходимо использовать множество параметров для максимизации вероятности [2].

Однако за счет возможностей модификации и автоматической сегментации данный метод является одним из основных в области распознавания слитной речи[4].

Алгоритм Витерби и моделирование длительности состояний

Рассмотрим основные элементы СММ [5].

(1) Конечное множество скрытых состояний $S = s_1, s_2, s_3, \ldots, s_N$;

(2) Конечное множество наблюдаемых состояний $V = v_1, v_2, v_3, \ldots, v_M$ – дискретный алфавит;

(3) q_t – состояние модели в каждый момент времени t;

(4) Матрица переходных вероятностей $A = \{a_{xy}\}$, где $a_{xy} = P\langle q_t = s_j \,|\, q_{t-1} = s_{j-1}\rangle$;

(5) Матрица вероятности наблюдаемых символов (матрица эмиссий) $B = \{b_i(k)\}$, где $b_i(k) = P\langle o_t = v_k \,|\, q_t = s_{j-1}\rangle$, $k = 1 \ldots M$, o_t – наблюдаемый символ в момент времени t;

(6) Начальная вероятность состояний $\pi = (\pi_j)$, $\pi_j = P(q_1 = s_j)$;

(7) Набор параметров, определяющий СММ, $\lambda = (\pi, A, B)$.

Пусть задана последовательность наблюдений $O = o_1, o_2, o_3, \ldots, o_T$ и модель $\lambda = (\pi, A, B)$. Задача – вычислить и максимизировать $P\langle O|\lambda\rangle$. Решить ее можно с помощью 5 алгоритмов: прямого хода, обратного хода, прямого–обратного хода, алгоритма Витерби и алгоритма Бауэма-Уэлша. Но обязательно при этом учитывать, что каждый алгоритм подходят к конкретным задачам. В таблице 1 показаны некоторые особенности двух последних алгоритмов.

Алгоритм	Преимущества	Базовый алгоритм	Особенности вычислительной сложности
Витерби	Поиск оптимальной модели; максимизация вероятности	Аналогичен алгоритму прямого хода	Усложнение вычислительной сложности
Баума-Уелша	Переоценка параметров, максимизация вероятности	Основан на алгоритме прямого-обратного хода	Наиболее сложный алгоритм

Таблица. Сравнительная характеристика алгоритмов на основе СММ

Оба представленных алгоритма максимизируют вероятность модели. Но при этом важно то, что алгоритм Бауэма-Уэлша является самый трудоемким алгоритмом в вычислительном плане. Кроме того, не существует известного аналитического выражения для параметров такой модели [3]. Все это дает право утверждать, что алгоритм Витерби является оптимальным для максимизации вероятности модели и, тем самым, повышения точности распознавания.

Рассмотрим участок декодирования (рис. 2). Цифрами условно обозначены вероятности появления символов наблюдения $b_j(o_t)$.

Стандартный алгоритм Витерби основан на пошаговой максимизации функции правдоподобия наблюдений $P\langle O|\lambda\rangle$. Для отыскания наилучшей последовательности состояний $Q = q_1, q_2, q_3, \ldots, q_T$ по заданной последовательности наблюдений $O = o_1, o_2, o_3, \ldots, o_T$ необходимо пошагово переопределять величину. $\delta_j(t) = \max_{q_1, q_2, q_3, \ldots, q_{t-1}} P[q_1, q_2, q_3, \ldots, q_t = j, o_1, o_2, o_3, \ldots, o_T|\lambda]$. Согласно показанному алгоритму, для состояния S_j в момент времени t мы должны выбрать S_{j-1} как наиболее вероятное предыдущее состояние. Следовательно, оптимальной будет траектория ACDEG с вероятностью $p = 0{,}6 * 0{,}6 * 0{,}6 * 0{,}6 * 0{,}6 = 0{,}07776$.

Заметим, что результат поиска оптимальной последовательности состояний может не быть единственным [6]. Это обусловлено существованием нескольких приемлемых критериев оптимальности. Как в нашем примере, один из возможных критериев состоит в том, чтобы выбрать такие состояния q_t, каждое из которых, взятое отдельно, является наиболее вероятным. Можно также максимизировать вероятность появления пар состояний (q_t, q_{t+1}) или троек состояний (q_t, q_{t+1}, q_{t+2}) и т.д.

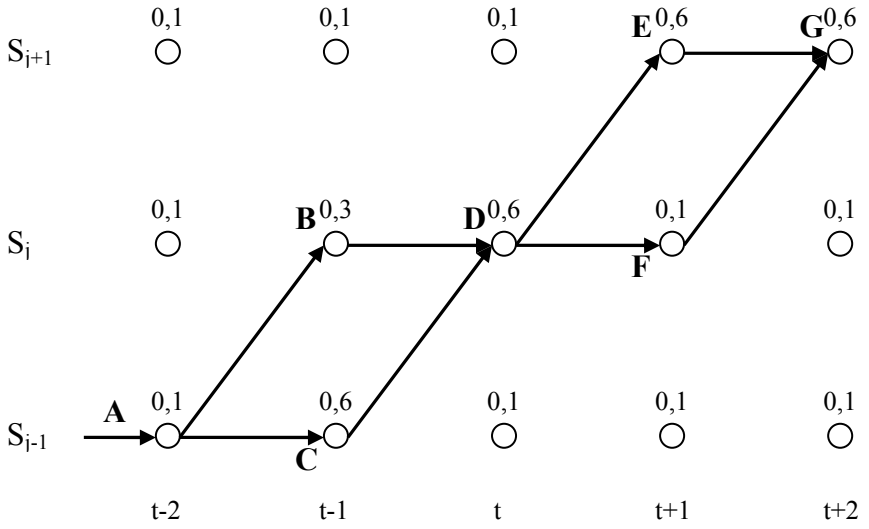


Рис. 2. Пример декодирования состояний модели на основе СММ

Нашим решением является введение информации о длительности состояния. Таким образом, введение явным образом информации о длительности накладывает дополнительное ограничение на длительность пребывания модели в одном состоянии $d_{\min} \le d \le d_{\max}$. При наложении такого ограничения использование вероятности только предыдущего состояния приводит к некорректной работе алгоритма Витерби.

Рассмотрим пример еще раз. Предположим, что на длительность состояния S_j наложено ограничение $d_j \ge 2$. Тогда единственно возможным переходом из состояния S_j в момент времени $t+1$ будут переход в то же состояние. Вследствие этого к моменту $t+2$ будет пройдена траектория ACDFG и накоплена вероятность $p = 0{,}6*0{,}6*0{,}6*0{,}6*0{,}1 = 0{,}01296$, которая отличается от указанной ранее. Для устранения этой проблемы нами предложен «модифицированный» алгоритм Витерби.

Модифицированный алгоритм Витерби

В предлагаемой модификации алгоритма Витерби максимизируется не предыдущее состояние, а длительность текущего. Для этого введем величину $\varphi_j^t(d)$, которая представляет собой вероятность того, что состояние j окончилось в момент времени t, и началось в момент времени $t-d$, т.е. имело продолжительность d:

$$\varphi_j^t(d) = \begin{cases} \prod_{t^j-t-d}^{t} b_j(o_t) * (p_j(d))^{\alpha} * \delta_{j-1}(t-d) & 1 < j \le N \\ \prod_{t^j-t-d}^{t} b_j(o_t) * (p_j(d))^{\alpha} & j = 1, \end{cases}$$

где $(p_j(d))$ – плотность вероятности длительности состояния; α – масштабный коэффициент для длительностей состояний; $\delta_{j-1}(t-d)$ – вероятность, накопленная к моменту $t-d$.

Введем также дополнительную переменную $Dj(t)$, в которой будем сохранять наиболее вероятную длительность состояния S_j, при условии его окончания в момент времени t. Тогда полную процедуру, требуемую для определения последовательности состояний, можно сформулировать следующим образом.

(1) Инициализация:

Для всех состояний $j = 1 \ldots N$:

$$\delta_j(1) = \begin{cases} b_j(o_1) & j = 1 \\ 0 & j > 1 \end{cases},$$

$Dj(1) = 0$.

(2) Рекурсия:

Для всех состояний $j = 1 \ldots N$:

Если $j > 1$:

$\delta_j(t) = \max_{t_s \le d \le t_e}(\varphi_j^t(d))$,

где $t_s = \max(1, t - d_{j\max})$, $t_e = \max(1, t - d_{j\min})$, $D_j(t) = \arg\max_{t_s \le d \le t_e}(\varphi_j^t(d))$.

Если j = 1:

$\delta_j(t) = \varphi_j^t(t)$,

$Dj(t) = t$.

(3) Окончание:

Итоговая вероятность:

$P(O) = \max_j(\varphi_j(T))$

Последнее состояние:

$q(T) = \arg\max_j(\varphi_j(T))$

(4) Восстановление пути:

$t = T$,

$q = q(t)$,

$d = D_q(t)$.

Пока $d \neq 0$:

$q_{t-d} \ldots q_t = q$,

$q = q - 1$,

$d = D_q(t)$.

Заключение

В связи со стремлением человечества к более универсальному и естественному способу взаимодействия с ЭВМ появились и продолжают развиваться различные системы распознавания речи. Для них показатели точности и производительности являются очень важными. Одним из важнейших является точность распознавания слитной речи, которая определяется отношением количества верно распознанных слов к сумме всех распознаваемых слов, пропущенных слов и лишних (неверно распознанных) слов. В настоящее время этот показатель стремиться к значению 95 % .

Некоторые специалисты считают, что за прошедшие десятки лет исследователи недалеко продвинулись. Другие считают, что задача уже практически решена. А в рамках проекта «Super Human Speech Recognition» IBM надеется к 2010 г. разработать коммерческие системы, преобразующие речь в печатный текст точнее, чем человек [7].

В ходе исследования нами представлен краткий обзор существующих алгоритмов распознавания речи, дано небольшое сравнение алгоритмов, основанных на СММ. При этом нужно учитывать, что каждая модель и каждый алгоритм подходят к конкретным задачам, и выбор необходимого делается на основании детального исследования.

Также нами кратко показана схема работы алгоритма Витерби и произведена его модификация, позволяющая максимизировать точность распознавания за счет введения дополнительной информации о длительности состояния.

Литература

1. Рабинер Л.Р., Шафер Р.В. Цифровая обработка речевых сигналов: пер. с англ./ Под ред. М.В. Назарова и Ю.Н. Прохоров – М.: Радио и связь, 1981. – 496 с.
2. Tebelskis J. Speech Recognition using Neural Networks. School of Computer Science Carnegie Mellon University. – 1995. – 190 p.
3. Иконин С.Ю., Сарана Д.В. Система автоматического распознавания речи SPIRIT ASR Engine // Цифровая обработка сигналов. – 2003. – № 4.
4. Леонович А.А. Проблемы распознавания слитной речи // Цифровая обработка сигналов. – 2007. – № 4.
5. Гультяева Т.А. Скрытые марковские процессы. – Новосибирск: Изд-во НГТУ.
6. Rabiner L.R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition // Proceeding of the IEEE. – 1989. – Vol. 99, no. 2. – P. 257–285.
7. Мазуренко И.Л. Компьютерные системы распознавания речи // Интеллектуальные системы. Москва. – 1998. –Т. 3. – Вып. 1–2.

ПОСТРОЕНИЕ СИСТЕМЫ ХРАНЕНИЯ КЛАСТЕРНОЙ АРХИТЕКТУРЫ

С.В. Богатырев
Научный руководитель – д.т.н., профессор А.А. Ожиганов

Рассмотрено построение компьютерных систем хранения кластерной архитектуры и влияние структуры коммуникационной подсистемы на отказоустойчивость распределенной системы хранения.

Необходимость повышения надежности, отказоустойчивости и производительности информационных систем привела к кластеризации компьютерных систем при распределении запросов с балансировкой нагрузки и адаптацией к отказам узлов. Под кластером понимается группа независимых компьютерных узлов (серверов, систем хранения и др.), совместно выполняющих общий набор приложений (функциональных задач), воспринимаемая внешними клиентами (источниками запросов) как единый ресурс. Все поступающие в кластер запросы по мере возможности равномерно распределяются между компьютерами (узлами) кластера, снимая риск их перегрузки.

Проблема создания высокопроизводительной и в то же время высоконадежной распределенной системы хранения данных продолжает оставаться актуальной, несмотря на прогресс, достигнутый в последние годы в решении целого ряда задач, связанных с распределенными системами хранения данных. Существующие технологии построения распределенных сетей хранения данных предлагают различные по своей сути подходы, в том числе SAN (Storage Area Network) и NAS (Network Attached Storage).

Технология построения распределенного хранилища данных должна обладать целым рядом свойств, среди которых, в первую очередь, выделяются:

- сетевая прозрачность – клиенты должны иметь возможность обращаться к удаленным файлам, пользуясь теми же самыми операциями, что и для доступа к локальным файлам;
- прозрачность размещения – имя файла не должно определять его местоположения в сети;
- независимость размещения – имя файла не должно меняться при изменении его физического месторасположения;
- мобильность пользователя – пользователи должны иметь возможность обращаться к разделяемым файлам из любого узла сети;
- устойчивость к отказам и сбоям – система должна продолжать функционировать при неисправности отдельного компонента (сервера или сегмента сети). Однако это может приводить к деградации производительности или к исключению доступа к некоторой части файловой системы;
- масштабируемость – система должна обладать возможностью масштабирования в случае увеличения нагрузки. Кроме того, должна существовать возможность постепенного наращивания системы путем добавления отдельных компонентов;
- мобильность файлов – должна быть возможность перемещения файлов из одного месторасположения в другое на работающей системе.

Распределенные файловые системы обладают рядом существенных недостатков, среди которых можно выделить:

- необходимость синхронизировать изменения между всеми узлами;
- большие накладные расходы на управление распределенными блокировками;
- проблема согласованности кешей на узлах сети;
- дорогая операция записи для сохранения целостности данных.

Для корпоративных систем и большинства Интернет-проектов характерно наличие фиксированных данных (fixed content) – неизменяемого информационного наполнения. В это понятие включается любая оцифрованная информация, сохраненная для последую-

щего использования, но не изменения – PDF-документы, электронная почта, контрольные изображения, потоковое видео и аудио, рентгеновские снимки, чертежи САПР и т. п. Чтобы эффективно управлять такими данными, необходим иной способ их размещения и извлечения. Таким способом является CAS – Content Addressed Storage [1].

CAS-системы для хранения и поиска объектов и доступа к данным используют адрес фиксированного контента, а не имя файла или его физическое местоположение. Поскольку зачастую накопление подобной информации никак не ограничивается, CAS для хранения такой информации обеспечивает простое масштабирование при сохранении времени доступа.

Типичной для CAS архитектурой является RAIN (Redundant Arrays of Independent Nodes). Системы состоят из независимо функционирующих узлов хранения. Каждый узел содержит дисковый массив, центральный процессор, память и соединения Ethernet, служащие магистралями связи внутри модуля. Каждый узел, также как и подключаемые к сети устройства хранения, работает со своей собственной операционной системой. RAIN – это своего рода реализация RAID на узлах, а не на дисковых массивах. Среди недостатков такой архитектуры можно выделить наличие узлов доступа, пропускающих через себя данные с узлов хранения. Это создает потенциальные узкие места в потоке данных в системе, снижает ее надежность.

Для организации высоконадежной системы хранения кластерной архитектуры, лишенной узких мест в узлах доступа, предлагается использовать архитектуру RADOS (Reliable, Autonomic Distributed Object Store). RADOS подразумевает разделение кластера на узлы хранения и управляющие узлы, при этом клиенты получают данные непосредственно с узлов хранения. Особенностью такого подхода является то, что управляющие узлы не содержат индекса данных, а занимаются постоянным мониторингом состояния кластера и поддержанием актуальности карты кластера. Размещение данных происходит независимо, без регистрации в каком-либо централизованном каталоге. Функция размещение опирается на карту сети и, используя заданные параметры хранения данных, возвращает узлы хранения, на которых может быть размещена информация. При отказе узла это событие учитывается в актуальной карте сети, что позволяет логически заменить вышедший из строя узел, сохраняя работоспособность системы.

Надежность и отказоустойчивость системы хранения кластерной архитектуры определяется многими факторами, в том числе:

- организацией размещения данных;
- организацией распределения запросов к данным;
- организацией восстановления информации после отказов и сбоев систем хранения;
- структурой коммуникационной подсистемы.

В данной работе проанализированы некоторые варианты коммуникационной подсистемы с точки зрения отказоустойчивости систем хранения.

Выбор рациональных вариантов построения коммуникационной подсистемы должен основываться на построении модели надежности и ее расчете. При простейшем одноуровневом построении коммуникационной подсистемы ее расчет может быть проведен с учетом надежности резервированных коммутационных узлов и цепей их связи с компьютерами и системами хранения [3]. Цепи связи включают порты коммутаторов и сетевые адаптеры (СА) компьютерных узлов хранения и обработки данных.

Для одноуровневой телекоммуникационной подсистемы оценим ее надежность (вероятность правильного функционирования) при условии обеспечения связи не менее чем между a_1 исправными узлами первой группы, включающей m_1 компьютерных узлов, и хотя бы a_2 исправными узлами из m_2 узлов второй группы (узлы хранения) при работоспособности связей хотя бы через a_3 из m_3 коммутаторов.

В предположении экспоненциального распределения времени между отказами вероятность работоспособности компонент сети определяется как

$p_1(t) = \exp(-\lambda_1 t)$, $p_2(t) = \exp(-\lambda_2 t)$, $p_3(t) = \exp(-\lambda_3 t)$, $p_a(t) = \exp(-\lambda_a t)$,

при этом $\lambda_1, \lambda_2, \lambda_3, \lambda_a$ – суммарные интенсивности отказов соответственно узла первой и второй групп, коммутатора и СА.

Точная оценка надежности исследуемых систем затруднена из-за сложного комбинаторного влияния распределения отказавших СА на работоспособность структуры. Рассмотрим некоторые приближенные верхние (оптимистические) и нижние оценки надежности системы и определим их погрешности [3].

Вначале рассмотрим простейший случай – без учета влияния отказов СА на надежность системы. Вероятность работоспособности системы в этом случае определяется как

$$P_0(t) = \sum_{k_1=a_1}^{m_1} \sum_{k_2=a_2}^{m_2} \sum_{k_3=a_3}^{m_3} C_{m_1}^{k_1} C_{m_2}^{k_2} C_{m_3}^{k_3} p_1(t)^{k_1} p_2(t)^{k_2} p_3(t)^{k_3} \times$$
$$\times (1-p_1(t))^{m_1-k_1} (1-p_2(t))^{m_2-k_2} (1-p_3(t))^{m_3-k_3},$$

Полученная оценка является оптимистической (верхней оценкой). В частности, если система сохраняет работоспособность при исправности хотя бы одного узла каждого вида (a_1=1, a_2=1 и a_3=1), то

$$P_0(t) = (1-(1-p_1(t))^{m_1})(1-(1-p_2(t))^{m_2})(1-(1-p_3(t))^{m_3}).$$

При нижней оценке надежности будем считать, что в работоспособной системе должно быть, по крайней мере, a_3 исправных коммутаторов, обеспечивающих связь с a_1 исправными из m_1 узлами первой группы и a_2 исправными из m_2 узлами второй группы, причем для каждой из указанных связей СА, расположенные в соответствующих узлах, должны быть исправны. Пессимизм оценки обусловлен тем, что система может быть работоспособна, помимо учитываемых в приближении состояний, и в состояниях, при которых связь между a_1 исправными узлами первой группы и a_2 исправными узлами второй группы обеспечивается всей совокупностью коммутаторов, а не каждым из них в отдельности.

С учетом сформулированных допущений нижнюю оценку вероятности работоспособности системы вычислим как

$$P(t) = \sum_{k_1=a_1}^{m_1} \sum_{k_2=a_2}^{m_2} \sum_{k_3=a_3}^{m_3} C_{m_1}^{k_1} C_{m_2}^{k_2} C_{m_3}^{k_3} p_1(t)^{k_1} p_2(t)^{k_2} b(k_1,k_2)^{k_3} \times$$
$$\times (1-p_1(t))^{m_1-k_1} (1-p_2(t))^{m_2-k_2} (1-b(k_1,k_2))^{m_3-k_3},$$

где $b(k_1,k_2)$ – вероятность исправного состояния коммутатора с учетом надежности, сетевых адаптеров, поддерживающих его подключение не менее чем к a_1 узлам первой группы и к a_2 узлам второй группы, при условии исправности k_1 и k_2 узлов указанных групп; k_3 – число исправных коммутаторов. Вероятность $b(k_1,k_2)$ найдем по формуле:

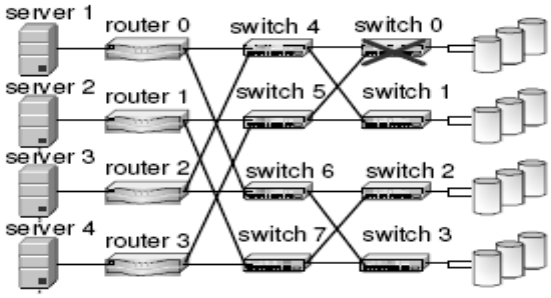
$$b(k_1,k_2) = p_3(t) \sum_{k_4=a_1}^{k_1} C_{k_1}^{k_4} p_a(t)^{k_4} (1-p_a(t))^{k_1-k_4} \times$$
$$\times \sum_{k_5=a_2}^{k_2} C_{k_2}^{k_5} p_a(t)^{k_5} (1-p_a(t))^{k_2-k_5},$$

причем k_4 и k_5 – число исправных СА, подключающих исправный коммутатор к исправным узлам первой и второй группы. Погрешность предлагаемого приближения не хуже, чем $\Delta_0(t) = P_0(t) - P(t)$.

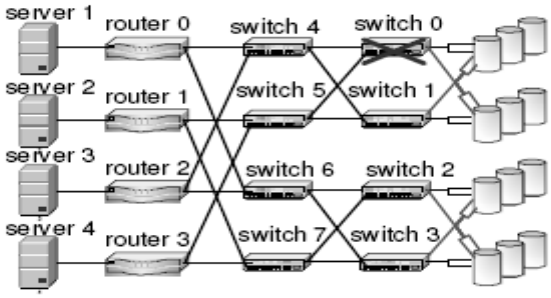
Для более сложных топологий коммуникационной подсистемы требуется анализ влияния вариантов отказов на устойчивость системы. Анализ устойчивости к отказам коммуникационной подсистемы системы хранения кластерной архитектуры, имеющей топологию «дерево», «баттерфляй» и гиперкуб, проведен в [2].

Иерархическая структура коммуникационной подсистемы «дерево» не может обеспечивать достаточную отказоустойчивость системы, при этом слабым звеном системы являются отказы узлов верхних уровней.

Топология «баттерфляй» не обеспечивают высокую надежность, потому что имеется только единственный путь с сервера на запоминающее устройство. Например, когда отказ происходит в коммутаторе 0, как показано на рис. 1(a), доступность теряется для многих запоминающих устройства. Решить эту проблему удается при добавлении резервных связей, как показано на рис. 1(b). Однако при этом возможна перегруженность коммутатора 1 и увеличение времени доставки информации



(a) A failure on switch 0 disconnects several devices.



(b) Add spare links between switches and devices.

Рис. 1. Отказы в сетевой топологии «баттерфляй»

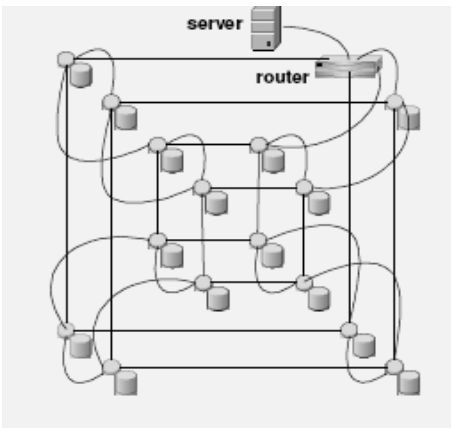


Рис. 2. Топология «Гиперкуб»

Большей отказоустойчивостью по сравнению с рассмотренными обладают структуры типа гиперкуб. Однако системная стоимость топологий типа «гиперкуб» выше, чем топологии «дерево» «баттерфляй». Устойчивость топологии «гиперкуб» к некоторым типам отказов коммуникационной подсистемы проиллюстрирована на рис. 3.

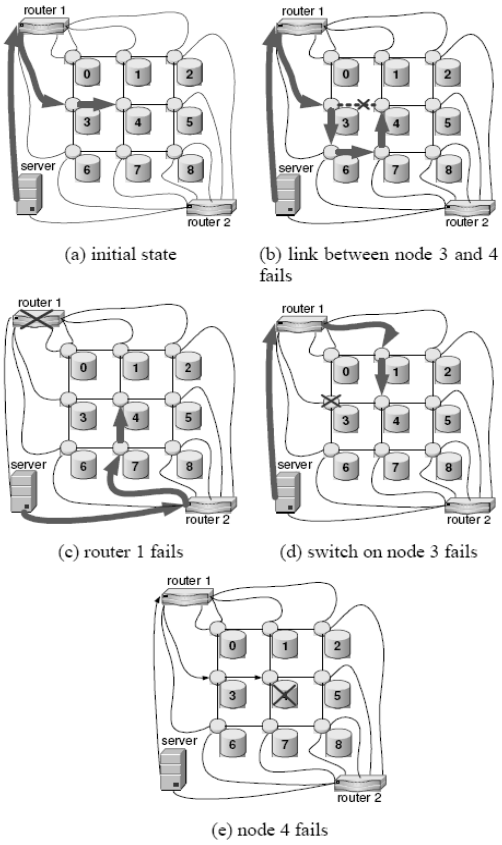


Рис. 3. Устойчивость к отказам топологии «Гиперкуб»

В качестве альтернативного варианта топологии построения коммуникационной подсистемы может рассматриваться топология типа «дерево» с резервированием коммутирующих узлов. При этом для ликвидации узких по производительности и надежности мест предлагается кратности резервирования узлов различного уровня определять на основе решения задачи векторной оптимизации.

Таким образом, проанализированы различные варианты сетевой организации систем хранения и рассмотрено влияние некоторых отказов коммуникационной подсистемы на устойчивость функционирования системы для различных топологий, что может быть использовано при построении модели надежности исследуемых систем и выборе их оптимальных вариантов построения.

Литература

1. Sage A. Weil, Andrew W. Leung, Scott A. Brandt, Carlos Maltzahn. RADOS: A Fast, Scalable, and Reliable Storage Service for Petabyte-scale Storage Clusters. – Petascale Data Storage Workshop SC07, November, 2007.
2. Qin Xin, Ethan L. Miller Impact of Failure on Interconnection Networks for Large Storage Systems // Proceedings of the 22nd IEEE / 13th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST 2005), Monterey, CA, April 2005.
3. Богатырев В.А. Надежность и эффективность резервированных компьютерных сетей // Информационные технологии. – 2006. – № 5.

НАШИ АВТОРЫ

Антонов Семен Евгеньевич – студент кафедры физики
Балакшин Павел Валерьевич – магистр кафедры вычислительной техники
Белоус Роман Олегович – аспирант кафедры вычислительной техники
Беляева Татьяна Сергеевна – аспирант Санкт-Петербургского государственного университета телекоммуникаций имени профессора М.А. Бонч-Бруевича
Богатырев Станислав Владимирович – аспирант кафедры вычислительной техники
Волков Антон Андреевич – студент Костромского государственного университета им. Н.А. Некрасова
Дёмин Сергей Анатольевич – старший преподаватель Татарского государственного гуманитарно-педагогического университета
Дзбоев Евгений Ахсарбекович – аспирант Санкт-Петербургского государственного университета сервиса и экономики
Донецкая Юлия Валерьевна – аспирант Центрального научно-исследовательского института «Электроприбор»
Жихарев Александр Геннадиевич – студент кафедры прикладной информатики
Жуань Чжипэн – аспирант кафедры вычислительной техники
Зайцева Анастасия Николаевна – студент Нижегородского филиала государственного университета – Высшей школы экономики
Залялиева Венера Мингазовна – аспирант Татарского государственного гуманитарно-педагогического университета
Зеленковский Александр Леонидович – студент кафедры компьютерных образовательных технологий
Зырянов Артем Юрьевич – студент кафедры физики
Игнатов Максим Сергеевич – аспирант кафедры информатики и прикладной математики
Клименков Сергей Викторович – ассистент кафедры вычислительной техники
Ковязина Динара Раисовна – аспирант кафедры вычислительной техники
Корнев Дмитрий Васильевич – студент Уральского государственного университета имени А.М. Горького
Коченятов Григорий Алексеевич – студент Ярославского государственного университета имени П.Г. Демидова
Крысина Ирина Вадимовна – студент Нижегородского филиала государственного университета – Высшей школы экономики
Лукичев Александр Николаевич – аспирант кафедры вычислительной техники
Лядов Максим Алексеевич – студент Тамбовского государственного технического университета
Максимов Андрей Николаевич – магистр кафедры вычислительной техники
Миненко Максим Иванович – соискатель Московского физико-технического института (государственного университета)
Набатов Роман Александрович – аспирант Костромского государственного технологического университета
Невдах Марина Михайловна – аспирант Белорусского государственного технологического университета
Никитин Анатолий Евгеньевич – студент Ярославского государственного университета имени П.Г. Демидова
Новосельский Вениамин Борисович – аспирант кафедры информатики и прикладной математики

Ольшевская Анастасия Владимировна – студентка кафедры компьютерных образовательных технологий
Петров Евгений Владимирович – аспирант кафедры вычислительной техники
Платунова Светлана Михайловна – аспирант кафедры вычислительной техники
Рубинов Константин Викторович – аспирант кафедры компьютерных технологий
Сахаров Илья Евгеньевич – аспирант Костромского государственного технологического университета
Северов Дмитрий Станиславович – старший преподаватель Московского физико-технического института (государственного университета)
Седова Яна Анатольевна – студентка Астраханского государственного технического университета
Семенова Светлана Валерьевна – студентка Тамбовского государственного технического университета
Сутковой Сергей Игоревич – студент Ярославского государственного университета имени П.Г. Демидова
Токалов Никита Сергеевич – студент кафедры информационных систем
Трифонов Сергей Владимирович студент Московского физико-технического института (государственного университета)
Труфанов Денис Николаевич – магистр Тамбовского государственного технического университета
Фролов Сергей Владимирович – аспирант Тамбовского государственного технического университета
Фролова Мария Сергеевна – студентка Тамбовского государственного технического университета
Харитонова Анастасия Евгеньевна – соискатель кафедры вычислительной техники
Хорунжий Михаил Дмитриевич – аспирант Санкт-Петербургского государственного университета кино и телевидения
Чернятина Юлия Александровна – студентка кафедры вычислительной математики
Шпаковский Юрий Францевич – ассистент Белорусского государственного технологического университета
Щекочихин Олег Владимирович – аспирант Костромского государственного технологического университета

СОДЕРЖАНИЕ

Научно-технический вестник СПбГУ ИТМО. Выпуск 46. **ИНФОРМАЦИОННЫЕ И ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ** / Главный редактор д.т.н., проф. В.О. Никифоров. – СПб: СПбГУ ИТМО, 2008. – 247 с.

НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК СПбГУ ИТМО
Выпуск 46

ИНФОРМАЦИОННЫЕ
И ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ/

Главный редактор
доктор технических наук, профессор
В.О. Никифоров
Дизайн обложки В.А. Петров, А.А. Колокольников
Редакционно-издательский отдел СПбГУ ИТМО
Зав. РИО Н.Ф. Гусарова
Лицензия ИД № 00408 от 05.11.99.
Подписано в печать 10.04.08.
Заказ 1186. Тираж 100 экз.