

ГОУ ВПО «Донецкий национальный технический университет»
Факультет «Компьютерные науки и технологии»
Кафедра «Программная инженерия»

I Международная научно-практическая конференция

**Программная инженерия:
методы и технологии разработки
информационно-вычислительных систем
(ПИИВС - 2016)**



Сборник научных трудов

Донецк 2016

Конференция ПИИВС - 2016

УДК 004.4
ББК 32.81
П78

Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2016): сборник научных трудов I научно-практической конференции. 16-17 ноября 2016 г. – Донецк, ГОУ ВПО «Донецкий национальный технический университет», 2016. – 307 с.

Цель научно-практической конференции: создание условий для обмена новыми идеями, технологиями и результатами между профессионалами программной инженерии, принимающими непосредственное участие в деятельности по анализу, спецификации, проектированию, разработке, сертификации, сопровождению и тестированию программного обеспечения компьютерных систем различного назначения, а также для расширения сотрудничества специалистов в области индустриального программирования с коммерческими структурами.

Основные направления работы конференции:

- современные языки и технологии программирования;
- методы и средства автоматизированного проектирования;
- информационная безопасность и защита данных;
- информационные системы и базы данных;
- компьютерная графика: синтез и обработка изображений;
- управление требованиями, тестирование и обеспечение качества ПО;
- программирование мобильных, распределённых и сетевых систем;
- интеллектуальный анализ данных, распознавание и классификация;
- прикладные интеллектуальные системы управления и принятия решений;
- компьютерное моделирование в фундаментальных и прикладных исследованиях.

Конференция организована Донецким национальным техническим университетом Министерства образования и науки ДНР. В организации конференции приняли участие: Донецкий национальный университет, Министерство связи ДНР, Ульяновский государственный технический университет, НИУ «Высшая школа экономики» (г.Москва), Российский экономический университет им. Г.В. Плеханова (г.Москва), Южный федеральный университет (г.Ростов-на-Дону, г.Таганрог), Южно-Российский государственный политехнический университет им. М.И. Платова (г.Новочеркасск).

В сборнике научных трудов представлены доклады учёных высших учебных заведений и государственных НИИ, ведущих специалистов из России, ДНР и ЛНР, работающих в области инженерии программного обеспечения.

УДК 004.4
ББК 32.81

ГОУ ВПО «Донецкий национальный технический университет», 2016

ОГЛАВЛЕНИЕ

СЕКЦИЯ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И БАЗЫ ДАННЫХ»

Шестаков В.К.	6
Способ построения информационных Wiki-систем на базе онтологий	
Федяев О.И.	12
Верификация циклических программ с массивами данных	
Костюк А.И., Лебедева Е.О.	21
База данных учета кадров организации	
Черкасов В.А., Азаров В.Н.	29
Технологии и системы электронного обучения	
Семисалов С.Я., Бакланов Н.А., Гурин А.Г., Тырса С.В.	35
Программный комплекс диагностики заболевания позвоночника и спинного мозга на основе методов термометрии	
Аноприенко А.Я.	41
Программная инженерия и обобщенный закон Мура	
Борота А.В., Борота А.А., Кухто А.П., Базиян-Кухто Н.К., Федяев О.И., Бакаленко В.С.	48
Информационная система для анализа результатов лечения пациентов с воспалительными заболеваниями толстой кишки	

СЕКЦИЯ «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И ЗАЩИТА ДАННЫХ»

Медгаус С.В., Чернышова А.В.	56
Обфускатор программного кода языка JavaScript	
Калинина Е.С., Чернышова А.В.	65
Разработка модуля аутентификации PDM-подсистемы в САД-компонентах САПР с открытым кодом	
Скороход С. В., Касьянов О. А.	72
Анализ средств JPWL в условиях зашумленности канала передачи данных	

СЕКЦИЯ «КОМПЬЮТЕРНАЯ ГРАФИКА: СИНТЕЗ И ОБРАБОТКА ИЗОБРАЖЕНИЙ»

Минков О.В., Ручкин К.А.	82
Распознавание окружностей на изображении с помощью алгоритма BFOA	

СЕКЦИЯ «МЕТОДЫ И СРЕДСТВА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ»

Бочков С.И., Степанов А.С.	87
Разработка и исследование виртуальных рабочих мест в среде OpenSim	
Курейчик Л.В., Курейчик В.В., Курейчик Вл.Вл.	93
Интегрированная инструментальная подсистема генетического поиска	
Лебедев Б.К., Лебедев О.Б., Лебедева Е.М.	97
Использование метода муравьиной колонии для построения кратчайших связывающих сетей	
Мальчева Р.В., Авксентьева О.А.	102
Разработка виртуальной лаборатории для изучения и моделирования архитектур	

процессорных элементов	
Григорьев А.В.	108
Формальные грамматики в семиотической концептуальной модели предметной области.	
Морозова О.В.	135
Механизм срабатывания модулей знаний интеллектуальной оболочки	
СЕКЦИЯ «ПРИКЛАДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИНЯТИЕ РЕШЕНИЙ»	
Дородных Н.О., Юрин А.Ю.	143
Прототипирование продукционных баз знаний на основе диаграмм классов	
Виноградов Г.П., Виноградова Н.Г.	151
Моделирование принятия субъективно рациональных решений	
Косенко Е. Ю., Синявская Е.Д., Косенко О.В.	160
Разработка структуры системы автоматического управления роботом-манипулятором для малоинвазивной хирургии	
Виноградов Г.П., Кузнецов В.Н.	168
Согласованная оптимизация целевой программы развития промышленного предприятия	
Федяев О.И., Зудикова Ю.В.	178
Агентно-ориентированное программирование задач распределённого искусственного интеллекта.	
СЕКЦИЯ «ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ, РАСПОЗНАВАНИЕ И КЛАССИФИКАЦИЯ»	
Коломойцева И.А., Завадская Т.В.	189
Компьютерный тезаурус для метапоисковой ГРИД-системы «Программирование для ОС «Android»	
Серёженко О. А., Коломойцева И. А.	194
Применение метапоиска к решению задач поиска цитат	
Логинов О.А., Кулиев Э.В., Страхов Н.Э	201
Методы интеллектуального анализа данных	
Казмина И.И.	207
Разработка подсистемы интеллектуального анализа данных учебного процесса	
Капустин Д.А., Антонюк И.С., Гречина А.А., Суворова Е.Ю., Швыров В.В.	213
Анализ основных технологий распознавания речи	
СЕКЦИЯ «КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В ФУНДАМЕНТАЛЬНЫХ И ПРИКЛАДНЫХ ИССЛЕДОВАНИЯХ»	
Бельков Д.В., Едемская Е.Н.	218
Анализ трафика сети с беспроводным доступом	
Анохина И.Ю.	227
Применение логистической регрессии и ROC – анализа при построении скоринговой модели	
Кузнецов П.А., Юдин А.В. Масло С.В., Клюковкин В.Р.	235
Математическое прогнозирование каскадных аварий электросетей	

Кравченко П. П., Пирская Л.В.	242
Особенности решения задачи определения координат летательного аппарата на основе дельта-преобразований	
Толстых В. К., Пшеничный К. А.	250
Численное моделирование тепловых процессов в теплообменных аппаратах с противоточными теплоносителями	
Сложеницын А.В., Землянская С.Ю., Андриевская Н.К.	255
Выбор модели для компьютеризированной системы прогнозирования поведения групп лиц	
Андрюхин А.И.	262
Интервальные оценки параметров хаотических чипов	
Филер З.Е., Андрюхин А.И.	269
Компьютерные расчёты и визуализации характеристик устойчивости систем: проблема центра-фокуса и гармонические функции	
Аверин Г.В., Звягинцева А.В.	275
О мере количественной информации в пространствах состояний сложных систем	
 СЕКЦИЯ «СОВРЕМЕННЫЕ ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ»	
Ткачёв Н.М., Андрюхин А.И.	284
Проектирование и разработка компилятора C-подобного языка программирования общего назначения с поддержкой исполнения кода на этапе компиляции	
Сидоренко А.С., Коломойцева И.А.	290
Анализ существующих подходов к разработке программ для анализа прогноза погоды	
Демчук В.Ю.	301
Языки и средства web-программирования	

СЕКЦИЯ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ И БАЗЫ ДАННЫХ»

УДК 004.822

Способ построения информационных Wiki-систем на базе онтологий

В.К. Шестаков

Институт систем информатики имени А.П. Ершова СО РАН
shestakov@iis.nsk.su

Шестаков В.К. Способ построения информационных Wiki-систем на базе онтологий. В статье описываются инструментальные средства, поддерживающие технологию построения информационных систем на основе Wiki-технологии и онтологий предметных областей. Технология позволяет строить Wiki-системы с хорошей структурой и согласованной системой понятий.

Ключевые слова: онтология, Wiki, семантические технологии, построение информационных систем.

Введение

Для удовлетворения все возрастающих информационных потребностей пользователей разработаны разнообразные средства построения информационных систем. Одним из таких удобных и простых в использовании средств сбора и хранения информации являются Wiki-системы [1]. Они позволяют работать не только с текстовым, но и с мультимедийным контентом, имеют удобный и интуитивно понятный интерфейс, просты в освоении. Однако их большим недостатком является то, что они позволяют отслеживать в создаваемых информационных системах только структурную целостность ссылок, не обеспечивая при этом логической непротиворечивости и семантической согласованности [2] используемых в них понятий и отношений. Общая идея предлагаемого подхода состоит в создании инструментария, который бы обеспечивал заполнение Wiki-систем содержимым с согласованной системой понятий (семантически согласованных Wiki). Wiki с такими свойствами можно получить, если строить ее на основе логически непротиворечивой онтологии, описывающей предметную область будущей системы.

Обзор средств создания информационных Wiki-систем с использованием семантических технологий

На сегодняшний день существует достаточно много проектов создания информационных Wiki-систем, так или иначе связанных с семантическими технологиями. Хотя эти проекты нацелены на решение различных задач, общим для них является использование онтологий на тех или иных этапах построения или функционирования информационных систем. Так как на данный момент существует большое количество средств создания Wiki-систем, то в этом обзоре мы сосредоточим наше внимание только на тех из них, которые в той или иной мере используют семантические технологии.

Сначала рассмотрим проект из Санкт-Петербурга [3], в котором в одном из первых начали применяться онтологии при построении таких информационных систем, как корпоративные Wiki-порталы. Исходя из того, что онтологии позволяют создавать более концептуально-ясные и целостные модели предметных областей, разработчиками было решено привязать контент Wiki-системы к онтологии, чтобы избежать проблем с плохой структурой портала, т.к. она в этом случае будет такого же качества, как и определяющая ее онтология. В рамках этого проекта был разработан прототип инструментария OntolingeWiki, который содержит внутри себя Wiki-движок и для каждого концепта онтологии автоматически создает Wiki-страницу, в которую эксперт вносит информацию об описываемом понятии. При этом конечные пользователи видят онтологию, представленную в виде графа, могут осуществлять навигацию по ее понятиям и видеть содержимое Wiki-страниц, соответствующих выбранному понятию.

Недостатком OntolingeWiki является то, что онтология, лежащая в основе, созданной с помощью нее Wiki-системы, не может быть изменена или дополнена средствами самой Wiki-системы.

Наиболее распространенным инструментальным средством для создания Wiki-систем является популярный Wiki-движок MediaWiki [4]. Именно для этого универсального движка больше всего строится расширений, позволяющих использовать средства семантических технологий. Одной из первых была создана надстройка над MediaWiki, получившая название Semantic MediaWiki [5]. Это расширение допускает возможность импорта онтологии [6] и позволяет использовать в Wiki-системе структуры и связи, определенные в ней. Однако предоставляемый Semantic MediaWiki модуль импорта онтологий на данный момент находится в стадии бета-версии и не включен в состав новых версий данного расширения. В связи с этим реализацию импорта онтологий пользователи Semantic MediaWiki должны выполнять самостоятельно.

Другим расширением MediaWiki является BOWiki [7] — основанный на онтологии семантический Wiki-движок. Первоначально он базировался на Semantic MediaWiki, а потом был реализован как «чистое» расширение MediaWiki. Достоинством BOWiki является то, что он позволяет пользователям рассматривать сущности, описанные Wiki-страницами, в качестве экземпляров онтологических категорий, определять новые отношения в Wiki-контенте, связывать Wiki-страницы, в том числе n-арными семантическими отношениями, запрашивать Wiki-страницы, удовлетворяющие определенным условиям, допускает импорт еще нескольких био-онтологий для ограниченного использования и обеспечивает экспорт Wiki-контента в OWL-представление.

BOWiki разрабатывался как средство для совместного создания и интеграции знаний в области биоинформатики, поэтому его функционирование основано на онтологии высокого уровня General Formal Ontology (GFO) [8] в OWL-версии. В своей работе он использует также две онтологии, основанные на GFO, — базовую биомедицинскую онтологию (GFO-Bio) и онтологию функций (Ontology of Functions или OF). Как было сказано выше, возможность использования других (внешних) онтологий существенно ограничена, поэтому BOWiki не может рассматриваться в качестве инструмента для построения Wiki-систем произвольного вида.

Также для MediaWiki существуют расширения, работающие совместно с Semantic MediaWiki, и добавляющие дополнительную функциональность по импорту семантической информации. Одним из таких является RDFIO [9]. Оно позволяет импортировать произвольные RDF-триплеты [10]. Но это расширение до сих пор имеет статус бета-версии и еще не готово для массового использования. Еще одним подобным расширением является LinkedWiki [11]. Оно применяется для получения данных с внешнего сервиса и отображения этих данных в Wiki-системе. Это расширение не может изменить структуру существующей системы, а только дополняет ее, поэтому имеет довольно ограниченную область применения.

Основные компоненты Wiki-систем и средства для работы с ними

В основе любой Wiki-системы лежит так называемый Wiki-движок — комплекс программных средств для преобразования Wiki-разметки в код, предназначенный для отображения в браузере. Одним из самых распространенных движков является MediaWiki (на его основе работает широко известная Википедия). Для него существуют специальные дополнения, называемые расширениями, позволяющие получить определенную функциональность. Так, расширение Semantic MediaWiki дает возможность добавлять семантическую информацию за счет расширения разметки, а также предлагает средства для работы с этой информацией.

Для проведения работ технического характера в Wiki-системах используют ботов — специальные программы для выполнения заданного набора операций. Они являются клиентскими приложениями, поэтому не требуют внесения изменений на стороне сервера (т.е. для их работы не нужно модифицировать движок или ставить какое-либо расширение). Например, в Википедии ботов используют для таких задач, как переименование категорий и статей, расстановка интервики ссылок (ссылок на родственные проекты), исправление ссылок, удаление спама и т.п. Для реализации ботов используются различные языки программирования, а также существуют разные библиотеки для облегчения их написания. Одной из наиболее развитых библиотек является Python WikipediaBot Framework [12]. Она использует MediaWiki API (специальный интерфейс прикладного программирования) для взаимодействия с MediaWiki-системой: авторизации, получения данных и внесения изменений.

Построение информационной Wiki-системы на основе онтологии: описание работы системы

Предлагаемая в рамках данной работы инструментальная система позволяет на основе ранее разработанной онтологии построить Wiki-систему, структура и содержание которой будет определяться этой онтологией. В дальнейшем построенная Wiki-система может расширяться как обычная Wiki-система с использованием традиционных средств Wiki-технологии.

Общая схема работы системы, реализованной в рамках предлагаемого подхода, объединяющего Wiki-технологии и онтологии, выглядит следующим образом (см. рис. 1). На первом этапе квалифицированный эксперт описывает онтологию предметной области. На втором при помощи разработанного автором инструмента (Onto2Wiki) по ней строится интерфейс на основе Wiki-технологии, состоящий из двух частей: конструкторского и пользовательского. При помощи первого эксперты предметной области могут вносить данные в систему, которые тут же становятся доступны пользователям посредством второго.

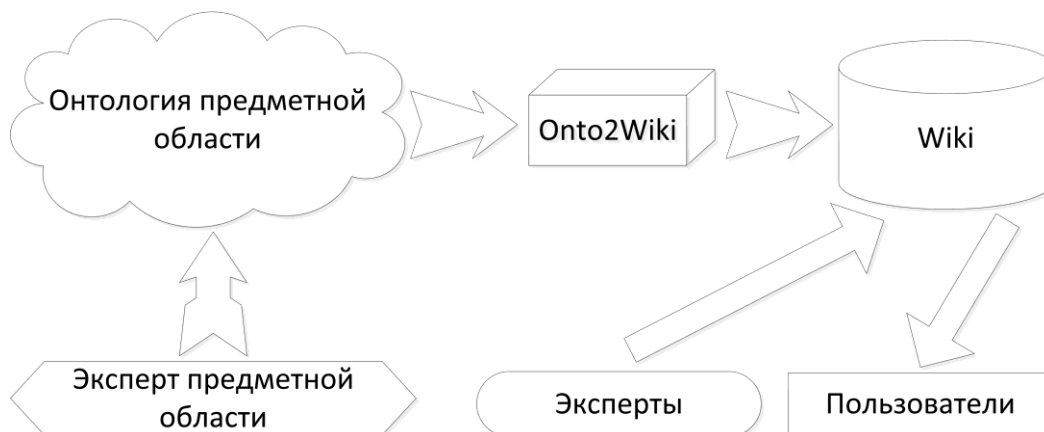


Рисунок 1 – Общая схема работы инструментальной системы

Подробная схема работы инструментальной системы представлена на рис. 2. Согласно ей разработка Wiki-системы в рассматриваемой инструментальной системе включает следующие этапы.

На первом шаге онтология, полученная в готовом виде или разработанная в Protégé [13] или каком-либо другом редакторе онтологий, сохраняется или конвертируется в OWL-формат [14]. После этого файл со спецификацией онтологии подается на вход специально разработанному в рамках данного проекта программному модулю Onto2Wiki, который разбирает его с использованием библиотеки RDFLib [15], а затем, при помощи среды Python WikipediaBot Framework, создает каркас информационной системы на основе пустого Wiki-сайта, работающего на базе MediaWiki с расширением Semantic MediaWiki. При этом в Wiki-систему добавляются нужные страницы, для которых указываются соответствующие атрибуты, расставляются категории и прописываются нужные связи. После выполнения этих действий Wiki-система готова к использованию.

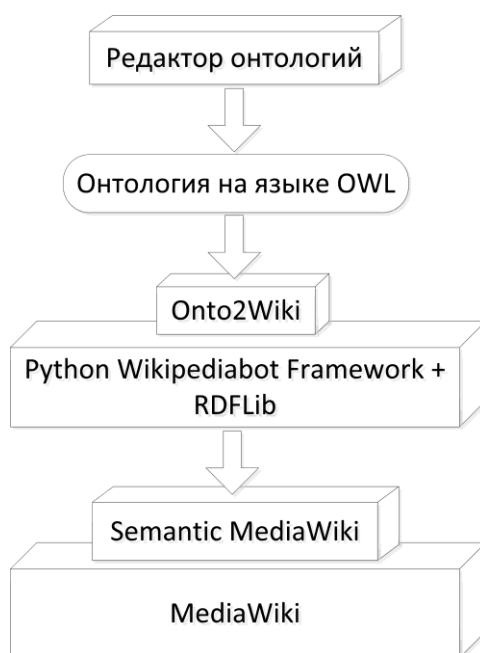


Рисунок 2 – Подробная схема работы инструментальной системы

В таблице 1 представлено соответствие между конструкциями языка OWL и семантической Wiki, которое используется при отображении онтологии в структуры и содержание Wiki-системы. В соответствии с этой таблицей для каждого класса онтологии будет создана своя категория, которая будет проставлена на каждой подкатегории, соответствующей подклассу данного класса, и на каждой странице, соответствующей экземпляру данного класса. Также все атрибуты экземпляра онтологии будут указаны в качестве атрибутов на

странице, соответствующей этому экземпляру. При этом все отношения онтологии отобразятся в типизированные ссылки [16] между соответствующими страницами и категориями.

Таблица 1. Соответствие конструкций языка OWL и Semantic MediaWiki

Конструкция OWL	Semantic MediaWiki
Класс	Категория
Подкласс	Подкатегория
Экземпляр	Страница
Атрибут	Атрибут
Отношение	Типизированная ссылка

Также при отображении онтологии ее типы данных преобразуются в типы данных Semantic MediaWiki. Пример соответствия некоторых типов данных представлен в таблице 2.

Таблица 2. Пример соответствия типов данных при отображении

Тип OWL	Тип Semantic MediaWiki
string	Строка
double	Число
boolean	Булево
dateTime	Дата

Стоит заметить, что созданную с помощью данного инструмента систему можно дорабатывать в соответствии с возникающими потребностями, взаимодействуя с ней как с обычной Wiki-системой. Т.е. можно создавать новые страницы, прописывать новые связи, указывать новые категории и т.д.

Таким образом, особенностью данного подхода является то, что полученная Wiki-система получается полностью независимой от той онтологии, на основе которой была построена.

Пример работы системы

Рассмотрим работу данной инструментальной системы на примере построения учебной информационной системы «Технология создания экспертных систем». Вот фрагмент построенной онтологии в формате Turtle:

```
ex:Как_построить_свою_ЭС rdf:type ex:Литература_по_ЭС ,
                           owl:NamedIndividual ;
ex:название_книги "Как построить свою экспертную систему" ;
ex:автор_книги "Нейлор К." ;
ex:упоминается ex:MYCIN ,
                ex:DENDRAL ,
                ex:PROSPECTOR ,
                ex:PUFF ,
                ex:Экспертные_системы .
```

Подаем эту онтологию на вход системе и после окончания ее работы получаем заполненную Wiki-систему.

На рис. 3 представлен пример одной из страниц полученной системы. На нем видны атрибуты и типизированные ссылки, находящиеся на странице.

Как построить свою ЭС

Экземпляр Как построить свою ЭС

название книги: Как построить свою экспертную систему

автор книги: Нейлор К.

упоминается: MYCIN

упоминается: DENDRAL

упоминается: PROSPECTOR

упоминается: PUFF

упоминается: Экспертные системы

Категория: Литература по ЭС

Рисунок 3 – Пример страницы

А вот так выглядит Wiki-разметка данной страницы:

```
Экземпляр Как построить свою ЭС
название книги: [[название книги::Как построить свою экспертную систему]]
автор книги: [[автор книги::Нейлор К.]]
упоминается: [[упоминается::MYCIN]]
упоминается: [[упоминается::DENDRAL]]
упоминается: [[упоминается::PROSPECTOR]]
упоминается: [[упоминается::PUFF]]
упоминается: [[упоминается::Экспертные системы]]
[[Категория:Литература по ЭС]]
```

Заключение

В данной работе рассмотрен способ построения информационных систем на основе Wiki-технологии и онтологий предметных областей. В рамках этого подхода предложен метод построения Wiki-систем на основе онтологий, разработан прототип инструментальной системы, реализующий данные методы в объеме, указанном в двух предыдущих разделах.

Применение метода построения Wiki-систем на основе онтологий позволяет строить Wiki-системы с хорошей структурой и согласованной системой понятий.

Литература

1. Leuf B., Cunningham W. The Wiki Way: Quick Collaboration on the Web. Addison-Wesley Professional, 2001. 435 p.
2. Baader F., Nutt W. Basic Description Logics. In the Description Logic Handbook, edited by Baader F., Calvanese D., McGuinness D.L., Nardi D., Patel-Schneider P.F., Cambridge University Press, 2002.
3. Муромцев Д.И., Горовой В.А., Малинин А.А., Гаврилова Т.А., Злобин А.Н., Катков Ю.В. Интеграция wiki-технологии и онтологического моделирования в задаче управления знаниями предприятия // Труды 11-ой национальной конференции по искусственному интеллекту с международным участием КИИ-2008 (г. Дубна, Россия). — М.: ЛЕНАНД, 2008. — Т.3. — С. 360-368.
4. Режим доступа: <http://mediawiki.org> — MediaWiki.
5. Режим доступа: <http://semantic-mediawiki.org> — Semantic MediaWiki.
6. Режим доступа: http://semantic-mediawiki.org/wiki/Help:Ontology_import — описание импорта в Semantic MediaWiki.
7. Backhaus M., Kelso J., Bacher J., Herre H., Hoehndorf R., Loebe F., Visagie J. BOWiki – a collaborative annotation and ontology curation framework // In Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge, СКК Banff, Canada, May 8, of CEUR Workshop Proceedings. Volume 273. Edited by Noy N., Alani H., Stumme G., Mika P., Sure Y., Vrandeic D. Aachen, Germany: CEUR-WS.org, 2007.
8. Herre H., Heller B., Burek P., Hoehndorf R., Loebe F., Michalek H. General Formal Ontology (GFO) – A foundational ontology integrating objects and processes [Version 1.0] // Onto-Med Report 8, Research Group

Ontologies in Medicine, Institute of Medical Informatics, Statistics and Epidemiology, University of Leipzig, Leipzig (2006).

9. Режим доступа: <http://www.mediawiki.org/wiki/Extension:RDFIO> — RDFIO.
10. Режим доступа: <http://www.w3.org/RDF/> — RDF.
11. Режим доступа: <http://www.mediawiki.org/wiki/Extension:LinkedWiki> — LinkedWiki.
12. Режим доступа: <http://pywikipediabot.sourceforge.net> — Python WikipediaBot Framework.
13. Режим доступа: <http://protege.stanford.edu> — Protégé.
14. Motik B., Patel-Schneider P.F., Parsia B., eds. OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax // W3C Recommendation, 27 October 2009, <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/>.
15. Режим доступа: <http://rdflib.net> — RDFLib.
16. Völkel M., Krötzsch M., Vrandečić D., Haller H. and Studer R. Semantic Wikipedia // In Proceedings of the 15th International Conference on World Wide Web (Edinburgh, Scotland, May 23–26, 2006). WWW '06. ACM Press, New York, NY, pp. 585–594, 2006, <<http://doi.acm.org/10.1145/1135777.1135863>>.

Шестаков В.К. Способ построения информационных Wiki-систем на базе онтологий. В статье описываются инструментальные средства, поддерживающие технологию построения информационных систем на основе Wiki-технологии и онтологий предметных областей. Технология позволяет строить Wiki-системы с хорошей структурой и согласованной системой понятий.

Ключевые слова: онтология, Wiki, семантические технологии, построение информационных систем.

***Shestakov V.K. Method of development information Wiki-system based on ontology.** The paper presents tools supporting technology of information system development based on Wiki-technology and subject domain ontology. The technology allows to build Wiki-systems possessing a good structure and coherent system of concepts.*

Keywords: ontology, Wiki, semantic technology, information system construction.

Верификация циклических программ с массивами данных

О.И.Федяев

Донецкий национальный технический университет
fedyaev@donntu.org

Федяев О.И. Верификация циклических программ с массивами данных. Статья посвящена проблеме формальной верификации программ. На примере доказательства правильности программы сортировки проведена апробация теоретико-функционального метода верификации. Формализм программных функций позволил математически строго преодолеть сложности доказательства правильности обработки элементов массива во вложенных циклах. Результаты статьи подтверждают возможность применения данного метода на практике в программной инженерии.

Ключевые слова: верификация программы, корректность программы, программная функция, доказательство правильности программы, теоретико-функциональный подход.

Введение

Важную роль в обеспечении качества создаваемого программного продукта играет формальная верификация программ [1,2]. Благодаря математическому доказательству корректности программного кода, т.е. его соответствия спецификации решаемой задачи, верификация повышает уровень логической строгости инженерии программного обеспечения [3,4,5].

Среди существующих подходов к формальной верификации выделяется своей оригинальностью и чёткой логикой теоретико-функциональный метод доказательства правильности структурированных программ, который может успешно применяться для проверки корректности циклических программ, использующих массивы [6].

В этом методе правильность программы определяется как соответствие между программой P и её заданной функцией f . Алгебраическая структура исходной программы P характеризует её как составную, что даёт возможность декомпозировать P на элементарные составляющие. Это позволяет задачу верификации программы P свести на основании аксиомы замещения к проверке правильности элементарных подпрограмм P_i , из которых она состоит. Верификация правильности элементарных программ без циклов осуществляется с помощью анализа соответствующих E - схем. Лемма о переходе от итеративных программ к рекурсивным позволяет свести задачу верификации циклических базовых программ P_i (типа while, until, for) к задаче верификации функционально эквивалентных программ без циклов. Для вывода программных функций $[P_i]$ используются трассировочные таблицы и разделяющиеся условные правила. В данном методе проблема перестановки элементов решается детальным развёртыванием и свёртыванием изменяющихся частей массива. Формально полная правильность программы P состоит в нахождении программной функции $[P]$ и сравнении её с заданной функцией f , т.е. в проверке равенства $f = [P]$.

В данной работе ставится задача оценить эффективность данного метода на примере формальной верификации программы сортировки по убыванию элементов массива методом выбора [7]. Алгоритм сортировки на рис.1 составлен специально так, чтобы в цикле присутствовали перестановки элементов массива, что делает задачу верификации не тривиальной. Процесс верификации программы P включает последовательность шагов доказательства правильности элементарных программ, из которых состоит P .

Шаг 1. Верификация подпрограммы типа «if-then-else»

В блок-схеме алгоритма сортировки на рис.1 первой элементарной программой (обозначим её P_1), с которой надо начать проверку правильности, является «внутренний» условный оператор (рис.2).

Поскольку функция f_1 для программы P_1 не задана, то выдвинем гипотезу о ней и запишем её в виде следующего предложения одновременного присваивания [6]

$$f_1 = (i, a_j, a_i := i + 1, \max(a_j, a_i), \min(a_j, a_i)) . \quad (1)$$

Эта запись означает, что в начале вычисляются значения всех выражений, стоящих справа от символа «:=», а затем полученные значения присваиваются соответствующим именам данных, стоящим слева. Правильность P_1 следует из доказательства равенства $f_1 = [P_1]$. Поэтому найдём программную функцию

[P₁] с помощью трассировочной таблицы.

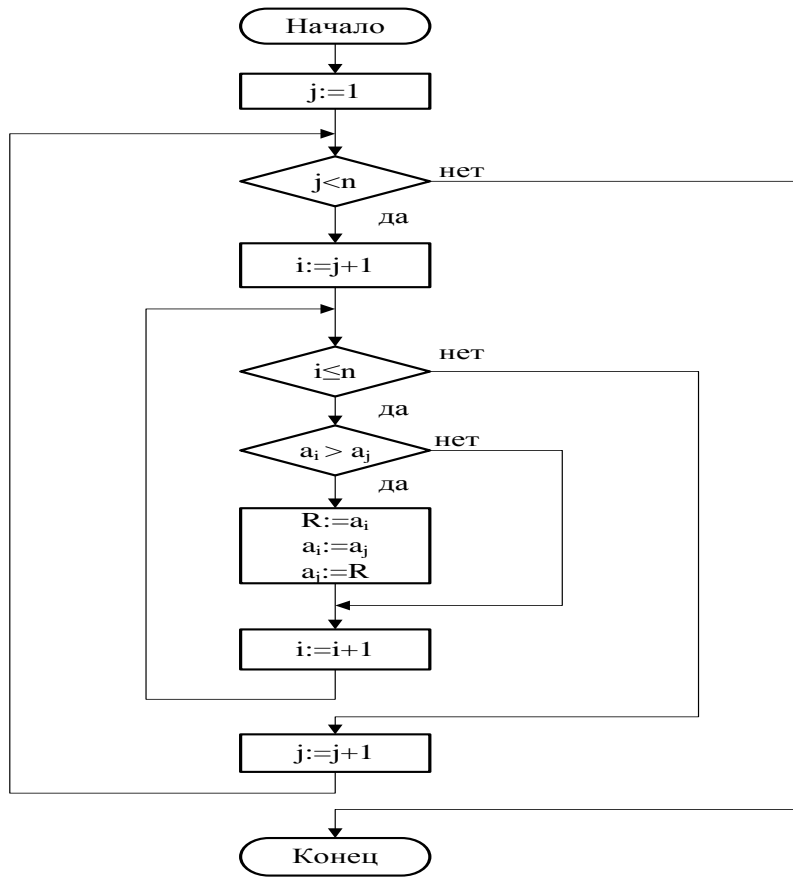


Рисунок 1 - Блок-схема программы P сортировки массива

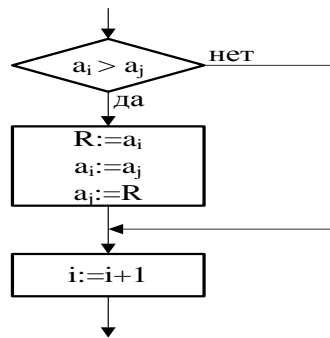


Рисунок 2 - Элементарная программа P₁ типа «if-then-else»

Трассировочная таблица позволяет формально записать вывод функции условно последовательной программы P₁. Трассировка первого пути выполнения программы P₁ показана в табл. 1.

Таблица 1. Трассировка пути выполнения P₁, когда (a_i > a_j) = true

N п/п	Фрагмент	a _j	a _i	R	i
1	$a_i > a_j$	$a_j^1 = a_j^0$	$a_i^1 = a_i^0$	$R^1 = R^0$	$i^1 = i^0$
2	$R := a_i$	$a_j^2 = a_j^1$	$a_i^2 = a_i^1$	$R^2 = a_i^1$	$i^2 = i^1$
3	$a_i := a_j$	$a_j^3 = a_j^2$	$a_i^3 = a_j^2$	$R^3 = R^2$	$i^3 = i^2$
4	$a_j := R$	$a_j^4 = R^3$	$a_i^4 = a_i^3$	$R^4 = R^3$	$i^4 = i^3$
5	$i := i + 1$	$a_j^5 = a_j^4$	$a_i^5 = a_i^4$	$R^5 = R^4$	$i^5 = i^4 + 1$

Если просматривать снизу-вверх соответствующие колонки таблицы и при этом систематически исключать все промежуточные значения верхних индексов, начиная с конечных значений, то получим программную функцию для первого пути, исключив побочное данное R:

– изменение данных

$$a_j^5 = a_j^4 = R^3 = R^2 = a_i^1 = a_i^0 ;$$

$$a_i^5 = a_i^4 = a_i^3 = a_i^2 = a_i^1 = a_i^0 ; i^5 = i^4 + 1 = i^3 + 1 = i^0 + 1 ;$$

– программная функция пути

$$(a_i > a_j) \rightarrow i, a_j, a_i := i + 1, a_i, a_j . \quad (2)$$

Выполнив аналогично трассировку второго пути выполнения программы P_1 , когда $(a_i > a_j = \text{false})$, получим другую часть программной функции, соответствующую второму пути:

$$(a_i \leq a_j) \rightarrow i, a_j, a_i := i + 1, a_j, a_i . \quad (3)$$

Объединение (2) и (3) даёт общую программную функцию $[P_1]$ в виде условного правила

$$[P_1] = (a_i > a_j) \rightarrow i, a_j, a_i := i + 1, a_i, a_j \mid$$

$$(a_i \leq a_j) \rightarrow i, a_j, a_i := i + 1, a_j, a_i .$$

которое можно записать в более компактной форме

$$[P_1] = (i, a_j, a_i := i + 1, \max(a_j, a_i), \min(a_j, a_i)) . \quad (4)$$

Сравнивая (1) и (4) видно, что $f_1 = [P_1]$ и, следовательно, высказанная гипотеза о программной функции верна.

Шаг 2. Верификация подпрограммы типа «while»

На основании аксиомы замещения [6] можно выполнить замену подпрограммы P_1 , правильность которой доказана на шаге 1, одним оператором с той же программной функцией f_1 . При этом программная функция исходной программы P не изменится. После такой подстановки выделяется следующий элементарный структурный элемент для верификации, который является циклом типа «while» (рис. 3).

Как видно из рисунка, тело цикла представлено одним оператором, который реализует функцию f_1 . Исходная функция, которую должна реализовать верифицируемая программа P_1 , не задана. Поэтому сначала сформулируем гипотезу о функции для элементарной циклической программы типа **while** (рис. 3). Для этого достаточно рассмотреть три шага работы оператора цикла, начиная с номера $i=j+1$.

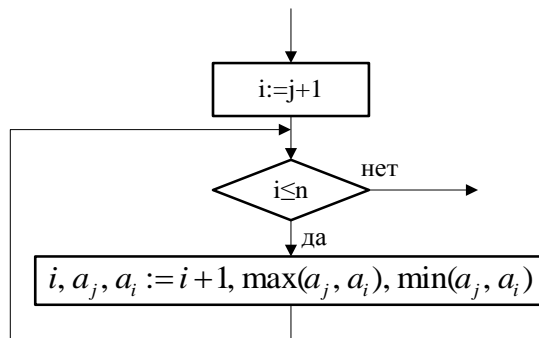


Рисунок 3 - Элементарная программа P_2 типа «while»

Перестановка элементов в группе из четырёх последовательных элементов массива



будет определяться следующими операторами одновременного присваивания, которые показывают, как изменяются значения элементов в этих позициях на каждом цикле в зависимости от исходных значений элементов перед началом циклического процесса:

$$\begin{aligned}
1 \text{ шаг: } & a_j^1, a_i^1, a_{i+1}^1, a_{i+2}^1 := \max(a_j, a_i), \min(a_j, a_i), a_{i+1}, a_{i+2} ; \\
2 \text{ шаг: } & a_j^2, a_i^2, a_{i+1}^2, a_{i+2}^2 := \max(\max(a_j, a_i), a_{i+1}), \min(a_j, a_i), \\
& \min(\max(a_j, a_i), a_{i+1}), a_{i+2} ; \\
3 \text{ шаг: } & a_j^3, a_i^3, a_{i+1}^3, a_{i+2}^3 := \max(\max(\max(a_j, a_i), a_{i+1}), a_{i+2}), \min(a_j, a_i), \\
& \min(\max(a_j, a_i), a_{i+1}), \min(\max(\max(a_j, a_i), a_{i+1}), a_{i+2}) .
\end{aligned}$$

Из этих шагов видна закономерность, которую можно записать в виде следующей функции:

$$\begin{aligned}
f_2 = (j < i < n) \rightarrow i, a_j, a(i:n) := n + 1, \max[a_j, \max[a(i:n)]], \\
\boxed{i} \quad \min[a_i, a_j], \\
\boxed{i+1} \quad \min[a_{i+1}, \max[a_j, a(i:i)]], \\
\boxed{i+2} \quad \min[a_{i+2}, \max[a_j, a(i:i+1)]], \\
\vdots \\
\boxed{n} \quad \min[a_n, \max[a_j, a(i:n-1)]] .
\end{aligned}$$

В прямоугольных рамках указаны номера элементов массива. Теперь докажем правильность следующей циклической программы

$$P_2 = \mathbf{while} \ i \leq n \ \mathbf{do} \ i, a_j, a_i := i + 1, \max(a_j, a_i), \min(a_j, a_i) \ \mathbf{od} ,$$

где ключевые слова **do** и **od** являются обычными ограничителями фрагментов программы.

Таблица 2. Трассировочная таблица для программы P_2'

Фрагмент программы	Условие	Массив a	i
1	$i_0 \leq n$	$a_1(1:n) = a_0(1:n)$	$i_1 = i_0$
2	$j < i_1 < n$	(2.1) $a_2(1:j-1) = a_1(1:j-1)$ (2.2) $a_2(j) = \max[a_1(j), a_1(i_1)]$ (2.3) $a_2(j+1:i_1-1) = a_1(j+1:i_1-1)$ (2.4) $a_2(i_1) = \min[a_1(j), a_1(i_1)]$ (2.5) $a_2(i_1+1:n) = a_1(i_1+1:n)$	$i_2 = i_1 + 1$
3	$j < i_2 < n$	(3.1) $a_3(1:j-1) = a_2(1:j-1)$ (3.2) $a_3(j) = \max[a_2(j), \max[a_2(i_2:n)]]$ (3.3) $a_3(j+1:i_2-1) = a_2(j+1:i_2-1)$ (3.4) $a_3(i_2) = \min[a_2(i_2), a_2(j)]$ (3.5) $a_3(i_2+1) = \min[a_2(i_2+1), \max[a_2(j), a_2(i_2:i_2)]]$... $a_3(n) = \min[a_2(n), \max[a_2(j), a_2(i_2:n-1)]]$	$i_3 = n + 1$

Используя лемму о рекурсивном представлении [6], заменим программу P_2 на рекурсивную нециклическую программу P_2' , которая будет иметь следующий вид:

$$\begin{aligned}
P_2' = \mathbf{if} \ i < n \\
\mathbf{then} \ i, a_j, a_i := i + 1, \max(a_j, a_i), \min(a_j, a_i); \\
(j < i < n) \rightarrow i, a_j, a(i:n) := n + 1, \max[a_j, \max[a(i:n)]], \\
\min[a_i, a_j], \\
\min[a_{i+1}, \max[a_j, a(i:i)]], \\
\min[a_{i+2}, \max[a_j, a(i:i+1)]], \\
\vdots \\
\min[a_n, \max[a_j, a(i:n-1)]]; \\
\mathbf{fi} .
\end{aligned}$$

Докажем справедливость равенства $f_2 = [P_2']$, что эквивалентно доказательству $f_2 = [P_2]$. Найдём с помощью трассировочной таблицы программную функцию $[P_2']$.

Условие:

$$\begin{aligned} & (i_0 \leq n) \wedge (j < i_1) \wedge (i_1 < n) \wedge (j < i_2) \wedge (i_2 < n) = \\ & (i_0 \leq n) \wedge (j < i_0) \wedge (i_0 < n) \wedge (j < i_0 + 1) \wedge (i_0 + 1 < n) = \\ & (j < i_0 \leq n) \wedge (j < i_0 + 1 < n) = (j < i_0 < n) \end{aligned}$$

Если проследить за изменением данных по трассировочной таблице путём систематического исключения всех промежуточных значений индексов, то получим функцию рекурсивной программы. Из формулы (3.1) в трассировочной таблице (см. табл. 2) выводим:

$$a_3(1:j-1) = a_2(1:j-1) = a_1(1:j-1) = a_0(1:j-1)$$

Таким же способом из формулы (3.2) получаем:

$$\begin{aligned} a_2(j) &= \max[\max[a_1(j), a_1(i_1)], \max[a_2(i_1+1:n)]] \\ &= \max[\max[a_0(j), a_0(i_0)], \max[a_1(i_0+1:n)]] \\ &= \max[\max[a_0(j), a_0(i_0)], \max[a_0(i_0+1:n)]] \\ &= \max[a_0(j), \max[a_0(i_0:n)]] \end{aligned}$$

Из формулы (3.3) получаем:

$$\begin{aligned} a_3(j+1:i_2-1) &= a_2(j+1:i_2-1) \\ a_3(j+1:i_1+1-1) &= a_2(j+1:i_1+1-1) \\ a_3(j+1:i_1) &= a_2(j+1:i_1) = a_1(j+1:i_1), \min[a_1(j), a_1(i_1)] \\ a_3(j+1:i_0) &= a_2(j+1:i_0) = a_0(j+1:i_0-1), \min[a_0(j), a_0(i_0)] \\ a_3(j+1:i_0-1) &= a_0(j+1:i_0-1) \\ a_3(i_0) &= \min[a_0(j), a_0(i_0)] \end{aligned}$$

Из формулы (3.4) получаем:

$$\begin{aligned} a_3(i_2) &= \min[a_2(i_2), a_2(j)] \\ a_3(i_1+1) &= \min[a_2(i_1+1), a_2(j)] \\ a_3(i_0+1) &= \min[a_1(i_1+1), \max[a_1(j), a_1(i_1)]] \\ &= \min[a_0(i_0+1), \max[a_0(j), a_0(i_0)]] \\ &= \min[a_0(i_0+1), \max[a_0(j), a_0(i_0:i_0)]] \end{aligned}$$

Из формулы (3.5) получаем:

$$\begin{aligned} a_3(i_2+1) &= \min[a_2(i_2+1), \max[a_2(j), a_2(i_2:i_2)]] \\ a_3(i_0+2) &= \min[a_2(i_1+2), \max[a_2(j), a_2(i_1+1:i_1+1)]] \\ &= \min[a_1(i_1+2), \max[\max[a_1(j), a_1(i_1)], a_2(i_1+1:i_1+1)]] \\ &= \min[a_0(i_0+2), \max[a_0(j), a_0(i_0:i_0+1)]] \quad \text{и т. д.} \end{aligned}$$

Если объединить результаты выводов по формулам (3.1) – (3.5) из табл. 2 и отбросить нулевые индексы, то получим программную функцию рекурсивной программы P_2' . Она полностью совпадает с выдвинутой гипотезой f_2 , т.е. $f_2 = [P_2']$. Принимая во внимание лемму о рекурсивном представлении программы и теорему правильности [3], можно утверждать, что циклическая программа P_2 также правильна.

Шаг 3. Верификация программы типа «последовательность»

Далее переходим к доказательству правильности следующей элементарной программы P_3 (последовательность), которая получается после применения аксиомы замещения правильного цикла **while** его функцией (рис.4). Выдвинем следующую гипотезу о функции программы P_3 :

$$\begin{aligned} f_3 = (j < n) \rightarrow i, a(j), a(j+1:n) := n+1, \\ & \max[a(j:n)], \\ & \min[a(j+1), \max[a(j:j)]], \\ & \min[a(j+2), \max[a(j:j+1)]], \\ & \min[a(j+3), \max[a(j:j+2)]], \\ & \dots \\ & \min[a(n), \max[a(j:n-1)]]. \end{aligned} \tag{5}$$

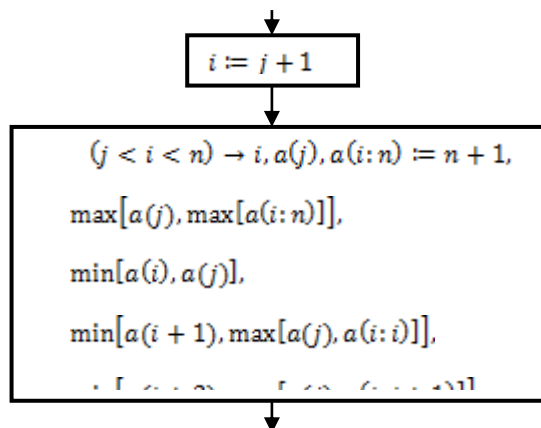


Рис. 4. Элементарная программа P₃ типа «последовательность»

Упростим запись функции f_3 , представив её в виде

$$f_3 = (j < n) \rightarrow i, a(j), a(j + 1:n) := n + 1, \max[a(j:n)], a(j + 1:n) \setminus a(j).$$

Символ «\» обозначает операцию исключения элементов. Поэтому запись $a(k:n) \setminus a(j:k)$ означает, что в массиве $a(j:n)$ среди элементов $a(k:n)$ нет элементов этого же массива, стоящих на позициях от j до k ($j < k < n$). Такое расположение элементов массива возникает после перестановки максимальных элементов, т.е. максимальные элементы «всплыли» в левую часть массива $a(j:k)$, а меньшие элементы переместились в позиции $a(k:n)$.

Представленную на рис. 4 последовательность из двух операторов можно рассматривать как трассировочную таблицу. В этом случае, если во втором операторе блок-схемы (т.е. в функции f_2) заменить i на $j+1$, то получится программная функция $[P_3]$, которая будет совпадать с f_3 . Таким образом, из равенства $f_3 = [P_3]$ следует правильность программы P₃.

Шаг 4. Верификация подпрограммы типа «while»

Последующая декомпозиция структуры исходной программы P даёт следующий простой фрагмент для верификации – цикл типа «while», представленный на рис. 5. Тело этого цикла получено на основании аксиомы замещения, путём замены правильной последовательности P₃ на функцию f_3 .

Сформулируем гипотезу для программы P₄ в виде функции f_4 :

$$\begin{aligned}
 (j < n) \rightarrow j, i, a(j:n) := n, n + 1, \max[a(j:n)], \\
 \max[a(j + 1:n) \setminus a(j)], \\
 \max[a(j + 2:n) \setminus a(j:j + 1)], \\
 \max[a(j + 3:n) \setminus a(j:j + 2)], \\
 \dots \\
 \max[a(n:n) \setminus a(j:n - 1)].
 \end{aligned}
 \tag{6}$$

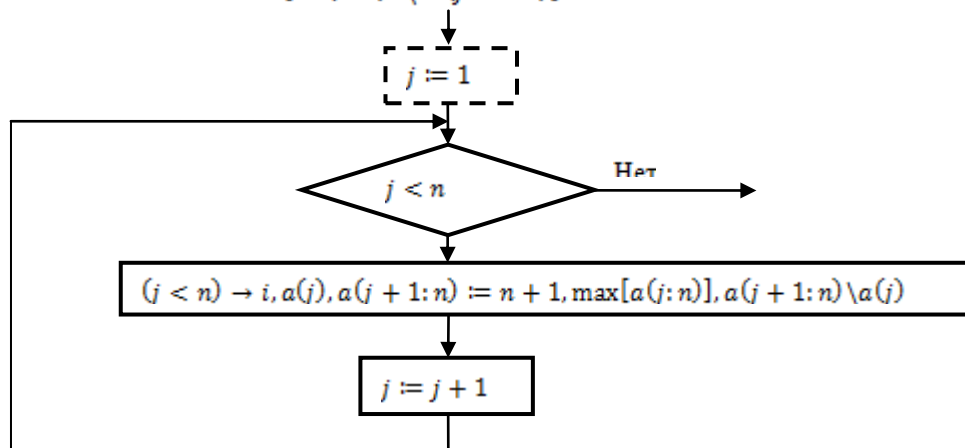


Рисунок 5 - Элементарная программа P₄ типа «while»

Текст циклической программы P_4 приведен ниже:

$$\begin{aligned}
 P_4 = & \textbf{while } j < n \textbf{ do } (j < n) \rightarrow i, a(j), a(j+1:n) := n+1, \\
 & \max[a(j:n)], a(j+1:n) \setminus a(j); \\
 & j := j+1; \\
 & \textbf{od};
 \end{aligned}$$

Далее, используя лемму о рекурсивном представлении [6], представим программу P_4 в рекурсивной ациклической форме P'_4 , что упростит получение программной функции и саму верификацию:

$$\begin{aligned}
 P'_4 = & \textbf{if } j < n \textbf{ then } (j < n) \rightarrow i, a(j), a(j+1:n) := n+1, \\
 & \max[a(j:n)], \\
 & a(j+1:n) \setminus a(j); \\
 & j := j+1; \\
 & (j < n) \rightarrow j, i, a(i:n) := n, n+1, \max[a(j:n)], \\
 & \max[a(j+1:n) \setminus a(j:j)], \\
 & \max[a(j+2:n) \setminus a(j:j+1)], \\
 & \dots \\
 & \max[a(n:n) \setminus a(j:n-1)]; \\
 & \textbf{fi}
 \end{aligned}$$

Докажем, что $f_4 = [P'_4]$. Для этого выведем из трассировочной табл.3 программную функцию $[P'_4]$. Проследим только за изменением элементов массива a и параметра j :

$$\begin{aligned}
 a_4(1:j_3-1) &= a_3(1:j_3-1) \\
 a_4(1:j_2) &= a_2(1:j_2) \\
 a_4(1:j_0) &= a_2(1:j_2) = a_2(1:j_1) = a_1(1:j_1-1), \max[a_1(j_1:n)] \\
 &= a_0(1:j_0-1), \max[a_0(j_0:n)] \\
 \\
 a_4(j_3) &= \max[a_3(j_3:n)] \\
 a_4(j_2+1) &= \max[a_2(j_2+1:n)] = \max[a_2(j_1+1:n)] \\
 a_4(j_1+1) &= \max[a_1(j_1+1:n) \setminus a(j_1)] \\
 a_4(j_0+1) &= \max[a_0(j_0+1:n) \setminus a(j_0)] \\
 \\
 a_4(j_3+1) &= \max[a_3(j_3+1:n) \setminus a(j_3:j_3)] \\
 a_4(j_2+2) &= \max[a_2(j_2+2:n) \setminus a(j_2+1:j_2+1)] \\
 a_4(j_1+2) &= \max[a_2(j_1+2:n) \setminus a(j_1+1:j_1+1)] \\
 &= \max[a_1(j_1+2:n) \setminus a(j_1) \setminus a(j_1+1:j_1+1)] \\
 a_4(j_0+2) &= \max[a_0(j_0+2:n) \setminus a(j_1:j_1+1)] \\
 &= \max[a_0(j_0+2:n) \setminus a(j_0:j_0+1)] \\
 & \dots \\
 a_4(n) &= \max[a_3(n:n) \setminus a(j_3:n-1)] \\
 &= \max[a_2(n:n) \setminus a(j_2+1:n-1)] \\
 &= \max[a_1(n:n) \setminus a(j_1) \setminus a(j_1+1:n-1)] \\
 &= \max[a_0(n:n) \setminus a(j_1:n-1)] \\
 &= \max[a_0(n:n) \setminus a(j_0:n-1)] \\
 \\
 j_4 &= n.
 \end{aligned}$$

Таблица 3. Трассировка штатного пути выполнения программы P'_4

Фрагмент программы	Условие	Массив $a(1:n)$	j
1	$j_0 < n$	$a_1(1:n) = a_0(1:n)$	$j_1 = j_0$
2	$j_1 < n$	(2.1) $a_2(1:j_1 - 1) = a_1(1:j_1 - 1)$ (2.2) $a_2(j_1) = \max[a_1(j_1:n)]$ (2.3) $a_2(j_1 + 1:n) = a_1(j_1 + 1:n) \setminus a(j_1)$	$j_2 = j_1$
3		$n) = a_2(1:n)$	$j_3 = j_2 + 1$
4	$j_3 < n$	(4.1) $a_4(1:j_3 - 1) = a_3(1:j_3 - 1)$ (4.2) $a_4(j_3) = \max[a_3(j_3:n)]$ (4.3) $a_4(j_3 + 1) = \max[a_3(j_3 + 1:n) \setminus a(j_3:j_3)]$ (4.4) $a_4(j_3 + 2) = \max[a_3(j_3 + 2:n) \setminus a(j_3:j_3 + 1)]$ \dots $a_4(n) = \max[a_3(n:n) \setminus a(j_3:n - 1)]$	$j_4 = n$

Из анализа изменения элементов массива видно, что перемещались только те элементы, которые расположены после j -го номера. Причём, функционально изменение полностью совпадают с выдвинутой гипотезой. Если рассмотреть также и другой путь выполнения программы P_4 , связанный с ложностью выполнения условия $(j_3 < n) = \text{false}$, то совпадать будут не только данные, но и предикат программной функции. Отсюда следует выполнение равенства $f_4 = [P_4]$ и правильность цикла P_4 .

Шаг 5. Верификация программы типа «последовательность»

На последнем этапе процесс верификации сводится к доказательству правильности элементарной программы (рис.6), полученной путём свёртки по аксиоме замещения последовательного ряда доказанных ранее правильных подпрограмм. Гипотезу о программной функции f_5 , реализуемую программой P_5 , представим в следующем виде:

$$f_5 = (1 < n) \rightarrow j, i, a(1:n) := n, n + 1, \max[a(1:n)],$$

$$\max[a(2:n) \setminus a(1)],$$

$$\max[a(3:n) \setminus a(1:2)],$$

$$\dots$$

$$\max[a(n:n) \setminus a(1:n - 1)].$$

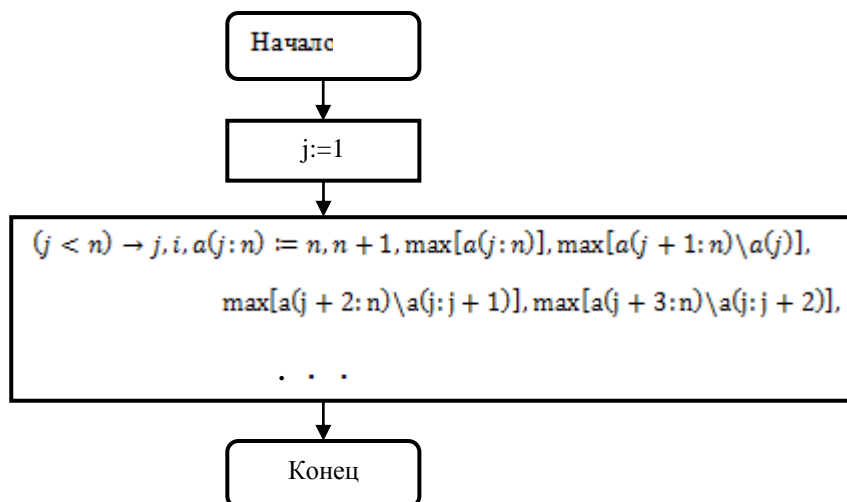


Рисунок 6 - Элементарная программа P_5 типа «последовательность»

Последовательность операторов программы, приведенная на рис.6, позволяет легко получить программную функцию $[P_5]$ без трассировочной таблицы. Для этого достаточно во втором операторе (функции f_4) заменить переменную j на константу 1. Полученная таким образом программная функция полностью совпадает с гипотезой, т.е. $f_5 = [P_5]$, что подтверждает правильность P_5 и, следовательно, правильность исходной программы P .

Поскольку значения функции \max удовлетворяют следующему неравенству

$$\max[a(1:n)] \geq \max[a(2:n) \setminus a(1)] \geq \max[a(3:n) \setminus a(1:2)] \geq \dots,$$

то можно утверждать, что элементы массива $a(1:n)$ отсортированы по убыванию правильно, т.е. справедливо $a(1:n) = \text{SORT}(a(1:n))$.

Заключение

Возрастающая сложность и важность разрабатываемых программ требует серьезного отношения к обеспечению качества производимого программного продукта. По этой причине в последние годы сделаны огромные усилия по созданию новых технологий разработки программного обеспечения (ПО), методов и инструментальных средств анализа корректности ПО [8,9]. Традиционное тестирование, опираясь на разнообразные методы и мощные средства автоматизации, однако, не могут исключить применение формальных методов доказательства корректности программ, особенно – критически важных. Наиболее перспективный путь в решении проблемы качества ПО заключается в сочетании методов тестирования и верификации программ в составе специализированных инструментальных CASE-систем [10].

В данной работе выполнена апробация формального метода верификации, основанного на понятии программной функции. На примере верификации достаточно сложной циклической программы обработки массива показана практическая возможность применения данного метода в программной инженерии.

Литература

1. IEEE 1012-2004. Standard for Software Verification and Validation. IEEE, 2005.
2. IEEE 1059-1993. Guide for Software Verification and Validation Plans. New York: IEEE, 1993.
3. Непомнящий В.А., Рякин О.М. Прикладные методы верификации программ / Под ред. А.П. Ершова. – М.: Радио и связь, 1988. – 256 с.
4. Кларк Э.М., Грамберг О., Пелед Д. Верификация моделей программ: Model Checking. Пер. с англ./ Под ред. Р.Смелянского. – М.: МЦНМО, 2002. – 416 с.
5. Вудкок Д. Первые шаги к решению проблемы верификации программ. Открытые системы. – 2006. – № 8. – С. 36-43.
6. Лингер Р. И., Миллс Х., Уитт Б. Теория и практика структурного программирования: Пер. с англ. М.: Мир, 1982. – 406 с.
7. Седжвик Р. Фундаментальные алгоритмы на C++. Анализ, структуры данных, сортировка, поиск: Пер. с англ./ Р. Седжвик. – СПб.: ООО «ДиаСофтЮП», 2002. – 688 с.
8. Beckert V., Hahnle R., Schmitt P.H., eds. Verification of Object-Oriented Software: The KeY Approach. Springer, 2007.
9. Hoare T., Misra J. Verified software: Theories, Tools, Experiments. Vision of Grant Challenge project. Microsoft Research Ltd and the University of Texas at Austin, 2005. – P. 1–43.
10. Васенин В.А. Кривчиков М.А. Языково-ориентированное программирование для формальной верификации программного обеспечения. Материалы четвертой Научно-практической конференции «Актуальные проблемы системной и программной инженерии». Сб. науч. тр. /Национальный исследовательский университет «Высшая школа экономики». – М.: Изд-во НИУ ВШЭ, 2015. – 234 с.

Fedyayev O. Verification of the cyclic program with data sets. The article is devoted to the problem of the formal verification of software. On the example of the proof of the correctness of the program sorting is carried out check method of the program functions. Formalism of the program functions allows a mathematically rigorous proof the correctness perform nested loops with arrays. The got results confirm the possibility of the proposed method application in the practice of software engineering.

Keywords: Verification of the program, program correctness, program function, proof of the program correctness, theoretical-functional approach.

База данных учета кадров организации

А.И. Костюк, Е.О. Лебедева
Южный федеральный университет
aikostyuk@sfedu.ru

Костюк А.И., Лебедева Е.О. База данных учета кадров организации. Предложена база данных учета кадров организации, автоматизирующая работу работников отдела кадров, разработанная на основе контекстной диаграммы, диаграммы декомпозиции, диаграмм логической и физической моделей базы данных. Разработанная база данных позволяет повысить эффективность автоматизации документооборота за счет существенного снижения времени, необходимого для обработки информации и оформления кадровых документов.

Ключевые слова: автоматизированное рабочее место, база данных, отдел кадров, программное обеспечение.

Введение

Одним из факторов, определяющих уровень развития современного общества и его интеллектуальные возможности, является оснащённость его средствами вычислительной техники – основы автоматизации умственной деятельности человека. Сфера использования ЭВМ в настоящее время настолько широка, что нет такой области, где применение ЭВМ было бы нецелесообразным. В настоящее время ЭВМ широко применяются во многих отраслях деятельности человека. Ни одна фирма не может обойтись в своей работе без применения компьютеров, которые с успехом заменяют рутинную работу, выполнявшуюся ранее вручную, повышая эффективность работы любой фирмы. Особенно важна роль ЭВМ для развития науки, роста промышленного производства и повышения эффективности управления.

Управление персоналом - основным ресурсом каждого предприятия - сегодня становится задачей первостепенной важности. Создание условий для привлечения и удержания лучших сотрудников, своевременная подготовка кадровых ресурсов для удовлетворения потребностей бизнеса - это необходимые составляющие успеха любой компании. Но для того чтобы иметь возможность принимать верные управленческие решения, нужно владеть актуальной, полной и достоверной информацией о персонале, который, к тому же, является наиболее непостоянным, изменчивым ресурсом предприятия. Возможности его использования зависят от многих факторов - экономических, законодательных, демографических. Чтобы анализировать ситуацию, руководитель должен знать о своих кадрах всё.

Одной из современных тенденций в применении средств автоматизации процессов управления персоналом на крупных предприятиях становится создание единой централизованной информационной системы управления персоналом, которая служит эффективным инструментом для автоматизации процедур HR-отдела, для консолидации и анализа информации о кадровых ресурсах. Единая система позволяет вводить данные однократно и в дальнейшем использовать в разных видах кадровых операций и учета. В централизованной автоматизированной системе поддерживается единое хранилище информации, что дает возможность формировать сводные отчеты и получать интересующую руководителя информацию.

Описание предметной области

Задачами отдела кадров организации-заказчика являются:

- подбор, расстановка и воспитание кадров;
- изучение деловых и моральных качеств работников по их практической деятельности;
- создание резерва кадров для выдвижения на руководящие и материально ответственные должности;
- организация и проведение всех видов подготовки и повышения квалификации кадров;
- учет кадров;
- обеспечение прав, льгот и гарантий работников организации;

В функции отдела кадров входит:

- разработка кадровой политики и стратегии;
 - разработка прогнозов, определение текущей потребности в кадрах и источников ее удовлетворения на основе изучения рынка труда;
 - комплектование организации кадрами служащих и специалистов требуемых профессий, специальностей и квалификации в соответствии с изменяющимися внешними и внутренними условиями деятельности организации;
 - формирование и ведение банка данных о количественном и качественном составе кадров;
 - подбор и отбор работников совместно с руководителями заинтересованных подразделений и внесение соответствующих предложений об их назначении на указанные должности, оформление приказов о приеме на работу и другой необходимой для этого документации;
 - разработка предложений о приеме на работу по конкурсу в соответствии с порядком, установленным законодательством, подготовка и организация работы конкурсной комиссии;
 - информирование работников внутри организации об имеющихся вакансиях, использование средств массовой информации для помещения объявлений о найме работников;
 - установление прямых связей с учебными заведениями и службами занятости;
 - оформление приема, перевода и увольнения работников в соответствии с трудовым законодательством, положениями, инструкциями и приказами руководства организации;
 - учет личного состава;
 - выдача справок о настоящей и прошлой трудовой деятельности работников;
 - прием, заполнение, хранение и выдача трудовых книжек;
 - ведение установленной документации по кадрам;
 - подготовка материалов для представления персонала к поощрениям
 - подготовка материалов по привлечению работников к материальной и дисциплинарной ответственности;
 - контроль за правильностью расстановки работников и использования их труда в структурных подразделениях предприятия;
 - организация проведения аттестации работников, ее методическое информационное обеспечение, участие в анализе результатов аттестации, осуществление постоянного контроля за ходом выполнения решений аттестационной комиссии;
 - подготовка необходимых материалов для рассмотрения на комиссии по установлению выслуги лет;
 - подготовка соответствующих документов по пенсионному страхованию и представление их в органы социального обеспечения;
 - выдача справок о работе на предприятии, занимаемой должности и размере заработной платы;
 - обеспечение социальных гарантий трудящихся в области занятости, соблюдения порядка трудоустройства и переобучения высвобождающихся работников, предоставления им установленных льгот и компенсаций;
 - составление графиков отпусков, учет использования работниками отпусков, оформление очередных отпусков в соответствии с утвержденными графиками и дополнительных отпусков;
 - оформление и учет командировок;
 - табельный учет;
 - анализ текучести кадров;
 - рассмотрение жалоб и заявлений работников по вопросам приема, перемещения и увольнения, нарушения трудового законодательства;
 - принятие мер к выявлению и устранению причин, порождающих жалобы работников.
- На начальника отдела кадров возлагается персональная ответственность за:
- организацию деятельности отдела по выполнению задач и функций, возложенных на отдел;
 - организацию в отделе оперативной и качественной подготовки и исполнения документов, ведение делопроизводства в соответствии с действующими правилами и инструкциями;
 - соблюдение сотрудниками отдела трудовой и производственной дисциплины;
 - обеспечение сохранности имущества, находящегося в отделе, и соблюдение правил пожарной безопасности;
 - подбор, расстановку и деятельность сотрудников отдела;
 - соответствие действующему законодательству визируемых (подписываемых) им проектов приказов, инструкций, положений, постановлений и других документов.

Анализ существующих программных продуктов

В настоящее время существует множество готовых решений для автоматизации работы отдела кадров: подмножество систем, разработанных на базе 1С:Предприятие, а также прочих программных продуктов. Рассмотрим более подробно некоторые из них.

Программа кадрового обеспечения и делопроизводства «Кадры SB» от компании «Софт СБ» обеспечивает:

- ведение личных карточек сотрудников;
- учет сведений о работе;
- учет состава семьи;
- расчет выслуги лет;
- учет отпусков, отгулов, назначений и перемещений, поощрения, нарушения;
- учет образования, послевузовского образования, повышения квалификации, аттестации, профессиональной переподготовки;
- учет льгот, материальная помощь.

Прочие возможности программы:

- неограниченное число уровней подразделений и отделов (можно использовать одну базу данных для головного предприятия и его филиалов);
- учет вакансий и претендентов на должности;
- учет листков нетрудоспособности (больничных листов). Формирование любых отчетов (реестров) по больничным листам в MS Excel;
- формирование приказов и записок соответствующих форм (MS Word и Excel);
- формирование любых отчетов по персоналу. (MS Word и Excel);
- возможность самостоятельно формировать любые отчеты при помощи SQL-запросов;
- возможность использования в программе КЛАДР (Классификатор адресов России);
- возможность экспорта базы данных в другие форматы БД;
- возможность работы нескольких пользователей одновременно (сетевая версия);
- возможность ведения баз данных нескольких юридических лиц с одного рабочего места (сетевая версия);
- управление правами пользователей для работы в данной программе;
- учет трудовых книг и расчет любых видов стажей;
- штатное расписание. Возможность формирования штатного расписания отдельно по отделам, подразделениям (Excel);
- интеллектуальный мастер устройства на работу, перевода сотрудника на другую должность и увольнения сотрудника;
- внешний интерфейс с индивидуальными настройками;
- будильник, дни рождения, органайзер и многое другое.

Другой пример информационной системы учета кадров на предприятии - программа «Кадры» от компании «БухСофт» - одна из самых простых, по утверждению разработчиков, кадровых программ. Это интуитивно понятная условно бесплатная программа для ведения кадрового учета на предприятии со всеми необходимыми возможностями. Программа «Кадры» - составная часть комплексной программы «БухСофт: Зарплата, Табель, Кадры», но может быть приобретена отдельно от остальных функций.

Как и многие другие кадровые программы, программа "Кадры" от БухСофт позволяет вести учет таких данных как:

- персональные данные - фамилия, имя, отчество, дата и место рождения, пол, адрес регистрации и фактический адрес местожительства, сведения о документе, удостоверяющем личность, предусмотрена возможность вставки фотографии работника;
- налогообложение - заносятся данные, учитываемые при налогообложении доходов работника и формирования отчетов: ИНН, Код ИМНС (формируется автоматически на основании первых цифр ИНН), сведения об инвалидности, предоставление стандартных вычетов, учет доходов с прежнего места работы, сведения об инвалидности, стандартные вычеты, полученные доходы с предыдущего места работы;
- долги - в данной закладке заполняются начисленные, но не выплаченные доходы прошлых лет, долг налогоплательщика по налогу на доходы на начало года;
- начисления - заносятся данные о постоянных начислениях и удержаниях, на основе которых производится автоматическое начисление зарплаты;
- прочее - сведения о семейном положении работника, воинский учет;
- кадровые данные - заносятся сведения, относящиеся к рабочей деятельности сотрудника: табельный номер, статус, место работы, условия труда, характер работы, стаж, сведения о трудовом договоре, категории плательщика страховых взносов и номер страхового свидетельства в ПФР;
- перемещения - программа обеспечивает учет кадровых перемещений (прием на работу, переводы, увольнение). Информация вводится в отдельные поля, а общий наглядный отчет об имеющихся перемещениях выводится в виде таблицы;
- больничные - возможность заведения сведений о периодах болезни сотрудника с последующим автоматическим заполнением табеля рабочего времени за соответствующий период болезни и автоматическим расчетом «больничных» после выхода на работу с помощью функции «Постоянные начисления» в Расчете

зарплаты;

- отпуск - возможность составления графика отпусков;
- командировки - программа обеспечивает учет сведений о командировках работника;
- образование - сведения о виде образования и учебном заведении работника, информация о послевузовском образовании, аттестация, квалификация, переподготовка, знание иностранных языков, профессия работника и т.д.
- сведения о поощрениях (наградах), льготах, на которые имеет право работник, возможность ввода любых дополнительных сведений.

Выбор и обоснование метода решения задачи

Так или иначе, всем организациям приходится сталкиваться с задачей учета кадров. Решить эту проблему можно по-разному.

Вариант первый - использовать для учета средства офисных приложений. Например, вести таблицу в MS Excel или другом табличном редакторе. Плюсы: тривиальное решение, требующее только навыков в использовании табличного редактора. Минусы: необходимость вводить все данные вручную, трудности при обновлении, отслеживании изменений в личных карточках работников.

Несколько лучше обстоят дела, когда вместо табличного редактора используется программа на базе популярной системы управления базами данных, например, 1С. В этом случае значительно облегчен поиск данных, производится проверка дублирования записей, имеются средства построения различных отчетов. Но в целом эта система не избавлена от главного недостатка - необходимости вручную собирать информацию.

Второй вариант - воспользоваться специализированной информационной системой (ИС) для автоматизации учета кадров. Хранение информации в формате базы данных значительно облегчает последующее построение отчетов. Автоматическое обновление информации позволяет повысить оперативность при работе с базой данных.

ИС обычно приобретаются на достаточно долгий срок (среднее время «жизни» ИС - около 10 лет, но это не предел - во многих компаниях используются системы с гораздо большим «стажем» работы, правда, и обрастающими за это время новыми возможностями). Чтобы система автоматизации приносила ожидаемый эффект, она должна соответствовать данному предприятию - его возможностям, уровню развития и т.д. Стоимость ИС для небольшой фирмы не так уж и мала.

Несмотря на то, что в настоящее время существует множество готовых решений в области автоматизации учета кадров, в этих системах реализованы также десятки других задач, которые не всегда найдут свое применение на конкретном предприятии в силу специфики его деятельности.

Критериев выбора систем автоматизации, как и многих других, достаточно сложных и дорогих товаров (например, автомобилей), существует, конечно же, много. Какие-то из них крайне важны, какие-то могут отражать очень индивидуальные потребности. В подобных ситуациях следует во многом ориентироваться на «здравый смысл», а также иметь в виду некоторые ключевые моменты, носящие специальный характер.

Выбирая систему автоматизации, стоит обратить внимание на следующее:

- что система автоматизации может делать, или какова ее функциональность;
- во что обойдется приобретение системы, запуск ее в эксплуатацию и поддержание в рабочем состоянии, т.е. какова ее совокупная стоимость владения (крайне важно знать именно общую стоимость, а не просто цену программного обеспечения).
- есть ли гарантии успешного завершения проекта внедрения и полноценного ввода системы в эксплуатацию;
- что у системы «внутри» и, следовательно, насколько она надежна, долговечна, производительна, в конце концов, современна;
- какова эффективность и возможные сроки окупаемости системы;
- уровень и качество сервиса в послепродажный период;
- возможность сопровождать и развивать систему силами специалистов самой фирмы;
- каковы перспективы системы, будет ли она развиваться и поддерживаться поставщиком в будущем

[1].

Очень важно сначала выявить реальные потребности организации.

Определить реальные потребности организации в автоматизации - дело не простое. Очень хорошо, если в организации разработан план развития на несколько лет вперед, в котором определена роль информационных технологий и описана последовательность создания корпоративной автоматизированной системы управления. Такой продуманный подход дает наибольшую отдачу, существенно снижает риск выбрать «не ту» ИС и избежать проблем так называемой «лоскутной» автоматизации. При этом в качестве первоочередных задач может рассматриваться автоматизация наиболее критичных на данном этапе видов деятельности («узких» мест, от которых существенно зависит жизнь фирмы) или наиболее трудоемких при обработке традиционным способом (среди последних - например, бухгалтерский и налоговый учет, бюджетирование, расчет зарплаты, и

др.) [2].

Таким образом, для реализации задачи автоматизации рабочего места сотрудника отдела кадров организации выбран вариант реализации посредством разработки БД Учет кадров организации.

Технология создания информационных систем предъявляет особые требования к методикам реализации и программным инструментальным средствам. Реализацию проектов по созданию информационных систем принято разбивать на стадии анализа (прежде чем создавать информационных систем, необходимо понять и описать бизнес-логику предметной области), проектирования (необходимо определить модули и архитектуру будущей системы), непосредственного кодирования, тестирования и сопровождения.

Сущность структурного подхода к разработке информационных систем заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. Основные этапы, на которые разбивается процесс проектирования информационной системы, следующие:

- концептуальное проектирование - сбор, анализ и редактирование требований к данным (обследование предметной области, изучение ее информационной структуры, выявление всех фрагментов, каждый из которых характеризуется пользовательским представлением, информационными объектами и связями между ними, процессами над информационными объектами, моделирование и интеграция всех представлений);
- логическое проектирование - преобразование требований к данным в структуры данных. На выходе получаем СУБД-ориентированную структуру базы данных и спецификации прикладных программ [3-6];
- физическое проектирование - определение особенностей хранения данных, методов доступа и т.д. [7-10].

Современные объектно-ориентированные CASE-средства позволяют эффективно решать задачи проектирования приложений. Среди таких пакетов - Rational Rose, Together Control Center, BPWin, ERWin, Model Mart, Silverrun Business Process Modeller, Process Analyst.

Для разработки функциональной модели использовалось CASE-средство Computer Associates BPwin. BPwin является мощным инструментом для создания моделей, позволяющих анализировать, документировать и планировать изменения сложных бизнес-процессов. BPwin предлагает средство для сбора всей необходимой информации о работе предприятия и графического изображения этой информации в виде целостной и непротиворечивой модели. BPwin поддерживает три методологии: IDEF0, DFD и IDEF3, позволяющие анализировать ваш бизнес с трех ключевых точек зрения:

- с точки зрения функциональности системы. В рамках методологии IDEF0 бизнес-процесс представляется в виде набора элементов-работ, которые взаимодействуют между собой, а также показывается информационные, людские и производственные ресурсы, потребляемые каждой работой;
- с точки зрения потоков информации (документооборота) в системе. Диаграммы DFD могут дополнить то, что уже отражено в модели IDEF3, поскольку они описывают потоки данных, позволяя проследить, каким образом происходит обмен информацией между бизнес-функциями внутри системы. В тоже время диаграммы DFD оставляют без внимания взаимодействие между бизнес-функциями;
- с точки зрения последовательности выполняемых работ: еще более точную картину можно получить, дополнив модель диаграммами IDEF3. Этот метод привлекает внимание к очередности выполнения событий.

Функциональное проектирование системы

В результате анализа предметной области была разработана функциональная модель БД. Проектирование проводилось на основе методологий IDEF0 и DFD.

Контекстная диаграмма, приведенная на рисунке 1, реализована с помощью методологии IDEF0.

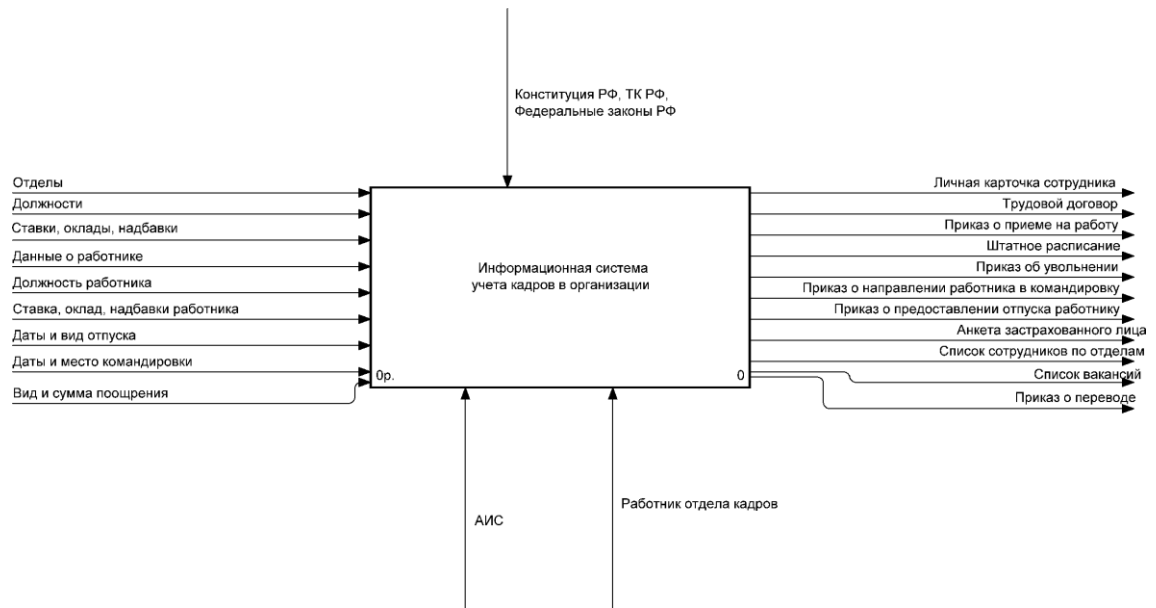


Рисунок 1 – Контекстная диаграмма системы

Входной информацией для системы являются отделы, должности, ставки, оклады и т.д. Ввод входной информации осуществляется работником отдела кадров.

Выходной информацией для системы являются выходные документы.

Функциональная декомпозиция системы, приведенная на рисунке 2, проводится на основе методологии IDEF0.

На этом уровне выполняются следующие функции:

- подготовка штатного расписания;
- определение вакансий;
- прием работника;
- выполнение операций с работником в ходе текущей работы сотрудника отдела кадров.

Функциональная декомпозиция активности «Прием работника», приведенная на рисунке 3, проводится на основе методологии DFD.

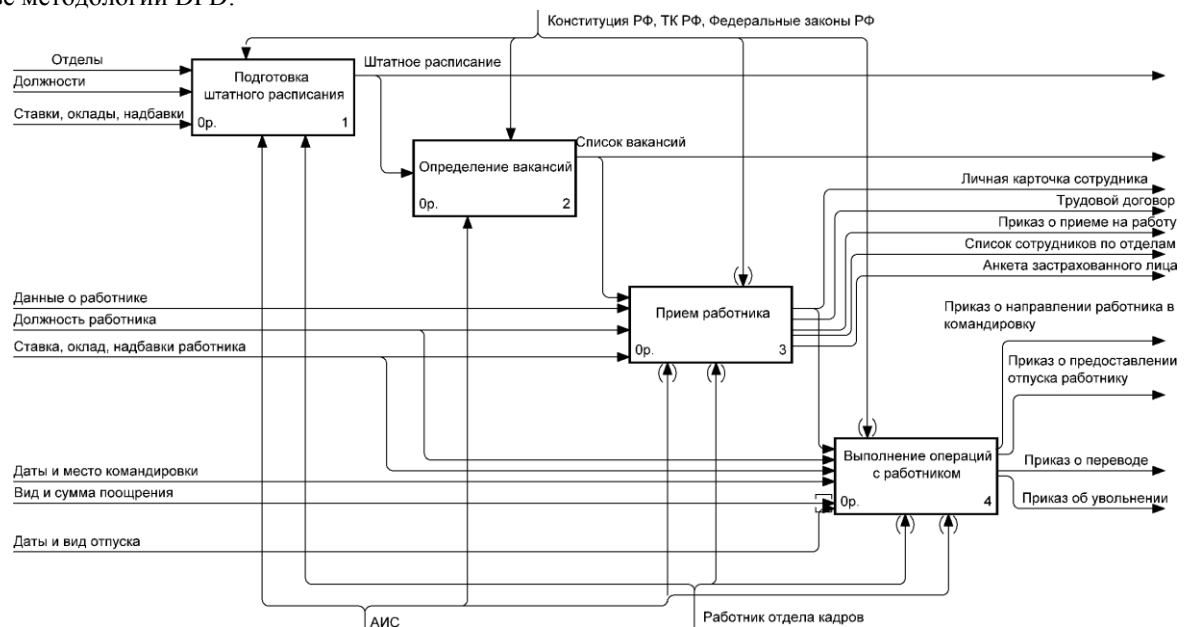


Рисунок 2 – Диаграмма декомпозиции системы

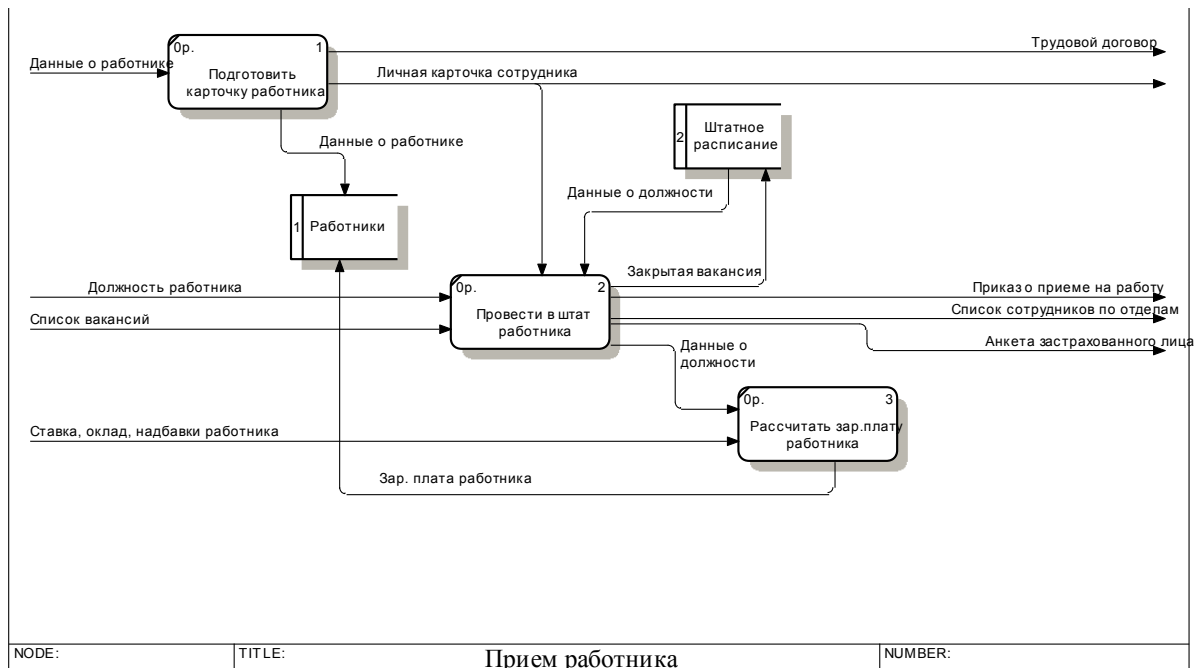


Рисунок 3 – Диаграмма декомпозиции деятельности «Прием работника»

Функциональная декомпозиция активности «Выполнение операций с работником», приведенная на рисунке 4, проводится на основе методологии DFD.

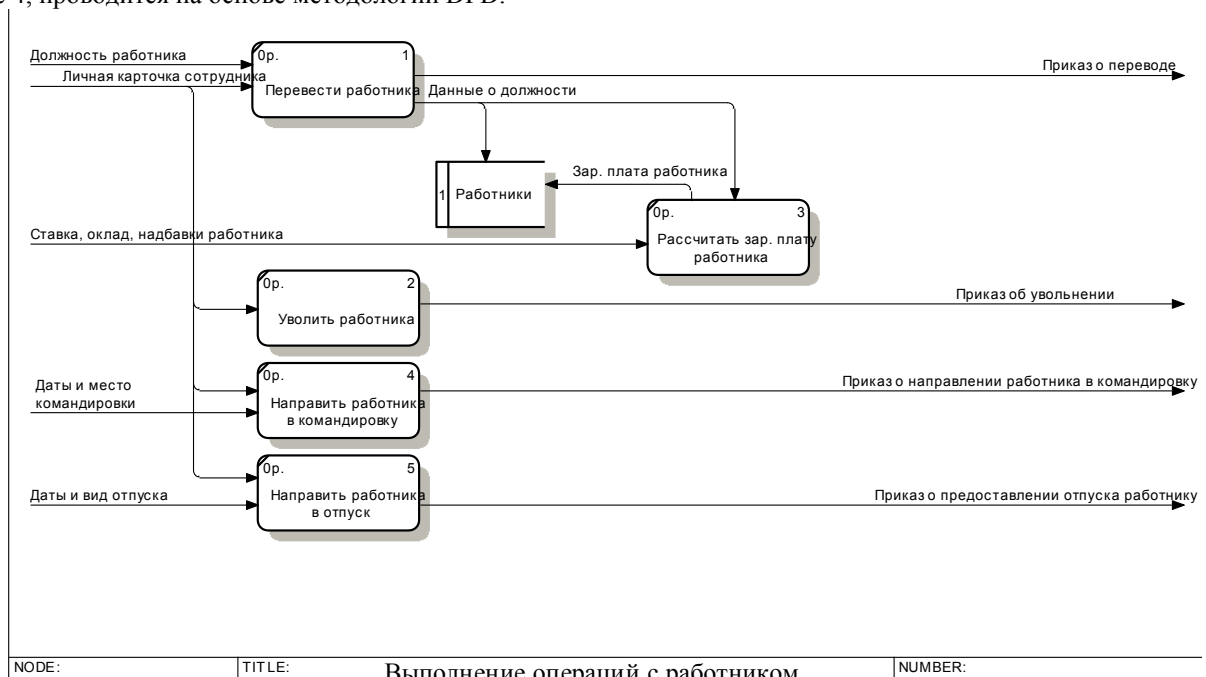


Рисунок 4 – Диаграмма декомпозиции деятельности «Выполнение операций с работником»

Заключение

В результате проведенной работы была разработана база данных учета кадров в организации, автоматизирующая работу работников отдела кадров.

Была проведена следующая работа:

- описана существующая система обработки информации в отделе кадров организации;
- представлены обоснование выбора обеспечивающих технологий и проектных решений по программному и информационному обеспечению;
- продуман пользовательский интерфейс, обозначены возможные пути расширения системы;

– приведена информационная модель и представлено ее описание.

В результате было спроектировано и разработано АРМ работника отдела кадров, реализованы следующие функции:

- подготовка и редактирование штатного расписания;
- учет вакансий;
- ведение личных карточек работников;
- прием сотрудника на работу, перевод работника на новую должность;
- ведение истории работы для каждого работника;
- учет отпусков и командировок работников;
- хранение в архиве базы данных информации об уволенных сотрудниках;
- расчет заработной платы для каждого сотрудника с учетом надбавок и ставки;
- подготовка данных для пенсионного фонда;
- автоматизированная подготовка документов.

Литература

1. Боровская М.А., Катаев Б.В., Костюк А.И., Масыч М.А., Федосова Т.В. Технические решения по оценке коммерческой эффективности инвестиционно-инновационных проектов в интерактивной среде// Инновационные технологии в экономике и управлении (статья). – Таганрог: Изд-во ТТИ ЮФУ, 2010. - № 9(10). -С. 183-196.
2. Боровская М.А., Морозова Т.В., Масыч М.А., Федосова Т.В., Шевченко И.К., Катаев Б.В., Кокорев А.А., Костюк А.И., Курунина Е.Н., Павлов А.Ю. Инструментарно-методическое обеспечение механизма взаимодействия инновационно-ориентированных субъектов на региональном уровне. Монография. / Под ред. Проф. М.А. Боровской. – Ростов-на-Дону - Таганрог: Изд-во ТТИ ЮФУ, 2010. –157 с.
3. А.И. Костюк. «Анализ методов типизации предметных областей баз данных». Материалы Международного конгресса по интеллектуальным системам и информационным технологиям (IS&IT'10). Таганрог, Изд-во ТТИ ЮФУ, 2010г.- С. 58-62.
4. В.Ф. Гузик, А.И. Костюк, Е.В. Ляпунцова, Б.В. Катаев, М.Ю. Поленов. Информационные технологии: Учебное пособие. – Таганрог: Изд-во ТТИ ЮФУ, 2013 – 115 с.
5. В.Ф. Гузик, Е. В. Ляпунцова, А.И. Костюк, Б.В. Катаев. Базы данных: Учебное пособие. – Таганрог: Изд-во ТТИ ЮФУ, 2007. – 128 с.
6. В.Ф. Гузик, А.И. Костюк, Е.В. Ляпунцова. Базы данных. Руководство по выполнению лабораторных работ: Учебное пособие. - Таганрог: Изд-во Технологического института ЮФУ, 2010. – 101 с.
7. А. Д. Хомоненко, В. Э. Гофман, Е. В. Мещеряков. Delphi 7 – 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2010. – 1136 с.
8. Советов Б. Я., Цехановский В. В., Чертовский В. Д. Базы данных: теория и практика: учебник для студ. вузов. - 2-е изд. - М. : Юрайт, 2012. - 463 с.
9. Култыгин О. П. Администрирование баз данных. СУБД MS SQL Server : учеб. пособие. - М. : Московская финансово-промышленная академия, 2012. - 228 с.
10. Баженова И. Ю. Основы проектирования приложений баз данных: учеб. пособие. - М. : БИНОМ. Лаборатория знаний: Интернет-Университет Информационных Технологий, 2006. - 324 с.

Костюк А.И., Лебедева Е.О. База данных учета кадров организации. Предложена база данных учета кадров организации, автоматизирующая работу работников отдела кадров, разработанная на основе контекстной диаграммы, диаграммы декомпозиции, диаграмм логической и физической моделей базы данных Разработанная база данных позволяет повысить эффективность автоматизации документооборота за счет существенного снижения времени, необходимого для обработки информации и оформления кадровых документов.

Ключевые слова: автоматизированное рабочее место, база данных, отдел кадров, программное обеспечение.

Kostyuk A.I., Lebedeva E.O. Database accounting staff of the organization. A personnel record database organization, automating the work of the personnel department employees, developed on the basis of contextual diagrams, decomposition diagrams, logical and physical database model diagrams of data developed database allows to increase the efficiency of workflow due to a significant reduction in the time required for data processing and clearance of personnel documents.

Keywords: workstation, database, human resources software.

Технологии и системы электронного обучения

В.А. Черкасов, В.Н. Азаров
Национальный исследовательский университет Высшая школа экономики
Cherk-vlad@yandex.ru; Vazarov@hse.ru

Черкасов В.А., Азаров В.Н. Технологии и системы электронного обучения. Поставлены задачи по исследованию систем и технологий электронного обучения в России. Рассмотрена технология создания электронных курсов на основе зарубежного опыта. Представлено и рассмотрено ряд аналогов дистанционного и электронного обучения. Сделаны выводы и предложены решения проблемы развития информационно-технологическим систем в России.

Ключевые слова: электронное образование, программное обеспечение, учебный контент, системы управления обучением, средства для разработки электронных курсов.

Введение

В настоящее время наиболее актуальным становится внедрение информационно - телекоммуникационных технологий в учебный процесс, поскольку для специальных категорий обучающихся, которые по объективным причинам не могут посещать образовательные учреждения, данный способ получения знаний представляется наиболее удобным и практичным. Система электронного обучения, её еще называют «*Electronic Learning*» (*e-learning*), в зарубежных странах практикуется достаточно давно и сейчас она является преобладающей формой образования. В России данная форма обучения пока только набирает свою популярность.

Сложность данной технологии состоит в том, что она требует привлечения значительных ресурсов, не только материальных, но и людских. Над созданием и реализацией данных систем трудится не один десяток программистов, например, такой как системы дистанционного тренинга Redclass, а также «Прометей». Сейчас в эпоху информационных технологий, электронное обучение является оптимальным решением для студентов, не только с точки зрения планирования своего времени, но и, с экономической, так как цена за электронные курсы существенно ниже.

Но отталкиваясь от того, что сама технология создания систем электронного обучения не так проста, мы решил исследовать данный вопрос на примере зарубежного опыта, так как на данный момент, система электронного обучения достаточно широко используется в различных зарубежных школах, университетах и других образовательных учреждениях.

Несмотря на то, что огромное количество образовательных учреждений были подключены к единой локальной информационной сети и проведена работа по совершенствованию системы обучения, электронное образование России все же значительно уступает зарубежным информационно-технологическим системам, это и является основной проблемой данной темы. Актуальность темы объясняется тем, что в России отсутствуют единые интегрированные учебные электронные программы, а также электронно-методические пособия, но сейчас прикладываются все усилия для достижения обозначенной цели.

В своей статье мы хотел бы поставить перед собой некоторые задачи:

- сформулировать определение электронного обучения;
- провести различия между электронным и дистанционным обучением;
- вывести плюсы и минусы электронного обучения;
- рассмотреть системы создания и управления учебным контентом;
- обозначить программные решения для создания системы электронного обучения;
- сделать выводы по данной проблематике.

Понятие электронного обучения

Понятие электронного обучения в России закреплено Федеральным законом от 29.12.2012 N 273-ФЗ "Об образовании в Российской Федерации", который гласит: «Под электронным обучением понимается организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих её обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих

передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников» [1].

Отсюда можно сделать вывод, что на данный момент, такая форма обучения признается в России и непосредственно регулируется, а также представляется возможным реализация любой образовательной программы в электронном виде. Первоначально, в конце 20-х годов в России появилось дистанционное обучение, суть которого предполагало взаимодействие обучающихся и педагогических работников даже без применения информационно-телекоммуникационных сетей, например, посредством телерадиовещания, пересылки материалов по почте и другие. В основе дистанционного обучения лежит целенаправленная и контролируемая интенсивная самостоятельная работа обучающегося, который сам устанавливает свой график обучения, т.е. учиться по индивидуальному расписанию, также при себе обучающийся должен иметь комплект специальных средств обучения и согласованную возможность контакта с преподавателем в процессе обучения. Основной плюс данной формы в том, что обучающийся в любое время прохождения обучения может проконсультироваться с преподавателем.

Европейская комиссия определяет *e-Learning* как: «Использование новых технологий мультимедиа и Интернет для повышения качества обучения за счет улучшения доступа к ресурсам и сервисам, а также удаленного обмена знаниями и совместной работы» [2]. *Electronic Learning* предполагает, создание электронной информационно-образовательной среды, с помощью которой, обучающийся самостоятельно осваивает курсы. В процессе электронного обучения используются интерактивные электронные средства доставки информации, преимущественно Интернет и корпоративные сети компаний, но не исключены и другие способы, как, например, CD-диски.

Из вышесказанного, можно определить, что целью электронного обучения является передача образовательных знаний независимым учащимся через платформу онлайн-курсов. Учащийся является независимым, поскольку его никто не ограничивает по времени прохождения курса, а также никто не предоставляет дополнительных консультаций при возникновении вопросов. Курс состоит из интерактивной обучающей информации, такой как мультимедийные презентации, графические технологии, расположенной и поддерживаемой на Web – сервере и, с помощью которого, обеспечивается доступ к содержимому курса.

Среди основных преимуществ электронного обучения можно выделить:

- возможность адаптации учебных курсов для обучающихся с разным уровнем подготовки;
- доступность и практичность использования учебных материалов во времени и в пространстве;
- достоверность проверки заданий с помощью тестов и хранение полученных результатов;
- возможность обновления теоретического материала, фактической и статистической информации в режиме on-line;
- наглядность учебных материалов на основе использования различных средства мультимедиа (анимации, аудио и видео), позволяющая задействовать большинство механизмов восприятия человеком новой информации;
- формирование навыков самоорганизации и рационального планирования учебного времени;
- развитие навыков владения информационно - телекоммуникационными технологиями;
- формирование и развитие навыков целеполагания, инициативности и ответственности.

В 2004 году был проведен опрос компанией SkillSoft [3], в котором участвовали студенты различных вузов. По результатам опроса, были выявлены одни из наиболее важных плюсов электронного обучения: гибкость, экономия времени, простота возвращения к пройденному учебному материалу.

Также нужно добавить, что перечисленные достоинства электронного обучения оценены и в России. *Московский государственный университет экономики статистики и информатики (МЭСИ, вошедший в РЭУ им. Г.В.Плеханова) прибегнул к использованию электронной формы обучения как основной.* МЭСИ входит в состав вузов активно практикующих внедрение *e-Learning* в учебный процесс, которые создали консорциум «Электронный университет». В настоящий момент в сети Интернет разработан уникальный проект, дающий возможность любому желающему усовершенствовать свои знания в области ИТ – *Интернет-Университет Информационных Технологий (ИНТУИТ)*. Лидером среди обучающихся интернет-программ для школьников являются два проекта: «Открытый колледж» и «Виртуальная школа Кирилла и Мефодия». Оба проекта разрабатывают создатели обучающих программ на компакт-дисках. Эти программы в «ознакомительном режиме» также предлагаются и в Интернете.

Основной проблемой данной тематики является то, что создатели технологии *e-learning* не могут гарантировать образование высокого качества. Это связано с тем, что требуется соблюдение определенных условий, которые предполагают значительные траты в данной области, например, административные, затраты на разработку и внедрение учебных материалов и другие.

Также проблема кроется в опыте использовании электронного обучения. Не все образовательные учреждения настроены на применение информационно - телекоммуникационных технологий в учебном процессе, поскольку это требует материальных ресурсов, но и затрагивает квалификацию преподавателей, составляющих программы данных курсов. Такие выводы были сделаны на основе Информационного материала Министерства образования и науки РФ от 19 мая 2014 года в преддверии парламентских слушаний [4].

Также можно сделать вывод о том, что обучаемый не находится под каким-либо присмотром или контролем, в связи с чем невозможно обеспечить достаточную мотивацию, а если курс является достаточно длительным по времени, то это не представляется возможным. Хотя данная точка зрения оспаривается американским исследователем систем обучения Джоном Келлером [5], который предложил модель повышения мотивации к обучению ARCS. Ее главными компонентами являются: внимание, значимость, уверенность и удовлетворение. Д. Келлер писал о том, что сначала нужно привлечь внимание обучаемого, затем нужно убедить его в важности и значимости обучения, после чего необходимо поддержать его уверенность в себе и, в конце концов, добиться удовлетворенности учащегося.

Эффективность модели Д.Келлера подкреплена результатами исследования: через год после внедрения этой технологии число участников, бросивших учебу, снизилось вдвое с 44% до 22%.

Последний недостаток *e-Learning* заключается в отсутствии обратной связи между преподавателем и обучаемым, так как нет вербального общения, поэтому электронное обучение имеет определенные ограничения в применении. Например, оно не подходит для развития навыков работы в команде, уверенности и коммуникабельности.

Система электронного обучения

Система электронного обучения включает в себя:

- программное решение;
- аппаратное решение.

Программные решения — это специально написанные программы-эмуляторы, позволяющие запустить программное обеспечение, разработанное для персональных компьютеров одного типа, на другом ПК. Эмулятор — специальная программа, выполняющая каждую команду исходной программы посредством одной или нескольких команд ПК, на котором происходит эмуляция.

Аппаратные решения представляют собой специальные платы, которые несут на себе дополнительный процессор, оперативную память и видеопамять другой аппаратной платформы. Преимущество заключается в том, что можно легко переключаться между двумя операционными системами, обмениваться между ними файлами и выполнять другие операции, причем производительность обеих систем остается достаточно высокой, и они не влияют друг на друга, так как практически не имеют разделяемых ресурсов, кроме мыши, клавиатуры и монитора.

Система электронного обучения предполагает наличие специальной базы данных, где содержится учебный контент и системы мониторинга обучения.

Три стандартных модуля полномасштабной системы электронного обучения:

- системы управления обучением (LMS - learning management system);
- учебного контента (электронных курсов);
- авторских средств (authoring tools).

Программное обеспечение для E-learning, представлено как простыми статическими HTML страницами, так и сложными системами управления обучением (LMS) и учебным контентом (LCMS), использующимся в корпоративных компьютерных сетях.

Успешное внедрение электронного обучения основывается на правильном выборе программного обеспечения, соответствующего конкретным требованиям, целям и задачам, которые предъявляются к нему организацией [6].

И в связи с этим к средствам организации электронного обучения применяются некоторые требования. Основными из них являются:

- **Функциональность.** Это достаточно важное требование, поскольку от этого зависит прагматичность представленных курсов, которая выявляется в наличии в системе набора функций различного уровня, таких как форумы, чаты, анализа активности обучаемых, управления курсами и обучаемыми и другие.

- **Удобство администрирования,** простота обновления учебного контента и стабильность системы. Это в первую очередь влияет на отношение пользователей к данной системе, а также эффективность ее использования. Стабильность системы определяет степень устойчивости к активной работе пользователей в предлагаемой системе.

- **Стоимость,** которая складывается из цены самой системы, а также из затрат на ее внедрение, разработку курсов и сопровождение, наличие или отсутствие ограничений по количеству лицензий на обучаемых.

- **Удобство использования для обучаемых,** поскольку данная система должна быть интуитивно понятной. Например: простое меню, «легкий» переход от одного раздела к другому.

- **Мультимедийность,** означает наличие аудио, видео, gif- и flash-анимации, 3D-графики различных файловых форматов, а не только текстовых, гипертекстовых и графических файлов.

Сегмент рынка электронного обучения можно разделить на два типа предлагаемых услуг [7]: компании-разработчики программного обеспечения и компании-разработчики электронных курсов и учебного контента.

Первая группа составляет основную долю рынка, которую, в свою очередь, можно разделить на:

- разработку LMS (Learning management system – система управления обучением) и LCMS (Learning Content Management Systems - система управления учебным контентом);
- инструменты для организации и проведения вебинаров или виртуальных классов;
- создание средств для разработки электронных курсов и учебного контента (системы разработки курсов).

Системы управления обучением (LMS - learning management system)

LMS обеспечивает доступ пользователям к содержанию учебных курсов, а также она позволяет разработчику или администратору обучения осуществлять оперативный контроль над самим процессом обучения всех обучающихся. Это представляется возможным за счет разграничения прав доступа к системе. Также LMS можно определить, как платформу для развертывания e-Learning, которая в ряде случаев может использоваться и для администрирования традиционного учебного процесса.

Обучаемые с помощью индивидуального имени пользователя и пароля, которые они получили при прохождении регистрации, имеют доступ к учебному ресурсу, а также к статистике своего обучения. LMS регистрирует: имя пользователя, пройденные им курсы, время начала и конца изучения каждого курса, полученные баллы в результате предварительного тестирования и тестирования после прохождения обучения. Другими словами, пользователь получает от LMS возможность доступа к учебному порталу, который является отправной точкой для доставки всего учебного контента, выбора подходящих учебных курсов на основе предварительного и промежуточных тестирований, использования дополнительных материалов с помощью предназначенных для этого ссылок.

Права администратора позволяют осуществлять контроль над доступом к учебному контенту, организовывать слушателей в группы для предоставления им общих курсов, отслеживать использование курсов, количество набранных баллов, анализировать недостающие навыки и составлять планы дальнейшего развития и обучения сотрудников.

Кроме того, LMS дает возможность администратору создавать отчетность по всему учебному процессу, на основе которой, в дальнейшем, возможно, сделать выводы об эффективности инвестиций в обучение.

Безусловно, для решения вопроса управления учебным процессом, LMS должна иметь возможность проигрывать разные готовые курсы от различных поставщиков. С этой целью в отрасли разработаны специальные стандарты интероперабельности, то есть совокупности технических спецификаций для создания учебного Web-контента. Достаточно известный и часто применяемый в данной области — это стандарт SCORM (Sharable Content Object Reference Model), а также ISM, AICC каждый из которых поддерживается, развивается и имеет всеобщее признание. Таким образом, для полноценной работы все модули системы электронного обучения должны поддерживать один и тот же стандарт.

В совокупности система управления обучением LMS представляет следующие возможности: *поддержки смешанного обучения, интеграции с HR, администрирования с помощью различных инструментов, интеграции контента, тестирования, управления знаниями.*

Традиционными лидерами западного рынка LMS являются решения компаний Saba Software, Docent, WBT Systems, Click2Learn, IBM. Свои предложения есть и на отечественном рынке, среди них — система дистанционного тренинга Redclass (совместная разработка компании Redlab и учебного центра Redcenter) и «Прометей» производства НИЦ АСКБ, e-Learning компании «ГиперМетод», распространяемая в открытых кодах система NauLearning от компании Naumen и другие.

Системы управления учебным контентом (LCMS - Learning Content Management System)

Данные системы имеют своей задачей управление содержанием учебных курсов. Они разработаны для разработчиков контента, специалистов по методологической компоновке курсов и руководителей проектов обучения. Основой является концепция представления содержания обучения как совокупности многократно используемых учебных объектов со своей целевой аудиторией и определенным контекстом использования.

На данный момент существуют различные варианты возможностей LCMS. Исходя из этого, выделяют четыре основных компонента данных систем.

Во-первых, это репозиторий учебных объектов, его определяют, как основную (центральную) базу, которая хранит и управляет учебным контентом. Итоговый продукт может быть доступен через Web, CD-ROM, или в бумажном виде. Каждый объект, может быть использован несколько раз и с различными целями, это зависит от предъявляемых требований.

Во-вторых, необходимо программное обеспечение автоматизированного аутсорсинга, которое используется при создании неоднократно используемых учебных объектов, в последствии, которые потом

будут доступны в репозитории. Эксперты, создатели медиа-продукции [8] а также иные авторы используют шаблоны и архивные образцы, которые содержат основные принципы дизайна, при разработке курсов, при этом они применяют имеющиеся объекты из репозитория, а также создают новые объекты, или используют комбинацию данных объектов.

В-третьих, это интерфейс проигрывания контента. Он создан для того, чтобы отображать материалы, которые запрашивает пользователь, например, для предварительного тестирования. Этот компонент также обеспечивает трекинг результатов, ссылки на источники информации, а также обратную связь для пользователей.

В-четвертых, средства администрирования как приложение, используемое для управления учетными записями, мониторинг результатов, запуск представленных курсов в каталоге, а также другие административные функции. Здесь, можно сказать о том, что данная информация может быть передана в LMS, так как она предназначена для осуществления более продвинутой административной функциональности.

Основные различия между LCMS и LMS состоят в том, что задачей LMS является администрирование, а LCMS – управление учебным контентом. Также, LCMS может управлять контентом на уровне грануляции ниже учебного объекта, что свидетельствует о возможности осуществлять реструктуризацию и перенаправление онлайн-контента. В дополнении, более продвинутые LCMS умеют динамически строить учебные объекты в соответствии с профилями пользователей или стилями обучения. Если обе системы придерживаются стандартов XML, информация может быть просто перемещена в LMS на уровне учебных объектов.

В настоящее время, рынок LCMS пока еще недостаточно фрагментирован, однако он быстро развивается, так как данные системы являются необходимой инфраструктура для электронного обучения.

Вторую группу составляет небольшая часть рынка электронного обучения применительно к России. Это компании, занимающиеся созданием электронных курсов и учебного контента, которые создают учебные курсы, т.е. «наполняют» платформу различными учебными материалами, лекциями, тестами и другое. Здесь можно выделить формы, такие как электронный курс, симуляция, лекция. Может использоваться практически любой электронный контент: графика, анимация, видео, документы и т.д. Разработка учебного контента может быть коллективной и индивидуальной, на основе готовых шаблонов или полностью авторской.

Авторские средства (authoring tools)

Для создания программных продуктов необходимы средства и различные инструменты для разработки учебного контента. В качестве таковых рассматривают и выделяют такие разновидности как:

- редакторы учебных курсов;
- средства для создания презентаций;
- средства для создания тестов, анкет и опросных листов;
- средства для захвата изображения с монитора;
- средства для проведения онлайн уроков.

Путём применения данных средств преподаватели создают электронные пособия, презентации, тесты, видео - уроки и многое другое, что впоследствии посещается в базу данных LMS.

Создание учебного контента – это наиболее сложный процесс на этапе внедрения электронного обучения. В настоящее время в России таких компаний пока немного, чуть более двадцати.

Рынок электронного обучения

По данным исследований Global Industry Analysts [9], представленных в 2010 году с прогнозом на 2015 год, оборот рынка электронного обучения в мире в 2010 году составил 52,6 млрд. долларов, увеличившись по сравнению с 2007 г. на 32% (40 млрд. долларов). К 2015 году аналитическая компания прогнозирует рост до 107 млрд. долларов.

Говоря о доле рынка в мировом секторе, то он принадлежит США и странам Евросоюза. При этом в европейских странах в сфере образования отрасль развивается в основном за счёт государственных дотаций, а в США уже действует коммерческая система.

Мировой рынок электронного образования для самостоятельного обучения (Market for Self-paced eLearning), по данным Ambient Insight, в 2009 году составлял 27,1 млрд. долларов со среднегодовым темпом роста в мире 12%, а в Восточной Европе, включая Россию – 23%. В 2010 году объём мирового рынка для самостоятельного обучения вырос до 32,1 млрд. За последние 5 лет темп роста составил 9,2% и к 2015 объём рынка обещает вырасти до 49,9 млрд. долларов [10].

К сожалению рынок электронного обучения в России не выделен на данный момент в самостоятельную отрасль, статистические данные отсутствуют, и информация об объёмах рынка носит оценочный характер. Например, VP Group и Blackboard считают, что рынок программного обеспечения e-learning в России составляет около 2% от общемирового объёма. Авторы исследования «Электронное образование и развитие инновационной экономики России» считают, что объём рынка e-learning в России в 2010 году составил около

4,7 млн. долл., а в 2011 году прогнозируется увеличение до \$10 млн. в связи с реализацией государственных образовательных программ и растущим спросом. При этом холдинг РБК оценил рынок электронного бизнес-образования в 1,4 млрд. рублей еще в 2008 году.

На сегодняшний день рынок e-learning в России находится в стадии развития и является пока «незрелым». Потенциальный объём рынка оценивается очень высоко, данный вывод можно сделать на основе данных «The Economist Intelligence Unit», граждане России тратят около \$10 млрд. в год на получение дистанционного образования в иностранных университетах. Прогноз российского рынка является положительным и говорит только том, что он будет расти и развиваться как за счет новых потребителей, так и за счет поиска опытными потребителями улучшений существующих систем.

Выводы

Исходя из всего вышесказанного, можно сделать несколько обоснованных выводов. Электронное обучение в России только начинает своё развитие, к тому же создаются аналоги так называемого дистанционного обучения, впоследствии, на основе которых могут создаваться электронные курсы. Это наиболее удобная форма обучения для пользователей, так как её преимущества заключаются в гибкости, практичности, низкой стоимости, функциональности, наглядности и простоте. Но также данная технология имеет и минусы, такие как отсутствие вербального контакта преподавателя и обучающегося, отсутствие мотивации для прохождения курса до самого конца. К сожалению, отсутствие приспособленности у образовательных учреждений и застоявшаяся система дистанционного обучения тоже являются отрицательными факторами.

Для создания системы электронного обучения требуется несколько компонентов для разработки учебного контента, а также его управления, администрирования с целью сделать его практичным и доступным для различных пользователей. А конкретно, системы управления обучением (LMS - learning management system), учебного контента (электронных курсов), авторских средств (authoring tools). Также нужно упомянуть о разделении рынка электронного обучения на два основных сегмента, это компании разработчики программного обеспечения и компании по разработке и составлению электронных курсов. Взаимодействуя между собой, они создают единую систему электронного обучения для пользователей. Но проблема кроется в невозможности создания подобных технологий в нашей стране и конечно же неприспособленности к данной форме обучения, что естественно сказывается на рынке предлагаемых услуг.

На данный момент электронное образование России всё же значительно уступает зарубежным информационно-технологическим системам, данный вывод сделан на основании статистик и исследований, представленных нами в данной статье. Здесь также стоит добавить, что отсутствие экономических ресурса для развития данной области, но и квалифицированных специалистов, разрабатывающих данные курсы существенно малы. Решение данной проблемы состоит в создании интегрированных электронных курсов на основе зарубежного опыта, заимствование и использования подобных технологий, разработка реформы электронного образования в России, а также привлечение инвестиций в данную область.

Литература

1. Российская Федерация. Законы. Об образовании в Российской Федерации [Текст] : федер. закон : [принят Гос. Думой 21 декабря 2012 г. : одоб. Советом Федерации 26 декабря 2012 г.]. [Электронный ресурс].: http://www.consultant.ru/document/cons_doc_LAW_140174/
2. Елена Тихомирова. Живое обучение: Что такое e-learning и как заставить его работать. М: Альпина Паблишер, 2013 - 238 с.
3. Опрос компании SkillSoft. [Электронный ресурс]. – Режим доступа: <http://www.skillsoft.com/>
4. Государственная дума РФ. Комитет по образованию. Информационный материал Министерства образования и науки РФ. М, 2014 – С. 7-8 .
5. Джон Келлер. Model of Motivation and Design. США, 1987. 8 с.
6. Комелина Е.В., Гусакова Т.М. Использование технологий web2.0 в учебном процессе вуза // Преподавание Информационных Технологий в России: Открытая всероссийская конференция [Электронный ресурс]. – Режим доступа: http://www.iteducation.ru/2009/reports/Komelina_Gusakova.htm
7. Лоскутникова В.М. Антропологические аспекты информатизации образования // Открытое и дистанционное образование. Выпуск 1(9), 2003. - С. 66 – 71.
8. Майкл Аллен. E-Learning: Как сделать электронное обучение понятным, качественным и доступным. М: Альпина Паблишер. 2014. - 200 с.
9. Владислав Бюрюлев. Статья: Электронное образование. М, 2015. [Электронный ресурс]. – Режим доступа: <http://ra-kurs.spb.ru/2/0/2/1/?id=83>
10. Елена Тихомирова. Статья: Рынка электронного обучения как такового в России пока нет. М, 2013. [Электронный ресурс]. – Режим доступа: http://www.hr-journal.ru/articles/op/op_1186.html

Черкасов В.А., Азаров В.Н. Технологии и системы электронного обучения. Поставлены задачи по исследованию систем и технологий электронного обучения в России. Рассмотрена технология создания электронных курсов на основе зарубежного опыта. Представлено и рассмотрено ряд аналогов дистанционного и электронного обучения. Сделаны выводы и предложены решения проблемы развития информационно-технологическим систем в России.

Ключевые слова: электронное образование, программное обеспечение, учебный контент, системы управления обучением, средства для разработки электронных курсов.

Cherkasov. V.A., Azarov V.N. Technologies and systems of E-learning. Are set tasks on a research of systems and technologies of electronic training in Russia. The technology of creation of electronic rates on the basis of foreign experience is considered. It is provided and considered a number of analogs of distance and electronic training. Conclusions are drawn and problem resolutions of development information and technological systems in Russia are proposed.

Keywords: electronic education, the software, educational content, learning management systems, means for development of electronic courses.

Программный комплекс диагностики заболевания позвоночника и спинного мозга на основе методов термометрии

С.Я. Семисалов

Донецкий национальный медицинский университет им. М.Горького
serg1948@yandex.ru, mailto:serg1948@yandex.ru

Н.А. Бакланов, А.Г. Гурин, С.В. Тырса

Донецкий национальный технический университет
a.i.001@mail.ru, alexandergurin1996@gmail.com, stas.tyrsa@gmail.com

Семисалов С.Я., Бакланов Н.А., Гурин А.Г., Тырса С.В. Программный комплекс диагностики заболевания позвоночника и спинного мозга на основе методов термометрии. Работа посвящена разработке программно-аппаратного комплекса, использующего методы и средства термометрии. На основе данного комплекса разработана новая методика диагностики заболеваний позвоночника и спинного мозга. Она позволяет оперативно снять показания температуры исследуемого места, проанализировать данные и вывести результаты диагностирования в графической и табличной форме.

***Ключевые слова:** термография, термометрия, микроконтроллер, АЦП (аналого-цифровой преобразователь), COM-port, Atmega8, standalone-приложения*

Введение

На рубеже XX и XXI веков в практику исследований заболеваний человека стали активно внедряться новейшие технические разработки, позволяющие во много раз повысить эффективность диагностики и лечения различных заболеваний.

Позвоночно-спинномозговая травма (ПСМТ) встречается в 0,7 - 4,0 % случаев среди всех травматических повреждений и составляет 6,3% травм костей скелета.^[1]

Одним из новых методов диагностики заболеваний позвоночника и спинного мозга является термометрия.

Следует отметить, что даже такой информативный метод диагностики как компьютерная магниторезонансная томография (МРТ) не позволяет полностью оценить характер и тяжесть изменений, вызванных травмой или заболеванием спинного мозга и позвоночника человека. Это обусловлено тем, что МРТ, являясь дорогостоящим методом обследования, не позволяет прибегать к частым повторным исследованиям и часто не дает возможности определить функциональное состояние изучаемого объекта.

Предложенный метод изучения температуры тела в проекции позвоночника, являясь безвредным, мобильным и достаточно недорогим методом исследования позволяет проводить изучение состояния области позвоночника и спинного мозга в динамике. К тому же метод термографии не требует особых навыков и доступен даже среднему медицинскому персоналу, обладающему знаниями пользователя компьютерной техникой.

Термография относится к методам так называемой «пассивной диагностики» – когда прибор никак не воздействует на объект, а только принимает от него информацию. Таких пассивных методов в медицине очень мало, разве что ЭКГ да ЭЭГ. Такая безопасность снимает все возможные ограничения для выполнения обследования. Термографию можно делать всем и в любом состоянии [2, 9].

Известны несколько технических решений определения температуры в проекции позвоночника и спинного мозга. К таким решениям относятся различные тепловизионные устройства (ТеМП – 79, ТеМП-82, ТеМП-91), а также методы регистрации температуры тела с помощью сложной радиотермометрической техники, позволяющей исследовать не только поверхностную температуру человека, но и изменения температуры в глубинных отделах.

Известен способ изучения поверхностной температуры молочных желез у женщин с помощью различного количества термодатчиков. Применение аналогичной аппаратуры при диагностике заболеваний позвоночника и спинного мозга ранее не было известно.

В данной статье рассматривается разработанный авторами аппаратно-программный комплекс для цифровой термометрии. Предлагаемый комплекс предназначен для построения температурной карты в проекции позвонков и спинного мозга с последующим компьютерным анализом.

Общая структура системы

Блок датчиков (см. рис. 1) представляет собой матрицу термоэлементов, которая контактирует с пациентом, и с помощью которой считываются значения температуры участка тела.

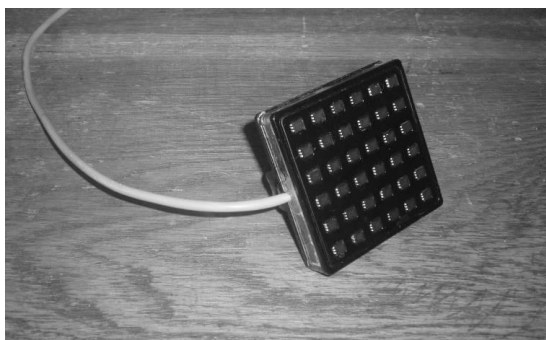


Рисунок 1 - Модуль датчиков

АЦП (аналого-цифровой преобразователь) обеспечивает преобразование данных в цифровой код. Контроллер обеспечивает считывание показаний датчиков и их передачу в компьютер. Также контроллер обеспечивает управление матрицей датчиков по командам от компьютера. Для связи с компьютером используются интерфейсы COM или USB.

Микроконтроллер (англ. MicroControllerUnit, MCU) — микросхема, предназначенная для управления электронными устройствами. Типичный микроконтроллер сочетает в себе функции процессора и периферийных устройств, может содержать ОЗУ и ПЗУ. По сути, это однокристалльный компьютер, способный выполнять простые задачи [8]. В качестве микроконтроллера был выбран Atmega8 (см. рис. 2) [5, 6].



а) внешний вид

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

б) Распиновка выводов

Рисунок 2 – Микроконтроллер Atmega8

Подключение контроллера к компьютеру осуществляется через COM-порт. Для этого был написан драйвер, который и отвечает за саму связь. При написании драйвера мы руководствовались статьей об использовании COM-портов в языке C# [4][7].

Общая структура разработанного аппаратного комплекса изображена на рисунке 3.

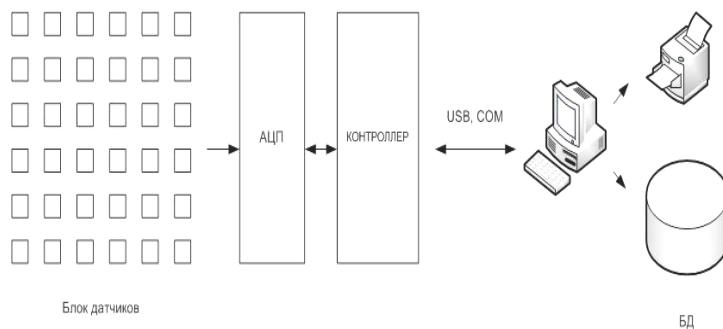


Рисунок 3 – Общая структура системы

Программное обеспечение и его использование

Наша команда пришла к решению разрабатывать программного обеспечения для микроконтроллера на языке C#. В настоящее время язык программирования C# - один из самых мощных, быстроразвивающихся и востребованных языков в ИТ-отрасли, он используется для разработки различного типа приложений: от небольших standalone-приложения утилит до крупных веб-порталов и веб-сервисов, обслуживающих ежедневно миллионы пользователей [3].

Программа предназначена для измерения температуры спины и позвоночника. Кроме считывания температуры, она также может представлять визуализированную карту температуры на экране монитора, осуществлять расчет показателей (среднее значение, отклонение, асимметрия, равномерность и т.д.), выводить отчет на печать и сохранять его на жестком диске или другом электронном носителе.

Возможна работа в режиме телемедицины (удаленного исследования). В этом режиме программное обеспечение для считывания данных устанавливается на переносных мобильных компьютерах, а результаты исследования передаются через сеть Интернет на сервер, где происходит их обработка с последующим занесением в базу данных.

Перед началом использования программы нужно выполнить настройку и калибровку датчиков. Для этого в главном окне программы надо выбрать настройки (см. рис. 4). В настройке можно сразу указать данные о враче, чтобы не вводить их в дальнейшем при каждом обследовании. Тут же указывается директория по умолчанию, в которой будут храниться отчеты исследований. Можно также указать ширину и высоту сетки по умолчанию и временной интервал, в течение которого проводится измерение. Здесь же осуществляется выбор порта, к которому подключено устройство. После успешного нахождения порта можно выполнить калибровку. После настройки программы можно начинать обследования.

Стоит заметить, что настройку достаточно выполнить один раз при подключении устройства, вместе с тем при каждом новом подключении устройства можно выполнить повторную настройку.

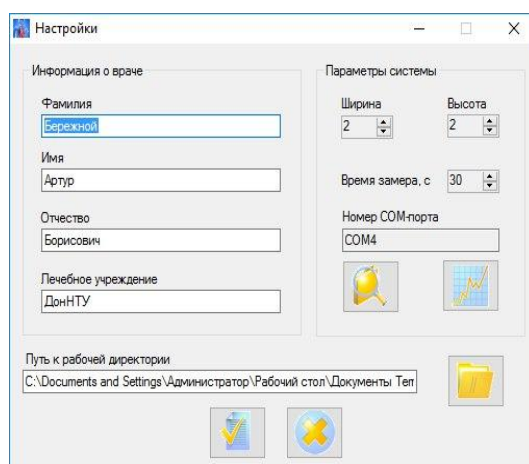


Рисунок 4 – Форма настроек

Во время начала обследования программа запросит данные пациента и врача, который проводит осмотр (см. рис. 5). Кроме этого следует указать данные о текущих температурах окружающей среды, спины, вес и рост пациента. Для врача есть возможность ввода предварительного диагноза. Затем можно приступить к настройке обследования: выбор уровня позвоночника время замера

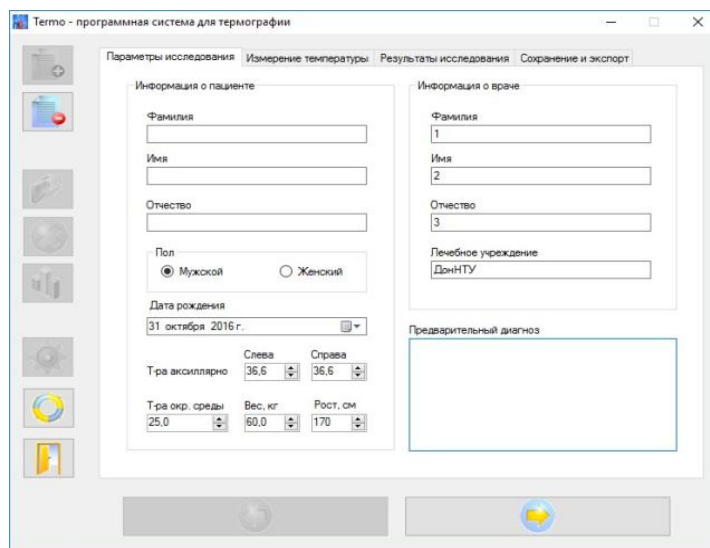


Рисунок 5 – Форма начала обследования

Программа реагирует на нажатие кнопки на модуле датчиков. При нажатии начнется замер температуры, после пройденного времени, введенного ранее, программа выдаст визуализацию полигона температур с возможностью просмотра температуры на отдельном взятом участке полигона. В конце всей процедуры программа создаст отчет о проделанной работе. На рис. 6 представлен пример сформированного программой отчета на основе проведенных измерений температуры.

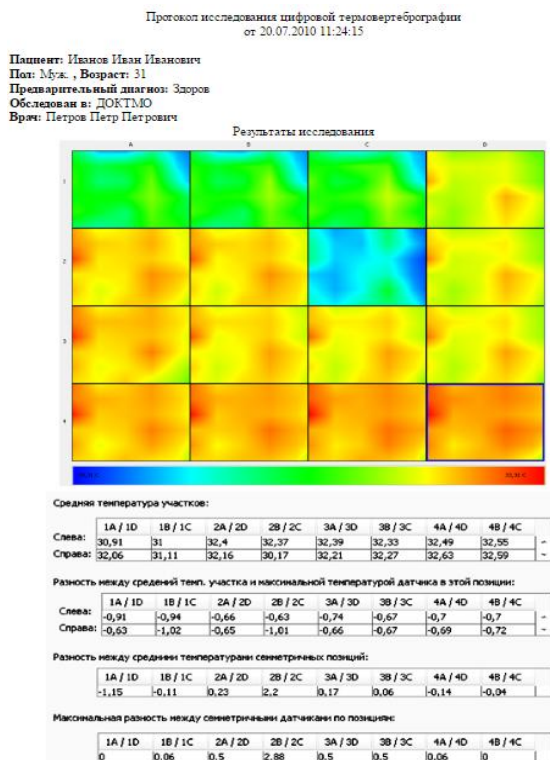


Рисунок 6 - Пример отчета

Технические характеристики разработанного аппаратно-программного комплекса:

Состав системы:

- блок датчиков температуры;
- блок обработки данных;
- контроллер + программное обеспечение

Возможности:

- 36 датчиков температуры (-55...+125°C)

- связь с компьютером по последовательному интерфейсу (COM, USB)
- считывание показаний в ручном и автоматическом режиме
- визуализация карты температур
- расчет показателей асимметрии температур
- сохранение результатов в файл и базу данных
- печать отчета
- кроссплатформенность (работает под управлением любой операционной системы).

Заключение

Был разработан и изготовлен аппаратно-программный комплекс для цифровой термометрии. Внедрение и эксплуатацию данного комплекса может позволить себе каждая клиника, по причине легко доступных компонентов. Разработанный комплекс термометрии вполне пригоден для практической эксплуатации в условиях клиники, причем возможна дальнейшая доработка (при возникновении необходимости) программного обеспечения и математического аппарата, применяемого при анализе данных. Предлагаемый комплекс поможет облегчить и ускорить получение данных о состоянии обследуемого.

Литература

1. Перльмуттер О.А. Травма позвоночника и спинного мозга. Н. Новгород. — 2000. — 144 с.
2. <http://dockozlovsky.ru/meditsinskaya-termografiya/medizinsckay-termografia/termografiya-v-meditsine-polozhitelnyie-storonyi>. Термография в медицине. Положительные стороны. Информационно-справочный портал «Здоровия желаю!». — 2012. Режим доступа: <http://dockozlovsky.ru/meditsinskaya-termografiya/medizinsckay-termografia/termografiya-v-meditsine-polozhitelnyie-storonyi>
3. <http://metanit.com/sharp/tutorial/1.1.php>. Введение в C#. Информационно-справочный портал. — 2015. Режим доступа: <http://metanit.com/sharp/tutorial/1.1.php>
4. <http://microsin.net/programming/pc/visual-studio-sharp-dot-net-com-port.html>. VisualStudioC#: работа с последовательным портом. Информационно-справочный портал. — 2013. Режим доступа: <http://microsin.net/programming/pc/visual-studio-sharp-dot-net-com-port.html>
5. <http://radio-point.narod.ru/datasheet/a1/atmega8.html> Микроконтроллер Atmega8 Datasheet (Описание PDF). Радиоточка. Информационно-справочный портал «Сайт о радиосхемах». — 2011. Режим доступа: <http://radio-point.narod.ru/datasheet/a1/atmega8.html>
6. http://cxema21.ru/load/datashity_dasheet/mikroskhemy/datasheet_na_mikrokontroller_atmega8_rus/25-1-0-33 Datasheet на микроконтроллер ATmega8 (RUS) – Микросхемы-Даташиты (Datasheet). Сайт радиолюбителя. Инф.-справочный портал «Собери свою радиосхему!». — 2015. Режим доступа: http://cxema21.ru/load/datashity_dasheet/mikroskhemy/datasheet_na_mikrokontroller_atmega8_rus/25-1-0-33
7. http://kbss.ru/blog/lang_c_sharp/11.html Пример работы с SerialPort в C# / Язык программирования C#. Информационно-справочный портал «КБ СОФТ СТУДИЯ». — 2013. Режим доступа: http://kbss.ru/blog/lang_c_sharp/11.html
8. <http://www.atmega8.ru/> Описание микроконтроллера, купить ATmega8 и программатор. Информационно-справочный портал «ATmega8». — 2013. URL: <http://www.atmega8.ru/>
9. <http://dockozlovsky.ru/meditsinskaya-termografiya/medizinsckay-termografia/termografiya-v-meditsine-otritsatelnyie-storonyi> Термография в медицине. Отрицательные стороны. Информационно-справочный портал «Здоровия желаю!». — 2012. Режим доступа: <http://dockozlovsky.ru/meditsinskaya-termografiya/medizinsckay-termografia/termografiya-v-meditsine-otritsatelnyie-storonyi>

Семисалов С.Я., Бакланов Н.А., Гурин А.Г., Тырса С.В. Программный комплекс диагностики заболевания позвоночника и спинного мозга на основе методов термометрии. Работа посвящена разработке программно-аппаратного комплекса, использующего методы и средства термометрии. На основе данного комплекса разработана новая методика диагностики заболеваний позвоночника и спинного мозга. Она позволяет оперативно снять показания температуры исследуемого места, проанализировать данные и вывести результаты диагностирования в графической и табличной форме.

Semisalov S., Baklanov N., Gurin A., Tyrsa S. Software complex of spine and spine cord diseases diagnostics using methods and facilities of thermometry. This work is dedicated to hard- and software complex development, using methods and facilities of thermometry. A new method of spine and spine chord diseases based on the given complex was developed. It helps to measure temperature of examined spot efficiently, to analyze data and output diagnostics' results in tabular and graphic forms.

Программная инженерия и обобщенный закон Мура

А.Я. Аноприенко

Донецкий национальный технический университет

anoprien@donntu.org

Аноприенко А.Я. Программная инженерия и обобщенный закон Мура. Классический закон Мура уже более 50-ти лет определяет темпы развития компьютерной индустрии и информационно-компьютерных технологий. Предложенное автором обобщение классического закона Мура позволяет описывать практически любые закономерности процессов экспоненциального развития, характерные в первую очередь для компьютерных наук и технологий. В данном докладе рассматривается применение обобщенного закона Мура для описания выявленных закономерностей развития в области программной инженерии.

Ключевые слова: программная инженерия, закон Мура, экспоненциальный рост, закономерности развития.

Введение

Классический закон Мура — это эмпирическое наблюдение, изначально, в 1965 году, сделанное Гордоном Муром, одним из основателей и руководителей фирмы Интел, и откорректированное в 1975 году, согласно которому (в современной традиционной формулировке) количество транзисторов, размещаемых на кристалле интегральной схемы, с начала 70-х годов удваивается каждые 24 месяца. Наиболее часто упоминаемый интервал удвоения в 18 месяцев связан с наблюдением Давида Хауса из Интел, согласно которому производительность процессоров должна удваиваться каждые 18 месяцев из-за сочетания роста количества транзисторов и быстродействия каждого из них. Первоначально Гордон Мур прогнозировал примерно 10-летний период действия своего эмпирического закона, однако, при завершении каждого из таких прогнозных периодов он продлевал предполагаемый срок действия закономерности еще минимум на десятилетие.

Несмотря на то, что не только сам Гордон Мур, но множество других специалистов все чаще предсказывают завершение действия закона Мура [1], он продолжает действовать феноменально долго, определяя невиданные ранее темпы и размах технического и технологического развития последних десятилетий.

Более детальный анализ темпов развития, как в области компьютерных наук и технологий, так и в других технических областях, позволил выявить множество аналогичных процессов экспоненциального развития [2-8], что привело к необходимости введения такого понятия как обобщенный закон Мура [9]. Данный обобщенный закон позволяет выявлять и единообразно описывать различные процессы экспоненциального развития, характерные, как правило, для этапов наиболее интенсивной эволюции технологий, длительность которых может варьироваться в пределах от одного-двух десятилетий до нескольких тысячелетий [2].

Выявление и использование таких закономерностей представляется чрезвычайно важным как с точки зрения улучшения прогнозирования и планирования развития технических систем и технологий, так и с точки зрения повышения эффективности связанного с их освоением образовательного процесса [10-12].

В данном докладе впервые рассматривается действие обобщенного закона Мура применительно к программной инженерии.

Программная инженерия: необходимость выявления системных закономерностей

Современное типичное определение данной области деятельности выглядит примерно следующим образом: «**Программная инженерия** — это область компьютерной науки и технологии, которая занимается построением программных систем, настолько больших и сложных, что для этого требуется участие слаженных команд разработчиков различных специальностей и квалификаций. Обычно такие системы существуют и применяются долгие годы, развиваясь от версии к версии, претерпевая на своем жизненном пути множество изменений, улучшение существующих функций, добавление новых или удаление устаревших возможностей, адаптацию для работы в новой среде, устранение дефектов и ошибок» [13, с. 10]. При этом характерным является замечание о том, что «**суть методологии программной инженерии** состоит в применении

систематизированного, научного и предсказуемого процесса проектирования, разработки и сопровождения программных средств» [13, с. 10].

Сам термин «программная инженерия», как известно, впервые был предложен в 1968 году на конференции, посвященной кризису программного обеспечения, обусловленному стремительным развитием компьютерных технологий, что в тот период выразилось в массовом переходе на вычислительную технику третьего поколения [14, с. 18]. Такие «кризисы роста» в дальнейшем регулярно повторялись. Однако, в большинстве работ, посвященных программной инженерии (см., например, такие фундаментальные издания как [13-15]), из всех возможных закономерностей упоминается лишь традиционный закон Мура применительно к росту степени интеграции цифровых микросхем и/или росту производительности компьютерных систем. Иногда фрагментарно рассматриваются некоторые количественные характеристики роста сложности программного обеспечения. Например, отмечается, что за десятилетие с 1990 по 2000 год общий объем исходного кода эксплуатируемых программных систем увеличился практически в 2 раза: со 120 млрд. строк до 250 млрд. [14, с. 570]. Другим примером конкретной количественной информации является данные о росте примерно в 2 раза количества проектов в области разработки свободного программного обеспечения за 2 года с начала 2002 г. до начала 2004 г. [15, с. 387]. Фрагментарность и недостаточность такой информации не позволяет сформировать более-менее целостную картину перманентного развития количественных показателей в области компьютерной и программной инженерии, без чего крайне затруднительными являются как текущие оценки эффективности и других параметров программного обеспечения, так и, особенно, перспективные, что, в числе прочего, резко снижает качество и прогнозирования, и планирования, и разного рода аналитических оценок.

Быстрые темпы изменений в области компьютерных наук и технологий требуют знания и учета хотя бы основных системных закономерностей, основой для которых может служить обобщенный закон Мура [2, 9].

Обобщенный закон Мура

В работах [2] и [9] предложены 3 варианта обобщения закона Мура.

Первый вариант предполагает три градации скорости экспоненциального роста, обозначаемые как M1, M2 и M3. При этом цифровой коэффициент соответствует количеству десятичных порядков роста за базовый десятилетний период изменений (т.е., рост в 10 раз за 10 лет для M1 и в 1000 раз за 10 лет для M3). Как показал детальный анализ, очень многие закономерности роста укладываются в один из этих 3-х вариантов.

Второй и третий варианты в качестве базового рассматривают 20-летний период. При этом второй вариант обобщения закона Мура предполагает 6 градаций скорости экспоненциального роста, обозначаемых от L1 до L6. Цифровой коэффициент, как и в предыдущем случае, соответствует количеству десятичных порядков роста за базовый период, равный в данном случае 20-ти годам (т.е., рост в 10 раз за 20 лет для L1 и в миллион раз за 20 лет для L6, при этом L2 эквивалентно M1, а L4 и L6 – соответственно M2 и M3). Практически все известные на сегодня закономерности экспоненциального роста в области компьютерных наук и технологий соответствуют какому-либо из 6-ти вариантов второго обобщения закона Мура. Например, рост с удвоением каждые 18 месяцев соответствует L4, а с удвоением каждые 2 года – L3 (таблица 1).

Таблица 1. Основные закономерности роста для второго обобщения закона Мура, упорядоченные по возрастанию темпов роста

L	Закономерность	КЭР								Интервал удвоения	
		1	2	3	4	5	6	10	20	Лет	Месяцев
1	Рост в 10 раз каждые 20 лет	1,122	1,260	1,414	1,587	1,782	2	3,175	10	6	72
2	Рост в 10 раз каждые 10 лет	1,260	1,587	2,000	2,520	3,175	4	10	102	3	36
3	Закон Мура 1975: удвоение каждые 2 года	1,414	2,000	2,828	4,000	5,657	8	32	1 024	2	24
4	Закон Мура 1985: удвоение каждые 1,5 года (ряд Фибоначчи)	1,587	2,520	4,000	6,350	10,079	16	102	10 321	1,5	18
5	Рост в 10 раз каждые 4 года	1,782	3,175	5,657	10,079	17,959	32	323	104 032	1,167	14
6	Закон Мура 1965: ежегодное удвоение	2	4	8	16	32	64	1024	1 048 576	1	12

В третьем, максимальном, и наиболее гибком варианте обобщения закона Мура предполагается, что основной оценкой скорости технического прогресса во всех случаях будет количество десятичных порядков, на которое возрастает (или в некоторых случаях уменьшается) значение того или иного показателя в ходе развития техники и технологий. При этом соответствующую скорость роста предложено обозначить как S (начальный символ англоязычного слова speed, означающего скорость). Такое обозначение представляется целесообразным также и потому, что большинство процессов развития в технике могут быть описаны различными S-образными кривыми, начальная часть которых (примерно первая половина или несколько более) представляет из себя экспоненциальную или квазиэкспоненциальную кривую).

Для различных скоростей экспоненциального роста обозначение в целом будет иметь вид SK, где K – это численное обозначение скорости роста, равное количеству десятичных порядков роста за базовый период. Это, естественно, отнюдь не предполагает, что соответствующий процесс роста будет длиться на протяжении всего базового периода. Чаще всего рост носит экспоненциальный характер на протяжении всего нескольких десятилетий. Но, как показывает практика, для сравнительного анализа привязка к базовым периодам является достаточно удобной и эффективной.

При рассмотрении технического прогресса можно ограничиться максимальным базовым периодом в 2 тысячи лет, т.к. за пределами данного периода процессы технического развития либо вообще плохо прослеживаются, либо просто недостаточно документированы для более-менее достоверных количественных оценок. В этом случае в обозначении K будет использоваться 3 десятичных знака. Нули справа в обозначении K в большинстве случаев могут отбрасываться.

В общем случае будем считать, что n – количество десятичных знаков (цифр), составляющих K. Тогда K показывает на **сколько десятичных порядков** вырастет соответствующее значение рассматриваемого параметра роста за $2 \cdot 10^n$ лет.

Символом S без цифровых коэффициентов будем обозначать переменную, принимающую значение K. Для того, чтобы значение S было корректным для различных n для определения его целесообразно использовать следующую формулу:

$$S = 10 \cdot 0, K. \quad (1)$$

Это, например, означает что для S1 $S=1$, для S05 $S=0,5$, для S225 $S=2,25$.

Минимальной единицей времени при расчетах будем считать один год, так в пределах года даже в случае самых быстрых темпов технического прогресса существенных изменений либо не наблюдается, либо они лежат в пределах возможных отклонений. Тогда для вычисления экспоненциально растущих значений P_i для любого i-го года ($i > 0$) при известном начальном значении P_0 может использоваться следующая зависимость:

$$P_i = P_0 \cdot 2^{S \cdot (Y_i - Y_0)/6}, \quad (2)$$

где

Y_0 – начальный год действия соответствующей закономерности;

Y_i – текущий год действия соответствующей закономерности;

Использование такой формы зависимости, описывающей закономерности роста, позволяет обеспечить совместимость системы обозначений для всех вариантов обобщенного закона Мура. При этом, например, для шести закономерностей второго варианта обобщенного закона Мура от L1 до L6 будем использовать, соответственно, обозначения от S1 до S6.

В таблице 2 приведены ежегодные коэффициенты роста sK для различных значений K, выраженных как X, 0X и 00X для 3-х вариантов базовых периодов.

В общем случае при произвольных значениях K таблица 1 может использоваться для определения значения K в соответствии со следующим алгоритмом:

Шаг 1: Определяем, за какой в период Y (в годах) происходит рост в Z раз.

Шаг 2: Извлекаем корень степени Y из Z.

Шаг 3: В таблице 1 в колонке «20 лет» находим ближайшее значение, не превышающее полученное на шаге 2. Соответствующий номер строки X будет первым десятичным знаком значения K. В случае полного совпадения значений на данном шаге алгоритм завершается.

Шаг 4: Определяем, во сколько раз значение, полученное на шаге 2 отличается от значения, определенного на шаге 3.

Шаг 5: В таблице 1 в колонке «200 лет» находим ближайшее значение, не превышающее полученное на шаге 4. Соответствующий номер строки X будет вторым десятичным знаком значения K. В случае полного совпадения значений на данном шаге алгоритм завершается.

Шаг 6: Определяем, во сколько раз значение, полученное на шаге 4 отличается от значения, определенного на шаге 5.

Шаг 7: В таблице 1 в колонке «2000 лет» находим значение, ближайшее к полученному на шаге 6. Соответствующий номер строки X будет третьим десятичным знаком значения K. На этом алгоритм завершается.

Естественно, что альтернативой значениям S могут быть просто коэффициенты ежегодного роста, но они не дают достаточно наглядного представления о реальных многолетних темпах технического прогресса.

Таблица 2. Коэффициенты роста для третьего обобщения закона Мура

X	Коэффициенты роста за базовый период (20, 200, 2000 лет)	Ежегодные коэффициенты роста		
		20 лет: sX	200 лет: s0X	2000 лет: s00X
1	10	1,12	1,012	1,0012
2	100	1,26	1,023	1,0023
3	1 000	1,41	1,035	1,0035
4	10 000	1,58	1,047	1,0046
5	100 000	1,78	1,059	1,0058
6	1 000 000	2,00	1,072	1,0069
7	10 000 000	2,24	1,084	1,0081
8	100 000 000	2,51	1,096	1,0093
9	1 000 000 000	2,82	1,109	1,0104
10	10 000 000 000	3,16	1,122	1,0116

Обобщенный закон Мура в программной инженерии: характерные примеры

Детальный анализ позволил выявить целый ряд действующих закономерностей, определяющих развитие программной инженерии, среди которых в первую очередь необходимо выделить устойчивый на протяжении уже более чем полувека экспоненциальный рост количества программируемых устройств (рис. 1).

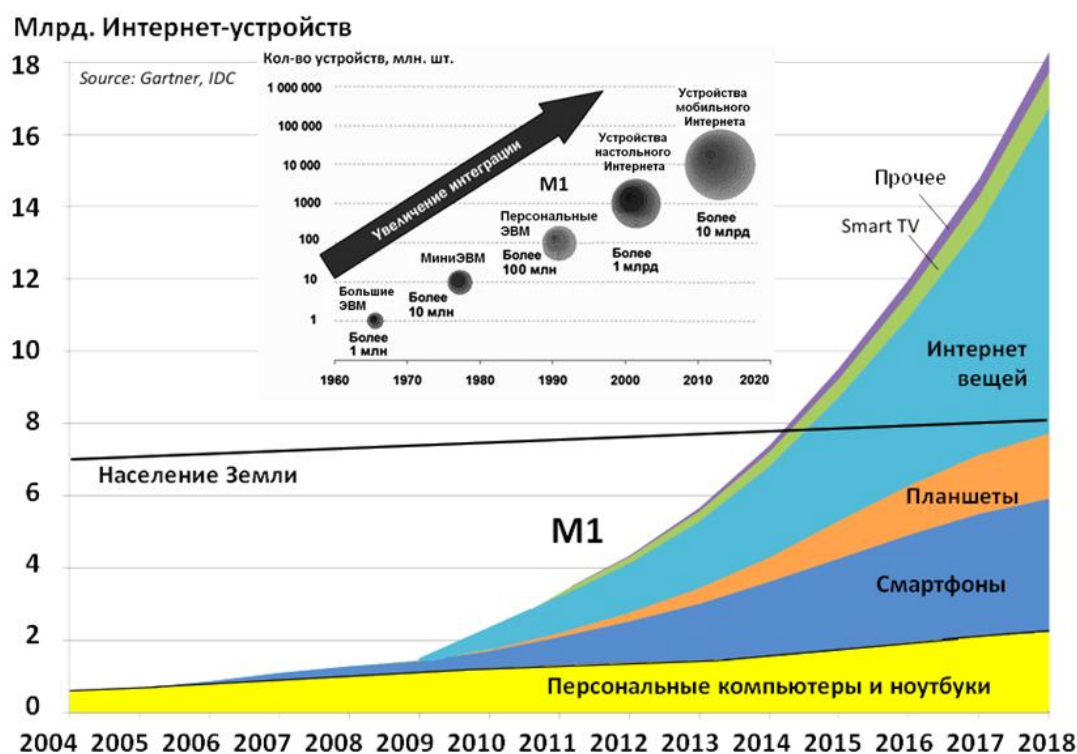


Рисунок 1 – Рост количества программируемых устройств в соответствии с закономерностью M1 (а также – L2 и S2) с 1960 года (врезка вверху) до 2018 и далее – с 2010 года закономерность поддерживается ростом в рамках развития «Интернета вещей»

В связи с тем, что программная инженерия будущего будет ориентирована преимущественно на глобальную сетевую инфраструктуру, то особый интерес представляют закономерности роста количества доступных в Интернет прикладных программных интерфейсов (рис. 2).

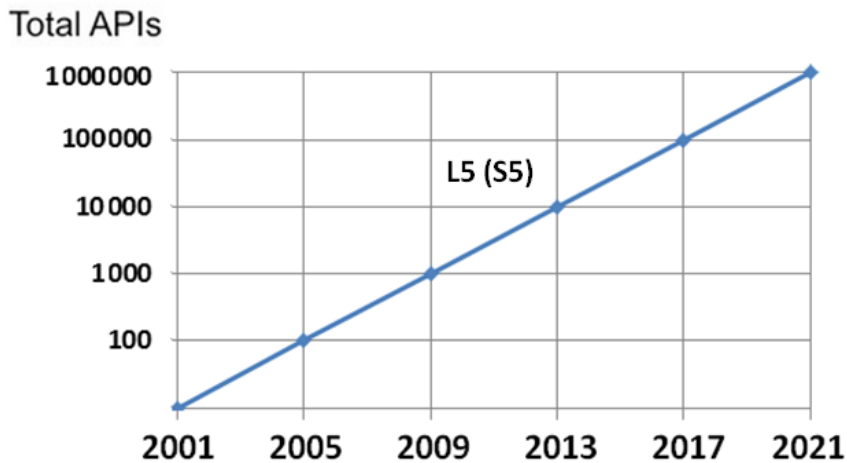


Рисунок 2 - Рост количества доступных в Интернет прикладных программных интерфейсов (API) в соответствии с закономерностью L5 (S5): рост на 5 десятичных порядков за 20 лет, или в тысячу раз каждые 12 лет или в 1,77828 раз ежегодно.

Постоянное усложнение программного обеспечения приводит, в частности, к перманентному росту объемов дистрибутивов операционных систем (ОС). В частности для ОС фирмы Microsoft эта закономерность соответствует L2 (рис. 3).

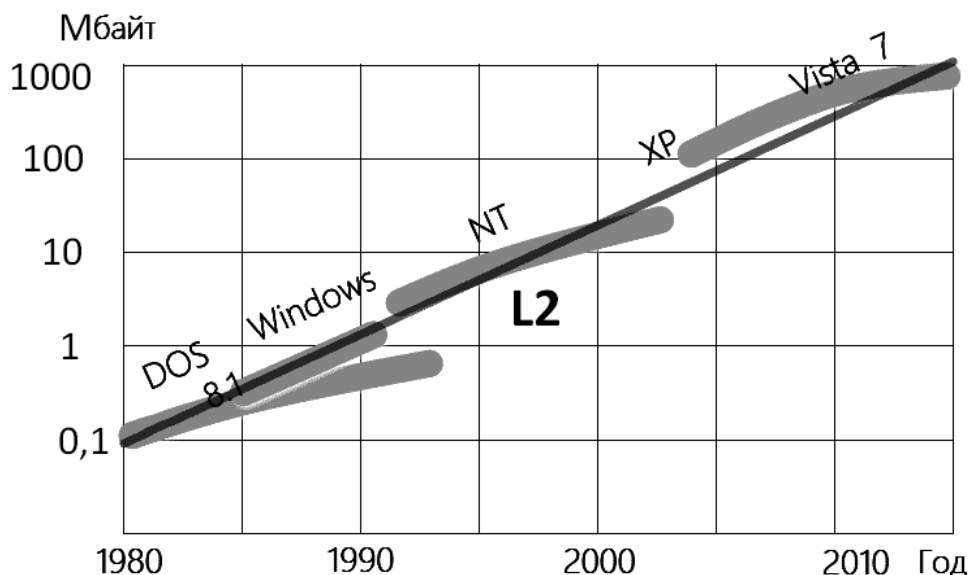


Рисунок 3 – Экспоненциальный рост размеров дистрибутивов операционных систем фирмы Microsoft в соответствии с закономерностью L2 (S2)

Поддерживать надежность программного обеспечения на требуемом уровне при постоянном росте его объемов позволяет устойчивый рост качества программных систем, выражаемый, например, в количестве ошибок на 1000 строк кода, которое с 60-х годов прошлого века устойчиво снижается в том же темпе L2, который полностью соответствует темпам роста объемов эксплуатируемого программного обеспечения (рис. 4).

Точно также темпы роста требований к минимальным объемам оперативной памяти по мере развития операционных систем закономерно соответствуют темпам роста степени интеграции микросхем памяти – в обоих случаях наблюдается закономерность L3 (рис. 5), соответствующая классическому закону Мура с удвоением каждые 2 года.

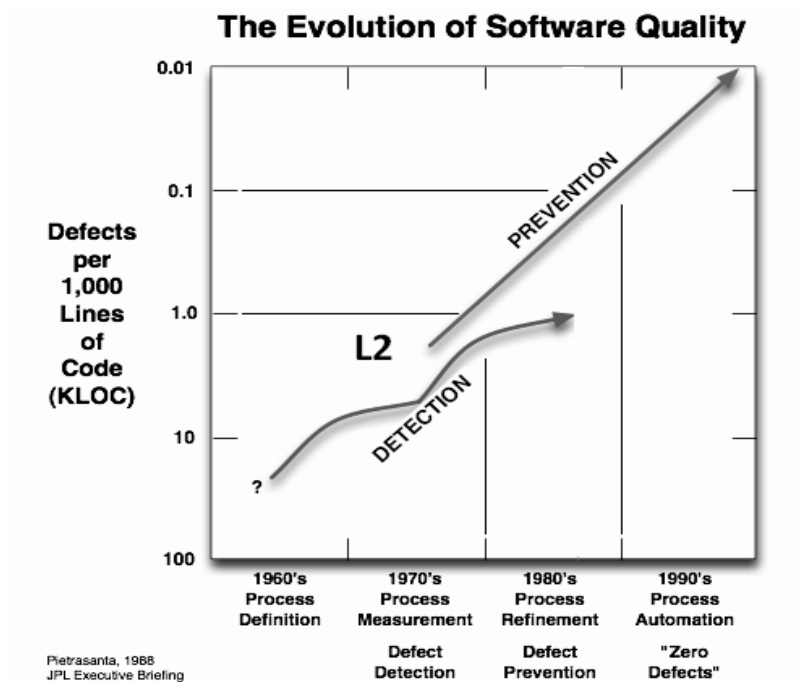


Рисунок 4 – Пропорционально экспоненциальному росту объемов программного обеспечения повышается его качество в соответствии с той же закономерностью L2 или S2 (Defects per 1000 Lines of Code – количество дефектов на 1000 строк кода), что в целом позволяет поддерживать надежность программного обеспечения на требуемом уровне

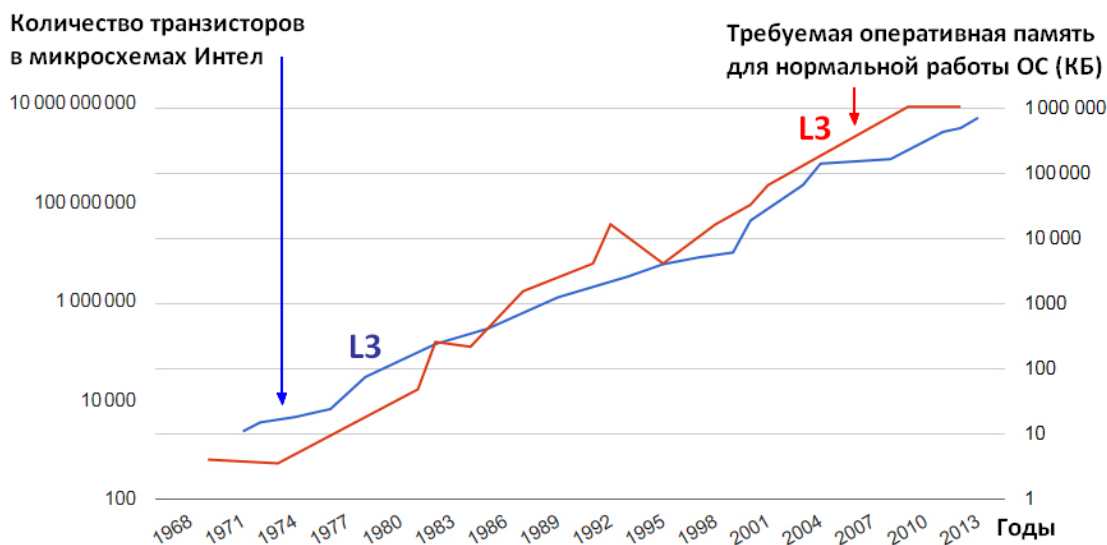


Рисунок 5 – Рост требований к минимальным объемам оперативной памяти по мере развития операционных систем закономерно соответствует темпам роста степени интеграции микросхем памяти – в обоих случаях наблюдается закономерность L3

Заключение

На основании данных, представленных в источниках [16-20] и прочих, выявлены также и другие закономерности, характеризующие количественный рост в самых различных областях программной инженерии, что подтверждает продуктивность предложенного подхода и возможность сформировать в дальнейшем целостную систему закономерностей, определяющих развитие программной инженерии.

Литература

1. Мета-закон Мура: количество новостей про конец закона Мура удваивается каждые 18 месяцев // «Типичный программист». 22 июня 2016. [Электронный ресурс] URL: <https://tproger.ru/news/meta-moores-law>.
2. Аноприенко А.Я. Системодинамика техносферы: как измерить технический прогресс // Системный анализ и информационные технологии в науках о природе и обществе, 2015. № 1(8)-2(9). С. 47-58.
3. Аноприенко А.Я. Периодическая система развития компьютерных систем и перспективы нанокompьютеризации // Инновационные перспективы Донбасса: Материалы международной научно-практической конференции. Донецк, 20-22 мая 2015 г. Том 5. Компьютерные науки и технологии. – Донецк: Донецкий национальный технический университет, 2015. С. 5-13.
4. Аноприенко А.Я. Основные закономерности эволюции компьютерных систем и сетей // Научные труды Донецкого национального технического университета. Серия «Проблемы моделирования и автоматизации проектирования» (МАП-2013). Выпуск № 1 (12) – 2 (13): Донецк: ДонНТУ, — 2013. С. 10–32.
5. Аноприенко А.Я. Закономерности развития компьютерных систем // «Научная дискуссия: инновации в современном мире». №10 (18): Сборник статей по материалам XVIII международной заочной научно-практической конференции. – М.: Изд. «Международный центр науки и образования», 2013. – С. 19-29.
6. Аноприенко А.Я. Модели эволюции компьютерных систем и средств компьютерного моделирования // Материалы пятой международной научно-технической конференции «Моделирование и компьютерная графика» 24-27 сентября 2013 года, Донецк, ДонНТУ, 2013. С. 403-423.
7. Аноприенко А.Я. Будущее компьютерных технологий в контексте технической и кодо-логической эволюции // Вестник Инженерной Академии Украины. Теоретический и научно-практический журнал Инженерной Академии Украины. Выпуск 3-4, 2011. С. 108-113.
8. Аноприенко А.Я. Компьютерные науки и технологии: следующие 50 лет // Материалы II всеукраинской научно-технической конференции «Информационные управляющие системы и компьютерный мониторинг (ИУС и КМ 2011)» – 12-13 апреля 2011 г., Донецк, ДонНТУ, 2011. Т.1. С. 7-22.
9. Аноприенко А.Я. Закономерности развития компьютерных технологий и обобщенный закон Мура // Вестник Донецкого национального технического университета, №2 (2), 2016. С. 3-17.
10. Аноприенко А.Я. Основные направления совершенствования профессиональной подготовки в области компьютерных наук и технологий // «Информатика и кибернетика» (Вестник Донецкого национального технического университета), №1 (1), 2015. С. 5-15.
11. Аноприенко А.Я. Современный компьютинг и программирование // Материалы IV всеукраинской научно-технической конференции «Информационные управляющие системы и компьютерный мониторинг (ИУС и КМ 2013)» – 24-25 апреля 2013 г., Донецк, ДонНТУ, 2013. В 2-х томах. Т. 1. С. 11-17.
12. Аноприенко А.Я. Вызовы и перспективы развития высшего образования в области компьютерных наук и технологий // Материалы III всеукраинской научно-технической конференции «Информационные управляющие системы и компьютерный мониторинг (ИУС и КМ 2012)» – 17-18 апреля 2012 г., Донецк, ДонНТУ, 2012. С. 12-15.
13. Липаев В.В. Программная инженерия. Методологические основы. – М.: ТЕИС, 2006. – 608 с.
14. Соммервилл И. Инженерия программного обеспечения. – М: Вильямс, 2002. – 624 с.
15. Одинцов И.О. Профессиональное программирование. Системный подход. – СПб: БХВ-Петербург, 2004. – 610 с.
16. Бородин В.А. Интернет вещей - следующий этап цифровой революции // Образовательные ресурсы и технологии, 2014, № 2(5). С. 178-182.
17. Larus P. Spending Moore's Dividend // Communications of the ACM, May 2009, Vol. 52, No. 5. – P. 62-69.
18. Strzałka D. Fractal Properties of Linux Kernel Maps // Computer Science and Engineering, 2012, 2(6). - P 112-117.
19. Fitzgerald B. How many lines of code are there? Less than you think // Software Development, 2013/09/15.
20. Deshpande A., Riehle D. The Total Growth of Open Source // Proceedings of the Fourth Conference on Open Source Systems (OSS 2008). Springer Verlag, 2008. - 13 p.

Anopriyenko A.Y. Software engineering and generalized Moore's Law. Classic Moore's law for more than 50 years determines the development of the computer industry and information technology. The proposed generalization of the classical Moore's Law allows to describe any pattern of exponential development processes that are typical primarily for Computer Science and Technology. This report examines the use of the generalized Moore's Law to describe the identified patterns of development in the field of software engineering.

Keywords: Software engineering, Moore's Law, exponential growth, development patterns.

Информационная система для анализа результатов лечения пациентов с воспалительными заболеваниями толстой кишки

А.В. Борота, А.А. Борота, А.П. Кухто, Н.К. Базиян-Кухто
Донецкий национальный медицинский университет
naiga-251088@mail.ru

О.И. Федяев, В.С. Бакаленко
Донецкий национальный технический университет
fedyaev@donntu.org, valeriy.bakalenko@gmail.com

Борота А.В., Борота А.А., Кухто А.П., Базиян-Кухто Н.К., Федяев О.И., Бакаленко В.С. Информационная система для анализа результатов лечения пациентов с воспалительными заболеваниями толстой кишки. В статье рассматривается разработка медицинской информационно-справочной системы, автоматизирующей процессы сбора, хранения, редактирования и предварительной статистической обработки данных для предоставления медицинским работникам необходимой информации о пред- и постоперационных периодах лечения заболеваний толстой кишки и органов брюшной полости. Для решения данной задачи использовались современные методы и технологии построения информационных систем с базой данных реляционного типа. В результате построена объектно-ориентированная система открытого типа, которая в дальнейшем будет функционально расширяться до системы поддержки принятия врачебных решений с целью эффективного лечения заболеваний кишечного тракта.

Ключевые слова: база данных, информационная система, UML-модели, медицина, органы брюшной полости, заболевание толстой кишки.

Введение

Важной научной программой, направленной на информационное обеспечение здравоохранения, является разработка информационных ресурсов и сервисов, обеспечивающих поддержку принятия решений в сфере здравоохранения [1]. При создании таких систем в первую очередь должны быть реализованы следующие основные процессы: ввод, хранение, обработка и вывод медицинской информации в удобном для пользователя виде. Автоматизация этих процессов, учитывая первичный характер обработки данных, успешно реализуются информационно-справочными системами, которые относятся к первому (нижнему) уровню в архитектурной иерархии информационных систем. В данной статье рассматривается разработка медицинской информационно-справочной системы как базы, которая в дальнейшем будет функционально расширяться до системы поддержки принятия врачебных решений в направлении лечения заболеваний кишечного тракта, как традиционными методами, так и симулятивными операционными вмешательствами (СОВ) [2,3].

За период с 2012 по 2016 год в клинике общей хирургии №1 Донецкого национального медицинского университета им. М. Горького на базе Донецкого колопроктологического центра пролечено 189 пациентов с диагнозом язвенного колита (ЯК), из которых 68 (35,9%) были оперированы в один или несколько этапов с последующим реконструктивно-восстановительным вмешательством. Проблема одновременной хирургической коррекции нескольких заболеваний интересует хирургов в течение нескольких десятилетий. Еще в 1985 году Всемирной Организацией Здравоохранения опубликованы статистические данные, согласно которым у 25%-30% больных, подлежащих оперативному лечению в связи с каким-либо заболеванием, выявляются дополнительно одно или несколько заболеваний, требующих одномоментного хирургического лечения. Однако, несмотря на наличие реальных возможностей для оказания полноценной медицинской помощи больным данной категории и достижения максимального медико-социального эффекта, до настоящего времени симулятивные оперативные вмешательства производятся лишь у 1,5%-6% этих пациентов [2,3]. Крайне незначительное количество выполняемых сочетанных операций, не соответствующее реальным потребностям в них, объясняется различными авторами целым рядом причин: неполное обследование больных в предоперационном периоде, недостаточная интраоперационная ревизия органов брюшной полости (ОБП), преувеличение степени операционного риска при определении возможности [3,4].

Предметом дискуссии в настоящее время продолжают оставаться вопросы, связанные с

классификацией СОВ, определением показаний и противопоказаний к их выполнению, выбором оперативного доступа, объемом выполнения при поражении смежных органов, последовательностью выполнения основного и симультанных этапов, оценкой и особенностями постагрессивных реакций в послеоперационном периоде, ведением послеоперационного периода, а также социально-экономической эффективностью сочетанных оперативных вмешательств.

Разработка оптимальных алгоритмов диагностики и эффективных методов лечения пациентов с заболеванием толстой кишки (ЗТК) и сочетанной патологией органов брюшной полости (ОБП) является актуальной проблемой колопроктологии. Все это определяет перспективность исследований, направленных на усовершенствование диагностики и лечения данной категории пациентов. Отсутствие единой информационной базы данных о результатах проведенных операций и автоматизированных средств интеллектуального анализа данных затрудняют объективную оценку СОВ и препятствуют их более широкому применению в клинической практике [5-7].

Поэтому целью исследования состоит в разработке и оценке эффективности информационной системы, автоматизирующей процессы сбора, хранения, отсеивания и предварительной статистической обработки данных для предоставления хирургам необходимой информации о пред- и послеоперационных периодах выполнения СОВ у пациентов с различными заболеваниями толстой кишки и хирургической патологией органов брюшной полости.

Предметная область исследования

Задача исследования состоит в анализе результатов лечения объекта исследования, в роли которого выступают сочетанные заболевания толстой кишки (ЗТК) и органов брюшной полости (ОБП). Осложнения со стороны желудочно-кишечного тракта (ЖКТ) создают угрозу для жизни пациента. Поэтому эти осложнения являются абсолютным показанием к хирургическому лечению болезни ЖКТ. Но проблема состоит в том, что на данном этапе развития медицины нет методики, которая достоверно снижала бы риск рецидива такого типа заболевания. Рассмотрим в качестве примера болезнь Крона (БК) [9]. Хирургическое лечение БК - проблема, которая волновала хирургов всего мира долгое время. Причин этому несколько: увеличение количества случаев заболеваний, отсутствие единого мнения в вопросах хирургической тактики, качество жизни больных после операции. В хирургическом лечении БК сейчас применяются реконструктивно - восстановительные операции. В этом случае очень важно доказать хирургу, используя имеющиеся данные о результатах выполнения пластических операций при БК, целесообразность выполнения реконструктивно - восстановительных операций у данной категории больных. Эта задача может эффективно решаться на основе анализа больших объемов медицинских данных путём использования соответствующих информационных систем.

Предметом исследования являются методы и технологии построения информационных систем с базой данных реляционного типа [10]. Информационная система для анализа лечения толстой кишки (ИС АЛТК) использует реляционную модель данных. Предметная область (медицинские данные о результатах лечения толстой кишки) представлена в виде реляционной модели данных, которая описывается отношением R со схемой r , являющимся подмножеством кортежей декартового произведения $D_1 \times D_2 \times \dots \times D_n$

$$\{d_1, d_2, \dots, d_n, d_1 \in D_1, d_2 \in D_2, \dots, n \in D_n, \}$$

Домены D_i определяют множество возможных значений атрибута A_i реляционной таблицы. Схема r отношения R определяется списком имён атрибутов, т.е. названиями колонок таблицы

$$r = \{A_1(D_1), A_2(D_2), \dots, A_n(D_n)\},$$

где $A_1(D_1), A_2(D_2), \dots, A_n(D_n)$ - имена атрибутов (ФИО пациента, пол, дата рождения и т. д.). В ИС АЛТК насчитывается более 70 атрибутов (арность $n > 70$), которые разбиты на четыре группы:

- информация о пациенте;
- информация об операциях пациента;
- информация о гистологических данных пациента;
- информация о результатах различных анализов.

Условия отображения рассматриваемой предметной области в кортежи значений атрибутов $\langle d_1, d_2, \dots, d_n \rangle$ описываются предикатом $P(d_1, d_2, \dots, d_n)$. Конкретные значения кортежей предоставила кафедра общей хирургии №1 ГОУ ВПО «Донецкий национальный медицинский университет им. М.Горького». Таким образом, любой операционный запрос к базе данных результатов лечения толстой кишки (БД РЛТК) может быть сформирован в виде последовательности, составленной из 8 основных операций реляционной алгебры: объединение, разность, декартово произведение, проекция, селекция и др.

Логические модели информационной системы

На этапе проектирования ИС АЛТК использовался объектно-ориентированный подход. Модели системы описывались на визуальном языке моделирования UML [10, 11]. Концептуальное описание ИС АЛТК представлено диаграммой вариантов использования, которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования (см. рис.1).

Медицинские работники, как пользователи системы, представлены одним актёром «Медицинский работник». Система предоставляет актёру две главные функции: создать пациента и просмотреть пациента, вызов которых возможен благодаря двум ассоциативным связям.

Режим ввода информации о пациенте декомпозирован на четыре подфункции, которые подключаются связями <include> к базовому прецеденту «Создать клиента». Каждая подфункция предусматривает ввод одной из групп показателей пациента: общая информация, информация о проведенных операциях, информация об обследованиях и информация о гистологических ответах.

Актёр (пользователь) в режиме просмотра данных о пациенте может редактировать информацию о пациенте или удалить её. Эти возможности на диаграмме прецедентов визуализируются двумя связями расширения <extend>. Режим редактирования информации также предполагает отдельный доступ к каждой группе данных в БД РЛТК по желанию пользователя, что визуализируется связями <extend>.

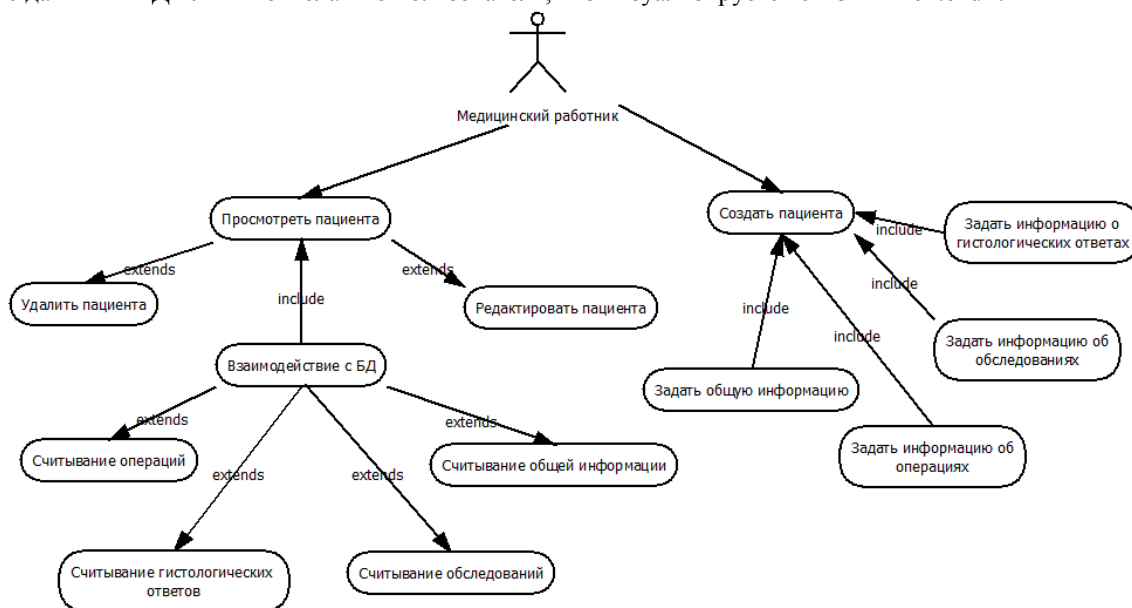


Рисунок 1 – Диаграмма прецедентов информационно-справочной системы с базой данных результатов лечения органов брюшной полости

Объектно-ориентированный анализ потоков управления каждого прецедента на рис.1 позволил составить логическую структуру информационной системы, которая представлена на рис.2.

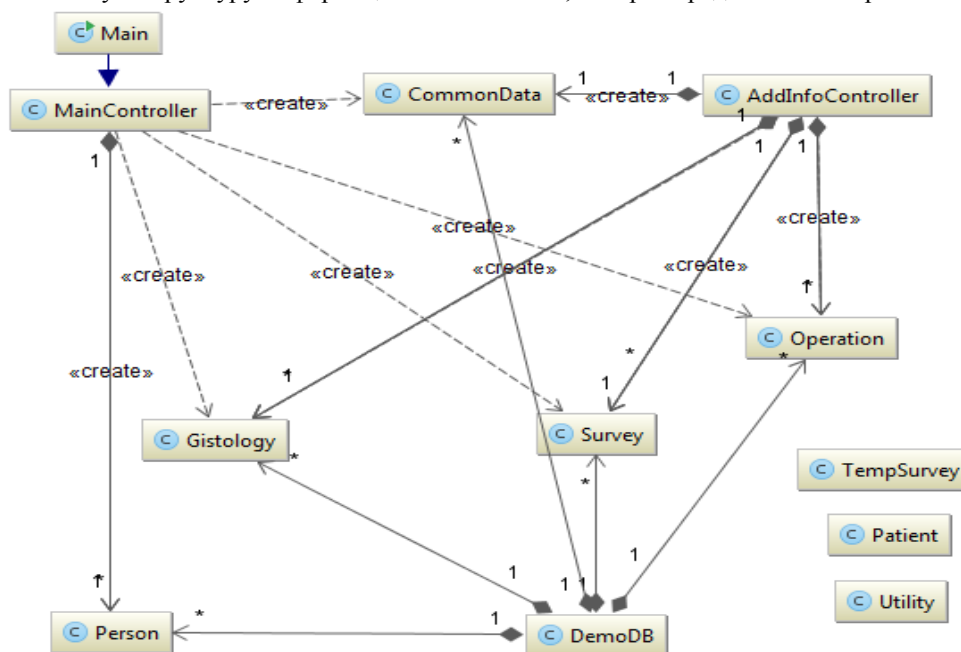


Рисунок 2 – Диаграмма классов информационной системы анализа лечения органов брюшной полости

Типовая декомпозиция выявила 12 классов, между которыми предусмотрены различные отношения (ассоциации, зависимости и агрегации), учитывающие специфику взаимодействия классов. Количественные отношения между классами могут быть двух видов: «один к одному» и «один ко многим».

Класс Main является основным классом, который запускает в работу информационную систему. Класс MainController отвечает за отображение главного окна приложения со всеми пациентами и полями поиска. Он также считывает из базы данных информацию и обрабатывает её, а также выводит информацию по пациентам в окно с помощью вспомогательного класса Person (поля класса содержат основную информацию о пациенте: ФИО, диагноз и операции).

Класс AddInfoController отвечает за отображение информации в окне пациента и за сохранение информации о пациенте в БД. Сохранение и отображение осуществляется с помощью вспомогательных классов: Operation (содержит информацию об операции), Survey (содержит информацию об обследованиях), Gistology (содержит информацию о гистологическом ответе).

Класс CommonData содержит в своих атрибутах общую информацию о пациенте, а именно: ФИО, адрес, пол, возраст и т.д.

Класс DemoDB – служебный, используется для локальных целей. Он собирает в одном месте всю информацию о пациенте.

Классы Utility, Patient и TempSurvey являются вспомогательными.

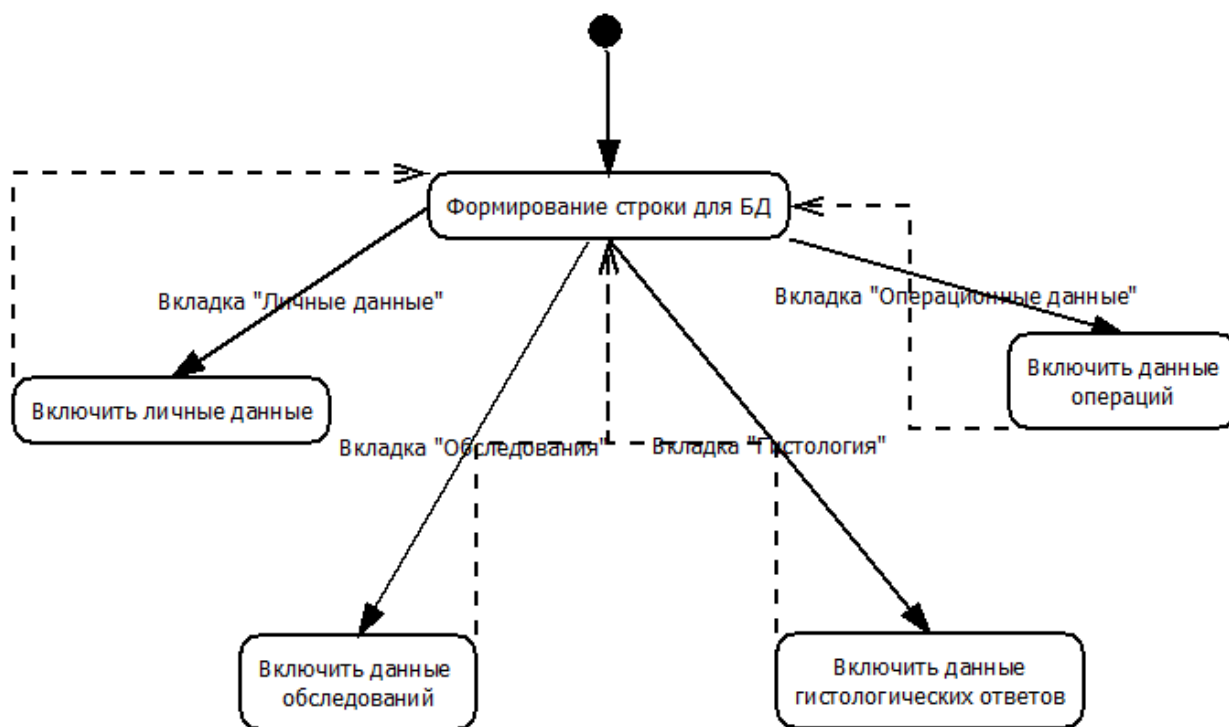


Рисунок 3 – Диаграмма состояний объекта класса AddInfoController

Жизненный цикл объекта AddInfoController описывается диаграмма состояний на рис.3. Названия состояний отражают деятельность объекта класса в каждом из них. В каждом состоянии объект заполняет часть записи соответствующими данными пациента, которая потом сохраняется в файле.

Имеющаяся при системе база данных позволяет в виде реляционных таблиц хранить сведения о пациентах: личные данные о пациента, данные обследования, данные об операциях и данные анализов. Опишем основные таблицы базы данных (см. рис.4). Таблица Patients хранит основную информацию о пациенте (ФИО, адрес, возраст, инвалидность и т.д.). Таблица Operations хранит информацию об операциях пациента (см. рис.5). Она содержит название операции, осложнения, возникшие при операции, тип наркоза, длительность операции, дату и тип операции. Таблица Gists хранит информацию о гистологических ответах пациента. Таблица Survey хранит информацию об обследованиях: типы обследований, дату, параметры исследований (анализы).

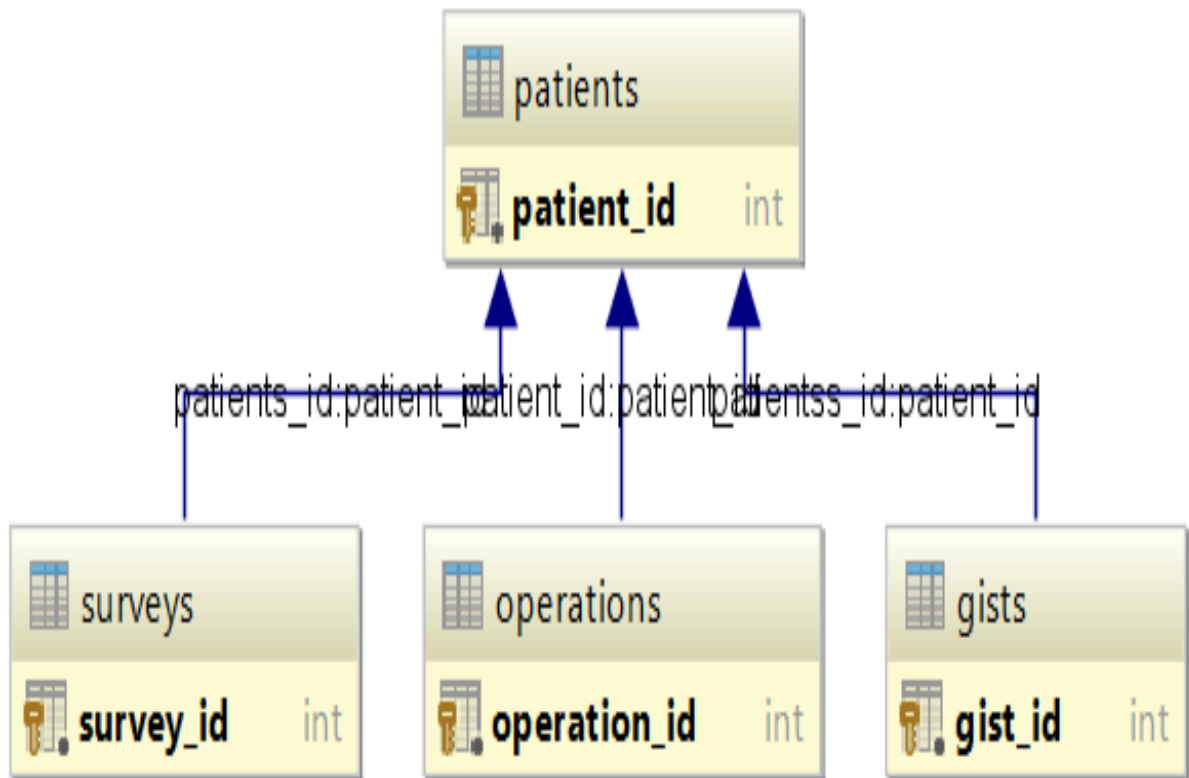


Рисунок 4 – Основные таблицы базы данных информационной системы

operations	
operation_id	int
patient_id	int
operation_text	varchar(255)
operation_complications	varchar(255)
operation_narcoz	varchar(20)
operation_lasts	smallint
operation_date	date
operation_type	varchar(20)

Рисунок 5 – Структура таблицы о перенесённых пациентом операциях

Диаграмма классов и базы данных сделана средствами IDE IntelliJ IDEA. Остальные диаграммы созданы в бесплатном приложении для создания диаграмм – Dia [12].

сделана указанная операция. Кнопка "Добавит пациента" переводит систему в режим ввода данных о новом пациенте.

Так как система разрабатывалась на объектно-ориентированном языке Java, то каждое окно представляет пограничным экземпляром класса, в котором запрограммирован внешний вид и функциональность соответствующего окна пользовательского интерфейса.

С помощью предусмотренных запросов система формирует более 11 различных статистических отчётов по имеющимся данным о пациентах с различными заболеваниями толстой кишки и хирургической патологией органов брюшной полости, которые выводятся на печать или сохраняются в виде файлов. Кроме того, из главного диалогового окна есть возможность просматривать всю информацию о клиентках, занесённых в архив медицинских данных.

Заключение

Проблемы в области медицины, которые, как известно, не имеют однозначных решений, сопровождаются большим объёмом информации. Именно поэтому для получения решения здесь целесообразно проводить анализ и моделирование ситуаций с помощью современных компьютерных средств - специализированных информационных систем. В статье описывается разработка одной из таких систем (ИС АЛТК), предназначенной для анализа лечения заболеваний толстой кишки и органов брюшной полости.

В клинике общей хирургии ДонНМУ им. М. Горького на базе хирургического отделения №2 ДОКТМО за период с 2012 по 2015 гг. проведено оперативное лечение 986 пациентам с патологией толстой кишки, из них 128 (12,8%) пациентам произведены СОВ. Было установлено, что течение послеоперационного периода у пациентов, перенесших СОВ, существенно не отличалось от такового после выполнения стандартных вмешательств на толстой кишке. Однако, обсуждение результатов лечения показало, что для определения целесообразности и эффективности выполнения данного типа хирургического вмешательства необходимо более тщательная сравнительная оценка результатов, выполняемых симультанных и стандартных оперативных вмешательств на основе достаточной большой статистической совокупности медицинских данных. Отсутствие единой информационной базы данных о результатах проведенных операций и автоматизированных средств интеллектуального анализа данных затрудняют объективную оценку СОВ и препятствуют их более широкому применению в клинической практике.

Поэтому в статье рассмотрен жизненный цикл проектирования информационной БД РЛТК в виде последовательности основных логических моделей на унифицированном языке моделирования UML, описывающие функции, логическую структуру и схему реляционной базы данных информационной системы.

Программная реализация информационной системы выполнена на языке программирования Java в инструментальной среде IDE IntelliJ IDEA. Система, построенная по открытому принципу, предусматривает возможность расширения своих функций. Пользовательский интерфейс спроектирован с учётом удобства работы с системой. По запросам пользователя система автоматически формирует более 11 различных статистических отчётов по имеющимся данным о пациентах с различными заболеваниями толстой кишки и хирургической патологией органов брюшной полости.

В заключение хотелось отметить аспекты, в которых может использоваться ИС АЛТК:

Во-первых, система работает как электронный архив, накапливая в структурированном виде информацию о показаниях и результатах лечения больных с ЗТК и хирургической патологией органов брюшной полости, которым выполнены СОВ. Она обеспечивает надёжность сохранения и достоверность информации.

Во-вторых, система может использоваться для статистической обработки собранной информации о проведенных медицинских операциях в количественных и процентных значениях по разным параметрам и их комбинациям.

В-третьих, разработанный программный продукт может использоваться для формирования более рациональных вариантов хирургического лечения данной категории больных на основе использования БД РЛТК и внедрения методов интеллектуального анализа данных (технологии Data Mining).

Наконец, *в-четвёртых*, созданный вариант медицинской информационно-справочной системы в дальнейшем будет функционально расширяться до системы поддержки принятия врачебных решений по лечению заболеваний желудочно-кишечного тракта.

Литература

1. Лебедев Г.С., Максаков В.В. Концепция портала «Медицинская наука» // Информационно-измерительные и управляющие системы. – 2014. - № 10, т.12. – С.1-6.
2. Евтихова Е.Ю., Кутырев Е.А., Гагуа А.К., Евтихов Р.М., Аль Бикай Рами А.А., Шурыгчн С.Н. Возможности симультанных операций при послеоперационных вентральных грыжах// Альманах клинической медицины, Ярославль,- 2008, - Том 17.- С.120-121.

3. Баулина, Н. В. Симультианные эндохирургические операции в гинекологии и хирургии Текст. : Тезисы 8-го Всероссийского съезда по эндоскопической хирургии // Эндоскопическая хирургия. - 2005. - С. 14-15.
4. Аль Бикай Рами А.А., Познанский С.В., Кукушкин А.В. Результаты симультианных операций у пациентов с желчнокаменной болезнью // Вестник Ивановской медицинской академии, 2008. - Том13. - № 3-4,- С. 87-88.
5. Маховский В. З. Одномоментные сочетанные операции в неотложной и плановой хирургии Текст. / В. З. Маховский // Хирургия. - 2002. - № 6. - С. 46.
6. Евтихова Е.Ю., Шурыгин С.Н., Грачев И.С., Муха А.В., Лебедева О.А., Аль Бикай Рами А.А., Сорокин С. Ю. Симультианные операции при патологии органов брюшной полости и послеоперационных вентральных грыжах// Вестник трансплантологии и искусственных органов. - 2006. - № 3. - С. 44-45.
7. Аль Бикай Рами А.А., Познанский С.В., Кукушкин А.В. Результаты симультианных операций у пациентов с желчнокаменной болезнью // Вестник Ивановской медицинской академии, 2008. - Том13. - № 3-4,- С. 87-88.
8. Евтихова Е.Ю. Шумаков Д.В., Гагуа АХ, Аль Бикай Рами А.А., Лебедева О.А., Евтихов Р.М. Симультианные операции при послеоперационных вентральных грыжах // Вестник Ивановской медицинской академии, 2007. - Том 12. - № 1-2, - С. 66-68.
9. Кузин, М. М. Лапароскопическая и традиционная холецистэктомия: сравнение непосредственных результатов Текст. / М. М. Кузин [и др.] // Хирургия. - 2000. - № 2. - С. 25—27.
10. Мюллер Р.Д. Базы данных и UML: Проектирование. – М.: Лори, 2002. - 420 с.
11. Джозеф Шмюллер. Освой самостоятельно UML 2 за 24 часа. Практическое руководство. - М.: Вильямс, 2005. - 416 с.
12. Васвани В. MySQL: использование и администрирование. - М.: «Питер», 2011. - 368 с.
13. Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL. Библия программиста, 2-е издание. - М.: «Диалектика», 2010. - 912 с.

Борота А.В., Борота А.А., Кухто А.П., Базиян-Кухто Н.К., Федяев О.И., Бакаленко В.С. Информационная система для анализа результатов лечения пациентов с воспалительными заболеваниями толстой кишки. В статье рассматривается разработка медицинской информационно-справочной системы, автоматизирующей процессы сбора, хранения, редактирования и предварительной статистической обработки данных для предоставления медицинским работникам необходимой информации о пред- и послеоперационных периодах лечения заболеваний толстой кишки и органов брюшной полости. Для решения данной задачи использовались современные методы и технологии построения информационных систем с базой данных реляционного типа. В результате построена объектно-ориентированная система открытого типа, которая в дальнейшем будет функционально расширяться до системы поддержки принятия врачебных решений с целью эффективного лечения заболеваний желудочно-кишечного тракта.

Ключевые слова: база данных, информационная система, UML-модели, медицина, органы брюшной полости, заболевание толстой кишки

Borota A.V., Borota A.A., Kukhto A.P., Baziyan-Kukhto N.K., Fedyaev O.I., Bakalenko V.S. Information system for the analysis of results of treatment of patients with inflammatory diseases of the colon. The article discusses the development of medical information and referral system that automates the collection of processes, storing, editing and preliminary statistical data for of health professionals the necessary information about the pre- and postoperative treatment of diseases of the colon and abdominal organs. To solve this problem, we used modern methods and technologies of construction information systems with relational database data type. As a result of built object-oriented system is open, which will be further functionally expand to support the adoption of a system of medical solutions to Effective treatment of diseases of the gastrointestinal tract.

Keywords: database, information system, the UML-model, medicine, organs abdominal bowel disease

СЕКЦИЯ «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ И ЗАЩИТА ДАННЫХ»

УДК 004.056.5

Обфускатор программного кода языка JavaScript

С.В. Медгаус, А.В. Чернышова

Донецкий национальный технический университет
medgaus-sergey@yandex.ru, alla@donntu.org

Медгаус С.В., Чернышова А.В. Обфускатор программного кода языка JavaScript. В тексте данной статьи представлен краткий обзор существующих обфускаторов для языка JavaScript. После их анализа предложено реализовать собственный обфускатор, применяющий сложные преобразования исходного кода. В статье подробно описаны применяющиеся алгоритмы запутывания.

Ключевые слова: обфускатор, Java, JavaScript, запутывание, обфускация, алгоритмы обфускации, Esprima, Nashorn

Введение

На данный момент проблема сохранения прав собственности на исходный код достаточно актуальна. В отличие от настольных приложений, в которых существуют различные средства защиты и лицензирования, в web-приложениях такие способы защиты не применимы и весь исходный код скриптов доступен для просмотра любому желающему через браузер.

Подробный анализ существующих способов защиты web-приложений рассмотрен в [1]. Исходя из изложенного в статье материала, можно сказать, что обфускация – это наименее ресурсоёмкий способ защиты исходного кода web-приложений. «Обфускация (от лат. obfuscare — затенять, затемнять; и англ. obfuscate — делать неочевидным, запутанным, сбивать с толку) или запутывание кода — приведение исходного текста или исполняемого кода программы к виду, сохраняющему его функциональность, но затрудняющему анализ и понимание алгоритмов работы.

Существуют специальные программы, производящие обфускацию, так называемые обфускаторы» [2].

На текущий момент существуют различные программные продукты, которые предлагают обфускацию исходного кода, однако эффективные алгоритмы обфускации предлагают только платные продукты, бесплатные же работают на уровне минификации кода (уменьшение размера исходного кода, путём сокращения имён переменных и функций, удаление символов форматирования кода). В работе [1] были рассмотрены и проанализированы такие полнофункциональные и эффективные программные продукты как YUI Compressor [3], Packer [4], JavaScript Obfuscator [5] и Google Closure Compiler [6]. После анализа существующих обфускаторов было принято решение разработать комбинацию эффективных обфусцирующих алгоритмов.

Целью работы является реализация алгоритмов обфускации для языка JavaScript с последующим объединением их в комплексный алгоритм обфусцирующего преобразования и использование его при создании обфускатора.

1 Проектирование обфускатора

После анализа существующих обфускаторов для языка JavaScript, была разработана диаграмма прецедентов для проектируемого обфускатора (см. рис. 1).

Диаграмма прецедентов UML – это диаграмма, показывающая связи между актёрами и действиями (прецедентами), которые они могут выполнять. Эта диаграмма позволяет точно описать полную функциональность программной системы [7].

Как видно из рис. 1, пользователь может загружать файл с исходным кодом, сохранять обфусцированный код в файл, запускать обфускацию, а также выбирать режимы обфускации:

- удаление форматирования;

- преобразование условных конструкций;
- логическое преобразование;
- сокращение констант;
- кодирование числовых констант;
- кодирование строковых констант;
- переименование переменных.

Все вышеперечисленные режимы обфускации применяются в заданном программой порядке, который невозможно изменить, то есть являются этапами. Это позволяет максимально эффективно запутать исходный код, так как появляются связи ещё и между самими режимами обфускации.

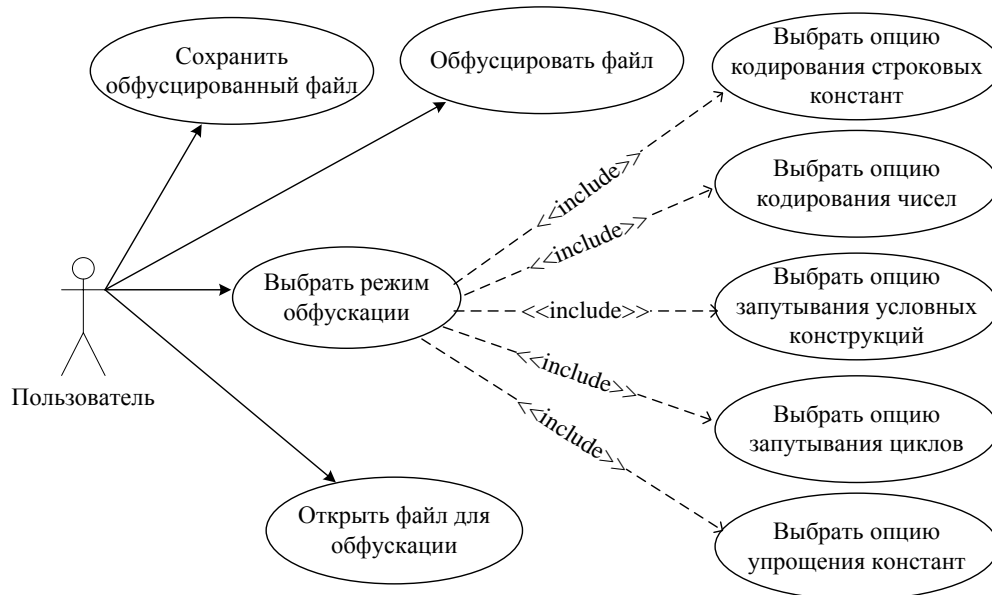


Рисунок 1 – Диаграмма прецедентов обфускатора

Исходя из спроектированной диаграммы прецедентов, была разработана диаграмма потоков данных обфускатора, которая представлена на рис. 2.

Приложение будет состоять из одной формы, в которой будет содержаться главный рабочий класс Обфускатор, который будет выполнять всю работу. В свою очередь, в зависимости от выбранных режимов, обфускатор будет использовать различные запутыватели, которые предоставляет программа. После обфускации программа будет выдавать выходной файл, который будет содержать обфусцированный JavaScript код.

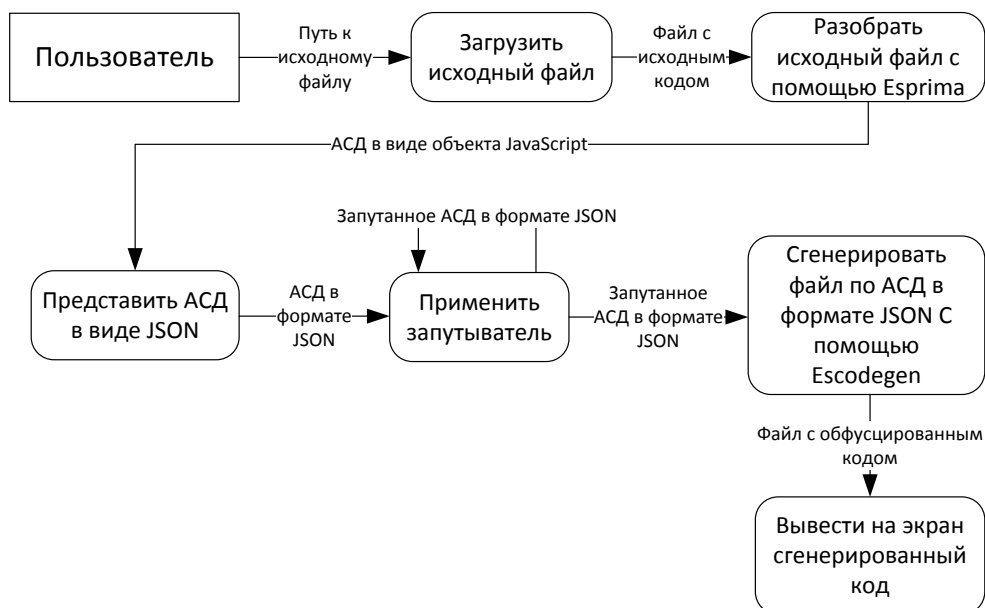


Рисунок 2 – Диаграмма потоков данных работы обфускатора

Все остальные диаграммы, разработанные при проектировании обфускатора, представлены в [8].

2 Алгоритмы обфусцирующих преобразований

Эффективность программы-обфускатора зависит от её алгоритмов, которые преобразовывают (запутывают) исходный код программ [9].

2.1 Общий алгоритм работы обфускатора

Необходимо отметить, что основным объединяющим алгоритмом программы является алгоритм, в котором объединяются все запутывающие алгоритмы и используются для приведения исходного кода в непонятный для человека вид (см. рис. 3).

```
подсчитать количество объектов-запутывателей;
подключить движок Nashorn;
распознать код с помощью Esprima;
перевести из формата Object в JSON с
    помощью встроенной библиотеки JSON;
распознать с помощью JSONSimple дерево в формате JSON;

FOR (для каждого запутывателя) {
    получить текстовое представление этапа;
    установить прогресс в зависимости
        от номера запутывателя;
    оповестить текстовым представлением;
    запутать код;
}

установить прогресс на последний этап;
опубликовать этап "Построение кода";

вызвать метод "Запутать" у объекта
    класса ПеремешивательФункций;
представить АСД в формате Object;
задать опции генерации кода, учитывая форматирование;
сгенерировать код с помощью библиотеки Escodegen;
оповестить об окончании обфускации;

вернуть обфусцированный код;
```

Рисунок 3 – Общий алгоритм работы обфускатора

2.2 Алгоритм преобразования условных выражений

Суть этого преобразования заключается в том, чтобы преобразовать допустимые условные выражения if-else в тернарный оператор. Этот алгоритм нацелен на ухудшение читаемости кода, без каких-либо фундаментальных преобразований. Алгоритм преобразования на псевдоязыке представлен на рис. 4.

2.3 Алгоритм логического преобразования

Данный алгоритм представляет собой простое инвертирование сравнений таким образом, чтобы вместо исходного оператора использовалось отрицание противоположного оператора с отрицаемыми операндами (см. рис. 5) [6]. На схеме алгоритма указана только часть логических операторов. Операторы, не попавшие на схему, преобразуются по такому же алгоритму.

Необходимо отметить, что можно использовать и другие логические преобразования, применяя логические эквиваленции, однако в JavaScript результатом логического выражения может быть не только true или false, но и ещё и объект, поэтому использование более сложных формул преобразований может приводить к ошибкам.

```

выполнить проход по АСД (
    IF (есть условные выражения с пустой веткой else) THEN
        устанавливаем флаг необходимости доп. функций;
)

IF (флаг необходимости доп. функций) {
    добавляем в массив функций случайное кол-во
    случайно сгенерированных функций;
    FOR (каждой функции) {
        сгенерировать случайное арифметическое выражение
        из нескольких арифметических операндов;
        представить выражение в виде строки;
        внести в тело функции в виде получения кода из
        символьных кодов;
        занести тело функций в конец исходного кода программы;
    }
}

выполнить проход по АСД (
    IF (текущий узел = "условное выражение" И
        (условное выражение не содержит вызовов
        методов объектов ИЛИ системных вызовов)) THEN {
        IF (ветка else пустая) {
            занести туда вызов сгенерированной функции;
        }
        поменять местами ветки if и else;
        поменять тип узла на "тернарный оператор";
    }
)

```

Рисунок 4 – Алгоритм преобразования условных выражений

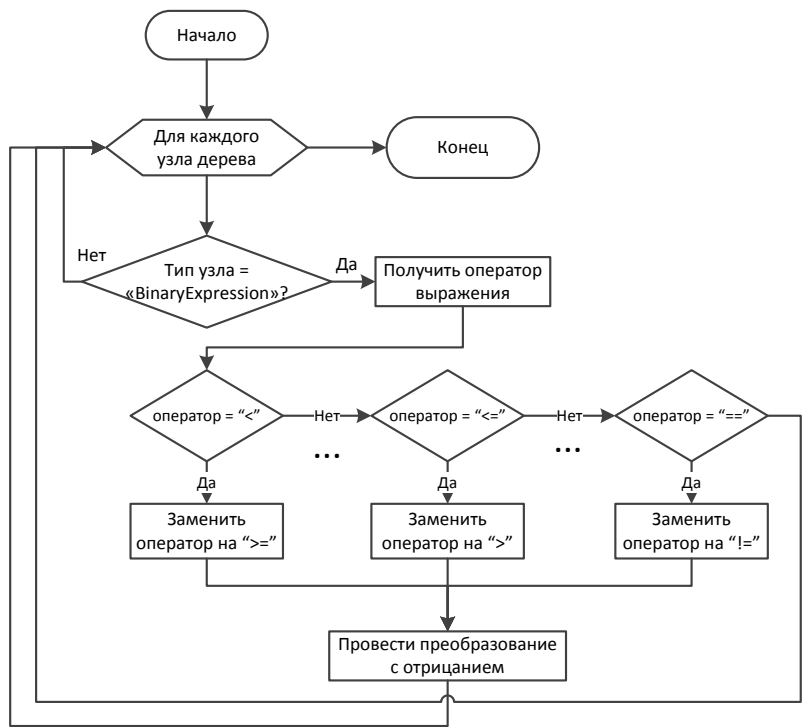


Рисунок 5 – Алгоритм преобразования логических выражений

2.4 Алгоритм сокращения констант

По стандарту ECMA Script 5.0 в языке JavaScript ещё нет констант, которые бы обеспечивал сам язык, поэтому программа сама анализирует переменные, которые не изменяют своего значения в исходном коде пользовательской программы.

Данное преобразование состоит из следующих этапов:

- найти возможные константы в коде;
- исключить меняющиеся переменные;
- исключить переменные, чьё значение не является простым, то есть не является числом, строкой или булевым значением;
- заменить в местах использования констант их значением.

Детальное описание алгоритма представлено на рис. 6.

```
FOR (для каждого узла дерева) {
  IF (узел - это объявление переменной) THEN {
    .....
    получить имя переменной;
    поместить в кеш-таблицу имя переменной и её значение;
  }
}
FOR (для каждого узла дерева) {
  IF (узел используется с оператором
    .....
    инкремента или декремента) THEN {
    .....
    удалить переменную из кеш-таблицы;
  }
  IF (узел - присвоение переменной значения) THEN {
    IF (переменная неинициализированная) THEN {
      .....
      присвоить новое значение и занести в кеш-таблицу;
    } ELSE {
      .....
      удалить переменную из кеш-таблицы;
    }
  }
}
FOR (для каждой переменной в кеш-таблице) {
  IF (инициализированное значение не простое) THEN {
    .....
    удалить переменную из кеш-таблицы;
  }
}
FOR (для каждого узла дерева) {
  IF (используются переменная, которая есть
    .....
    в кеш-таблице) THEN {
    .....
    заменить использование переменной её значением;
  }
}
FOR (для каждого узла дерева) {
  IF (объявление переменной, которая есть в кеш-таблице) THEN {
    .....
    удалить объявление переменной;
  }
}
```

Рисунок 6 – Алгоритм сокращения констант

2.5 Алгоритм кодирования чисел

Данный алгоритм очень важен для программной системы, так как большая часть исходного кода в любом языке программирования использует огромное количество чисел для счётчиков различного рода, для циклов и для числовых констант.

Суть этого алгоритма заключается в том, чтобы заменить числа некоторым набором арифметических выражений, и после этого представить все числа в 16-ичной системе счисления [6]. Детальный алгоритм представлен на рис. 7.

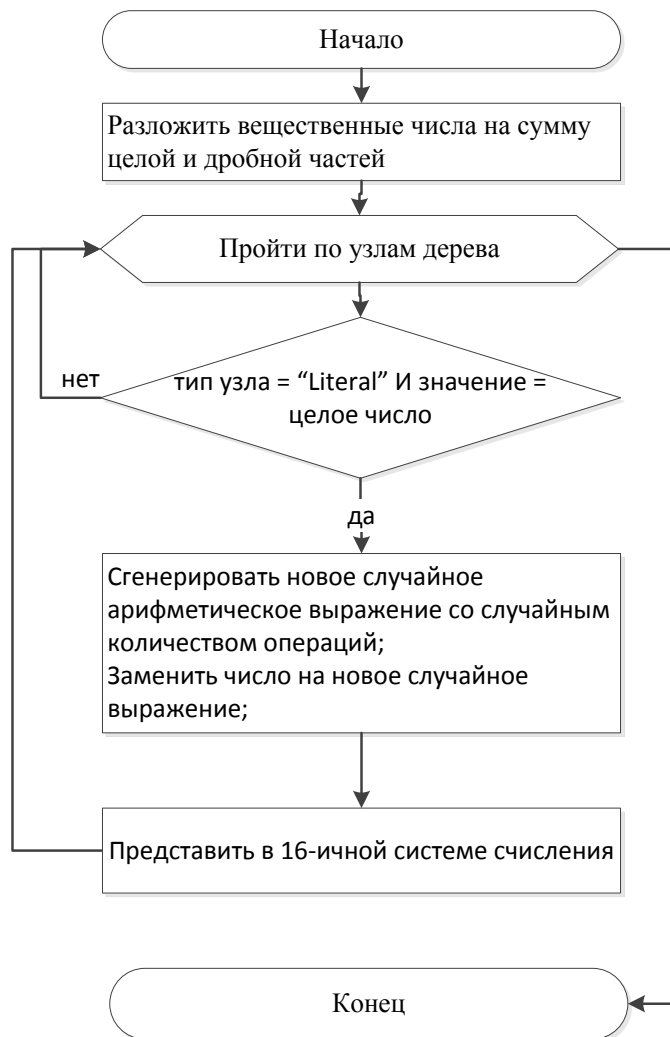


Рисунок 7 – Алгоритм кодирования чисел

2.6 Алгоритм кодирования строк

Данный алгоритм также является важным в работе программной системы. Почти во всех программах используются какие-либо строковые константы, будь-то сообщения об ошибках или текстовые элементы интерфейса. Для того чтобы труднее было разобрать исходный код программы, необходимо преобразовать и скрыть явное использование таких строковых констант.

В данном преобразовании строки представляются в виде конкатенации вызовов различных функций, а также в виде кодировки BASE64. Все разбиение строковых констант происходит каждый раз случайно, обеспечивая различный исходный код на выходе обфускатора [7].

Детальный алгоритм представлен на рис. 8.

2.7 Алгоритм переименования переменных

Данное преобразование исходного кода является самым не затратным с точки зрения ресурсов преобразования, но с точки зрения конечного запутывания является одним из самых эффективных алгоритмов обфускации.

Главная особенность этого алгоритма в том, чтобы представить имена переменных и функций в виде непонятного для человека набора символов [6]. Подробный разбор алгоритма представлен на рис. 9.

```

FOR (для каждого узла дерева) {
    случайно получить флаг "необходимо ли делить слово";
    IF (узел - это строковый литерал И
        есть флаг "делить слово") THEN {
        получить длину строки;
        получить случайное число до длины строки;
        разбить слово на два по случайному числу;
    }
}
FOR (для каждого узла дерева) {
    IF (узел - это строковый литерал) THEN {
        занести каждую строковую константу в массив констант;
    }
}
FOR (для каждого узла дерева) {
    случайно получить флаг "необходимо
    ли выносить слово в отдельную функцию";
    IF (узел - это строковый литерал И
        есть флаг "выносить слово") THEN {
        заменить использование строковой константы на
        вызов отдельной функции;
        установить флаг "отдельная функция использовалась";
    }
}
IF ("отдельная функция использовалась") {
    занести в конец исходного кода тело отдельной функции
    для строковых констант;
}
FOR (для каждого узла дерева) {
    случайно получить флаг "необходимо ли преобразовать в Base64";
    IF (узел - это строковый литерал И
        есть флаг "необходимо преобразовать") THEN {
        заменить использование строковой константы на
        закодированное значение в Base64 и вызов функции
        декодирования;
    }
}
}

```

Рисунок 8 – Алгоритм кодирования строковых констант

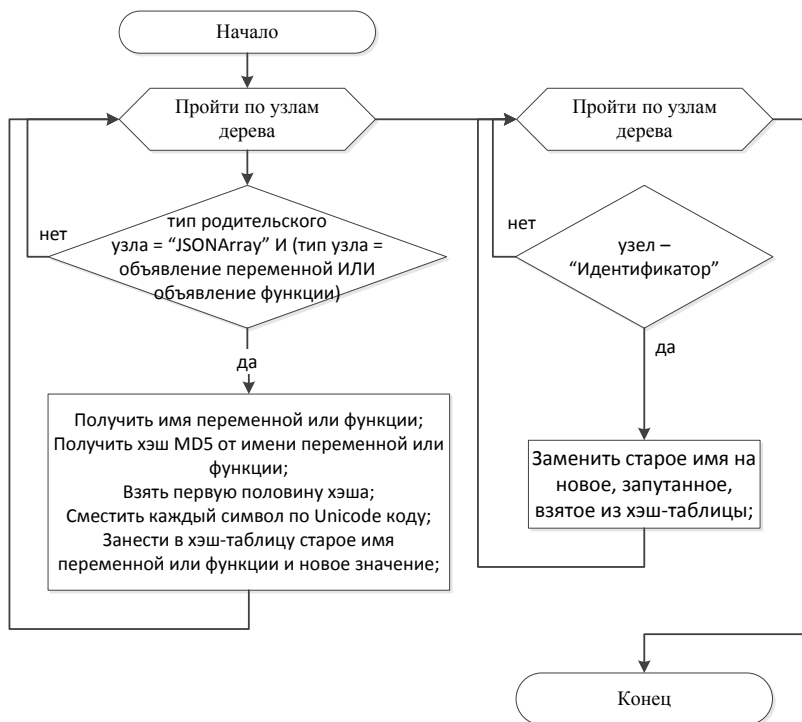


Рисунок 9 – Алгоритм переименования переменных

2.8 Алгоритм перемешивания функций

Данный алгоритм не является напрямую запутывающим преобразованием, его нельзя выбрать через интерфейс пользователя. Он служит для того, чтобы перемешивать глобальные функции, которые могли появиться вследствие других запутывающих преобразований (мусорные функции). Иначе, если это преобразование не использовать, то мусорные функции располагались бы в конце документа с исходным кодом и могли бы легко быть удалены человеком, который взялся бы деобфусцировать исходный код. Детальное описание алгоритма представлено на рис. 10.

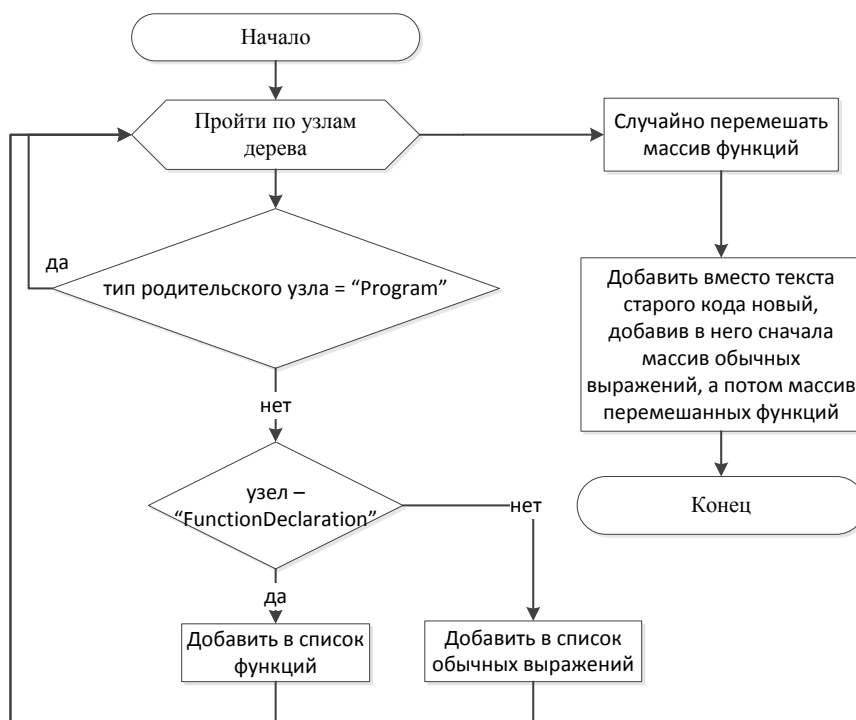


Рисунок 10 – Алгоритм перемешивания функций

3 Результат разработки программы-обфускатора

После разработки алгоритмов, они были реализованы на языке Java, где и были объединены в один главный обфусцирующий алгоритм.

На рис. 11 представлена экранная форма обфускатора, на которой видно, что пользователь выбрал все доступные режимы обфускации. Также видно, что исходный код существенно изменился.

Как показало тестирование на экспертных группах, код стал более запутанным. Эксперты сделали субъективное заключение – сложность понимания исходного кода увеличилась в 3 раза.

Выводы

В ходе выполнения данной работы были проанализированы существующие обфускаторы языка JavaScript. После их детального изучения, были выявлены присущие им недостатки, учитывая которые, было решено разработать и реализовать собственные эффективные алгоритмы обфускации.

Также была разработана последовательность объединения запутывающих преобразований (общий алгоритм работы обфускатора) с максимальной степенью запутанности.

Дальнейшие разработка и реализация алгоритмов обфускации являются перспективными, так как разработанные алгоритмы работают напрямую с кодом, меняя внешнее представление кода, в то время как, преобразования потока управления программ не производится. То есть разработанные алгоритмы принадлежат к категориям лёгкой и средней степеням обфускации, а алгоритмы обфускации сложной степени необходимо разрабатывать.

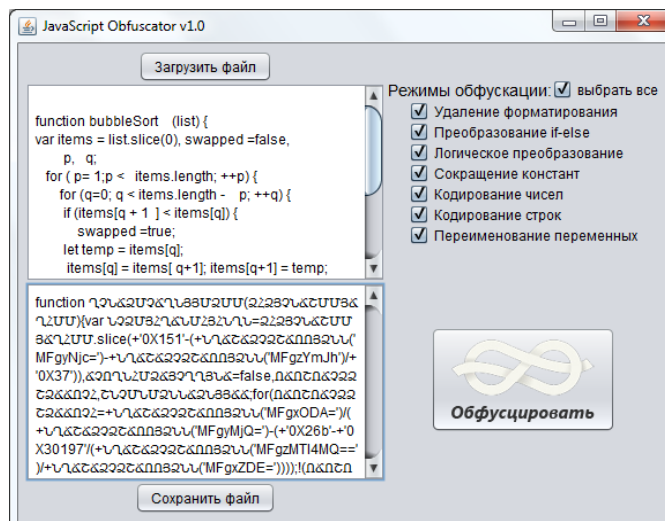


Рисунок 11 – Экранная форма обфускатора

Литература

1. Обзор существующих обфускаторов и их алгоритмов. Компьютерная и программная инженерия - 2015 год: - Донецк: ДонНТУ, 2015. – С.117-119.
2. Википедия. Обфускация [электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Обфускация>.
3. YUI Compressor [электронный ресурс]. – Режим доступа: <https://github.com/yui/yuicompressor>.
4. Packer [электронный ресурс]. – Режим доступа: <http://dean.edwards.name/packer>.
5. JavaScriptObfuscator. Canada [электронный ресурс]. – Режим доступа: <https://www.javascriptobfuscator.com/>.
6. Google Closure Compiler [электронный ресурс]. – Режим доступа: <https://developers.google.com/closure/compiler/>.
7. Диаграммы прецедентов: крупным планом [электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/1007/229/lecture/5962>.
8. Проектирование обфускатора для языка JavaScript. ИУСМКМ - 2016: VII Международная научно-техническая конференция, 26 мая 2016: - Донецк: ДонНТУ, 2016. – С. 167-173.
9. LynX. Обфускация и защита программных продуктов // CITForum. [электронный ресурс]. – Режим доступа: <http://citforum.ru/security/articles/obfus>.
10. Java Obfuscator – String Encryption // zelix.com [электронный ресурс]. – Режим доступа: <http://www.zelix.com/klassmaster/featuresStringEncryption.html>.

Медгаус С.В., Чернышова А.В. Обфускатор программного кода языка JavaScript. В тексте данной статьи представлен краткий обзор существующих обфускаторов для языка JavaScript. После их анализа, предложено реализовать собственный обфускатор, применяющий сложные преобразования исходного кода. В статье подробно описаны применяющиеся алгоритмы запутывания.

Ключевые слова: обфускатор, Java, JavaScript, запутывание, обфускация, алгоритмы обфускации, Esprima, Nashorn

Medgaus Sergey, Chernyshova Alla Obfuscator of JavaScript source code. There is presented short overview of existing obfuscators for JavaScript language in the text of this article. It is proposed to realize own obfuscator that uses complex transformation of source code. There are described different applied mangling algorithms.

Key words: obfuscator, Java, JavaScript, mangling, obfuscation, obfuscating algorithms, Esprima, Nashorn

Разработка модуля аутентификации PDM-подсистемы в САПР-компонентах САПР с открытым кодом

Е.С. Калинина, А.В. Чернышова
Донецкий национальный технический университет
lena00777@gmail.com, alla@donntu.org

Калинина Е.С., Чернышова А.В. Разработка модуля аутентификации PDM-подсистемы в САПР-компонентах САПР с открытым кодом. В статье представлен обзор средств аутентификации в PDM-подсистемах САПР, представлено описание разработанного программного модуля аутентификации в САПР-системе с открытым кодом.

Ключевые слова: статья, требования, оформление, сборник трудов.

Введение

Цель работы: исследование алгоритмов аутентификации, проектирование и реализация модуля аутентификации с оценкой сложности пароля в PDM-подсистеме САПР с открытым кодом.

Актуальность работы: в последнее время актуальным становится необходимость обеспечить эффективную защиту от несанкционированного доступа к проектным данным и документации, а также их нежелательной модификации. Одним из таких способов защиты информации является аутентификация пользователя, обеспечивающая разграничение доступа к информации в системах. Предлагаемый программный модуль аутентификации пользователя с оценкой сложности пароля может использоваться как средство защиты файлов проектов в САПР с открытым кодом.

Для корректной аутентификации пользователя необходимо, чтобы пользователь предъявил аутентификационную информацию – некую уникальную информацию, которой должен обладать только он и никто иной.

Существует три основных типа аутентификационной информации:

- проверяемый пользователь знает некую уникальную информацию (парольная аутентификация);
- пользователь имеет некий предмет с уникальными характеристиками или содержимым (смарт-карта, USB-токен и т.д);
- аутентификационная информация является неотъемлемой частью пользователя (отпечаток пальца, геометрия руки, лица, голос и другие виды биометрической аутентификации).

В любом из этих случаев процедура аутентификации выполняется в два следующих этапа:

1 этап. У пользователя однократно берется некий эталонный образец аутентификационной информации, например, запрашивается пароль (или данный образец генерируется случайным образом и затем записывается на смарт-карту пользователя). Данный образец хранится у субъекта системы, проверяющего аутентификацию – модуля аутентификации (например, сервера, который выполняет аутентификацию пользователей). Обычно существует некоторое время действия данного эталона, по завершении которого эталонный образец перезапрашивается.

2 этап. Каждый раз, при выполнении аутентификации, у пользователя запрашивается аутентификационная информация, которая сравнивается с эталоном. На основе данного сравнения делается вывод о подлинности пользователя [1].

Обзор средств аутентификации в PDM подсистемах САПР SolidWorks, ADEM

На сегодняшний день в САПР-системах (SolidWorks, ADEM) есть PDM-подсистемы, направленные на управление информацией об изделии и управление коллективной работой.

В САПР-системе SolidWorks на сегодняшний день существует SWR-PDM система, которая представляет собой решение задачи управления данными, поиск, извлечение, быстрый просмотр и управление версиями документа. Система построена по модульному принципу и состоит из следующих частей:

- SWR-PDM (базовая конфигурация);
- модуль ProductStructure (состав изделия);
- модуль LifeCycle (жизненный цикл изделия);

- модуль WorkFlow (управление потоком заданий).

SWR-PDM разрабатывается с использованием современных достижений в технологиях PDM-систем и программирования, реализуется на основе объектно-ориентированных технологий, а также включает тесно взаимосвязанные подсистемы, выполняющие следующие функции:

- управление документами (возможность хранить оригиналы документов в защищённом хранилище, мониторинг, контроль и запись изменений документов);
- управление потоками заданий, жизненным циклом документации, изделий, их экземпляров и партий;
- разграничение доступа на основе дискреционного и мандатного принципов контроля для реализации правил выполнения операций пользователями над объектами хранилища.

SWR-PDM позволяет большому количеству пользователей с разными полномочиями хранить и обрабатывать документы в едином хранилище независимо от остальных пользователей, предоставляя или запрещая доступ к данным со стороны других пользователей. Наряду с этим SWR-PDM позволяет одновременно хранить документы различного уровня секретности: документы общего доступа, секретные документы, совершенно секретные документы и т.д. Список уровней доступа настраивается администратором защиты [2].

CAD/CAMADEM — инструмент, с помощью которого инженер (конструктор или технолог) автоматизирует свой труд и повышает его производительность. Т.к. со временем информация имеет свойство накапливаться и разобратся в ней становится проблематично, то возникает потребность в её упорядочении, что достигается путём структурирования информации за счёт использования автоматизированных систем документооборота и электронного архива. Функционально архив состоит из трех частей: интерфейсной части архива, собственно базы данных (БД) и логической части, так называемой бизнес-логики. Перед началом работы пользователь должен пройти авторизацию, указав свои имя и пароль, лишь после этого будет открыт доступ к архиву.

В БД хранится только учётная и поисковая информация, а вся основная информация (CAD, CAM, TDM) хранится в виде файлов на диске. Такая организация позволяет существенно сократить размер БД, что, в свою очередь, скажется на скорости работы с архивом.

Бизнес-логика осуществляет связь между интерфейсной частью архива и БД. Она представляет собой сценарии (алгоритмы), для написания которых используется специализированный язык системы AdemTDM.

Администрирование архива включает в себя работу с пользователями. Оно построено следующим образом: каждый пользователь принадлежит к определенной группе, а группа, в свою очередь, обладает набором тех или иных прав доступа, согласно которым пользователи группы могут либо не могут совершать те или иные действия с архивом, а именно:

- создание, удаление пользователей. При создании пользователя задаются имя пользователя и пароль для доступа к архиву, устанавливается группа, членом которой он будет являться;
- создание, удаление групп пользователей. При создании группы необходимо определить имя группы и указать требуемые права. Имеются четыре предопределенные группы пользователей, обладающие различным набором прав: администратор, ведущий инженер, инженер, гость [3].

Описание структуры разрабатываемого модуля

При разработке модуля аутентификации PDM-подсистемы в САПР с открытым кодом необходимо определить требования к разрабатываемому модулю.

На первом этапе использования модуля пользователю сначала необходимо зарегистрироваться, введя логин и пароль в диалоговое окно. Введенный пользователем логин должен пройти проверку на валидность, а пароль должен удовлетворять определённым требованиям и быть достаточно стойким. При успешной регистрации данные, введенные пользователем, вносятся в базу данных.

При последующем использовании модуля пользователь должен будет войти в систему, т.е. пройти процесс аутентификации. Введенные им логин и пароль, который впоследствии хэшируется, будут сравниваться с теми, которые уже есть в базе данных. В случае совпадения данных пользователь входит в систему, иначе процесс аутентификации (по методу «запрос-ответ») придётся повторить.

На рис. 1 представлена диаграмма вариантов использования разрабатываемого модуля аутентификации.

Разрабатываемая система аутентификации построена на основе двухзвенной архитектуры. Двухзвенной (two-tier, 2-tier) она называется из-за необходимости распределения трех базовых компонентов между двумя узлами (клиентом и сервером). Такая архитектура используется в клиент-серверных системах, где сервер отвечает на клиентские запросы напрямую и в полном объёме, используя только собственные ресурсы [4].

Модель клиент/сервер позволяет разместить данное приложение или клиент на одном компьютере, а сервер базы данных - на другом, но они также могут находиться на одном и том же компьютере. Разрабатываемый модуль будет работать в любом из этих случаев.

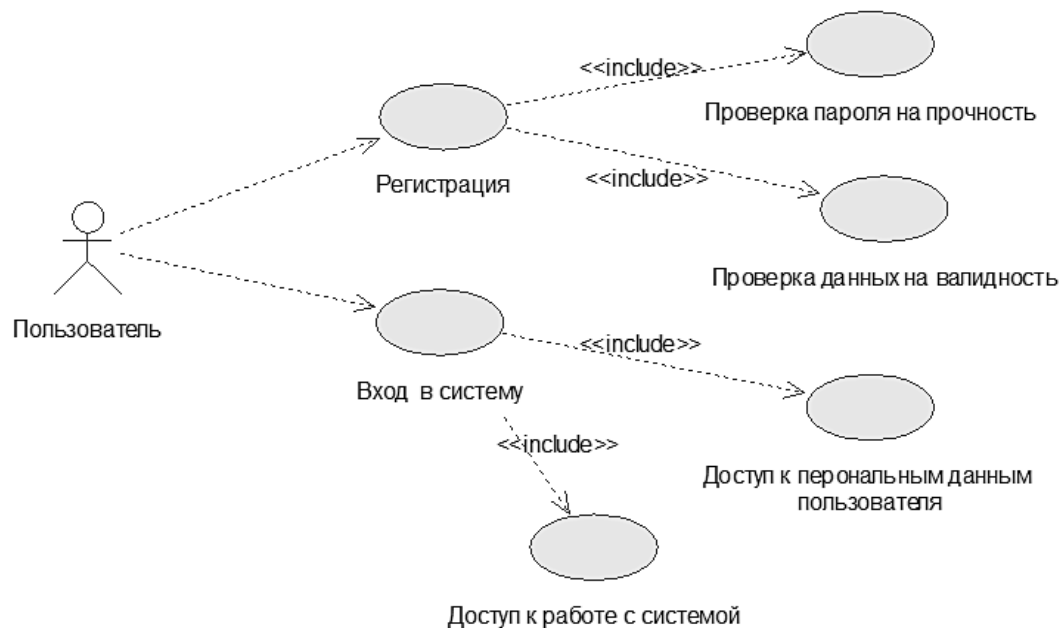


Рисунок 1 – Диаграмма вариантов использования разрабатываемого модуля

Процесс работы разрабатываемой системы происходит следующим образом: пользователь (клиент) вводит персональные данные в форму приложения и нажатием кнопки («Signin») отправляет их серверу. В том случае, если данные удовлетворяют поставленным системой требованиям, то они вносятся в базу данных, а пользователь (клиент) получает сообщение от сервера об успешной регистрации, в противном случае сообщение об ошибке.

В том случае, если пользователь уже зарегистрирован в системе, то ему необходимо пройти процесс аутентификации, который происходит следующим образом. Пользователь переходит на вкладку «Login» и в появившихся полях осуществляет вход в систему введя логин и пароль. По нажатию кнопки начинается процесс авторизации. Введенный пользователем пароль хэшируется, на «сервере» генерируется некое псевдослучайное число, которое хэшируется по алгоритму MD5 вместе с хэшем пароля (q).

В базе данных, которая содержит данные зарегистрированных ранее пользователей, осуществляется поиск логина, введенного пользователем для входа в систему. В том случае, если такой логин найден в базе данных, то от соответствующего ему пароля берётся хэш, который в последствии хэшируется по алгоритму MD5 [5] со сгенерированным числом (q'). В том случае, если «q = q'», пользователь входит в систему, иначе получает сообщение об ошибке и повторяет процесс аутентификации заново, введя свой логин и пароль.

На рисунке 2 представлена диаграмма последовательности процесса аутентификации в системе.

Схема аутентификации «запрос-ответ», которая используется в разработке данного модуля, защищает от несанкционированного воспроизведения пароля, поскольку для каждого сеанса аутентификации сервер генерирует уникальный challenge (псевдослучайное число).

Преимущество такого метода в том, что используется безопасное хранение паролей (в зашифрованном виде, или, что удобнее, в хэшированном виде). При проверке введенного пароля система вычисляет его хэш-образ и сравнивает результат с хранящимся хэш-образом пароля, соответствующим логину пользователя в таблице паролей [6].

В процессе реализации модуля в классе Main был разработан графический интерфейс программы, который имеет следующие методы:

- start();
- formLog();
- formSign();
- signAction();
- indicAction();
- butAction();
- opname().

Для взаимодействия программы с базой данных был создан класс UserController, который связывается с MySQLDatabase и позволяет сохранять и работать с данными пользователей. Также предусмотрена проверка

данных, введенных пользователем, на валидность при регистрации (пароль, логин, телефон и е-мейл). Для каждого из этих случаев реализован специальный класс:

- UsernameValidator;
- PhoneValidator;
- EmailValidator;
- PasswordStrength.

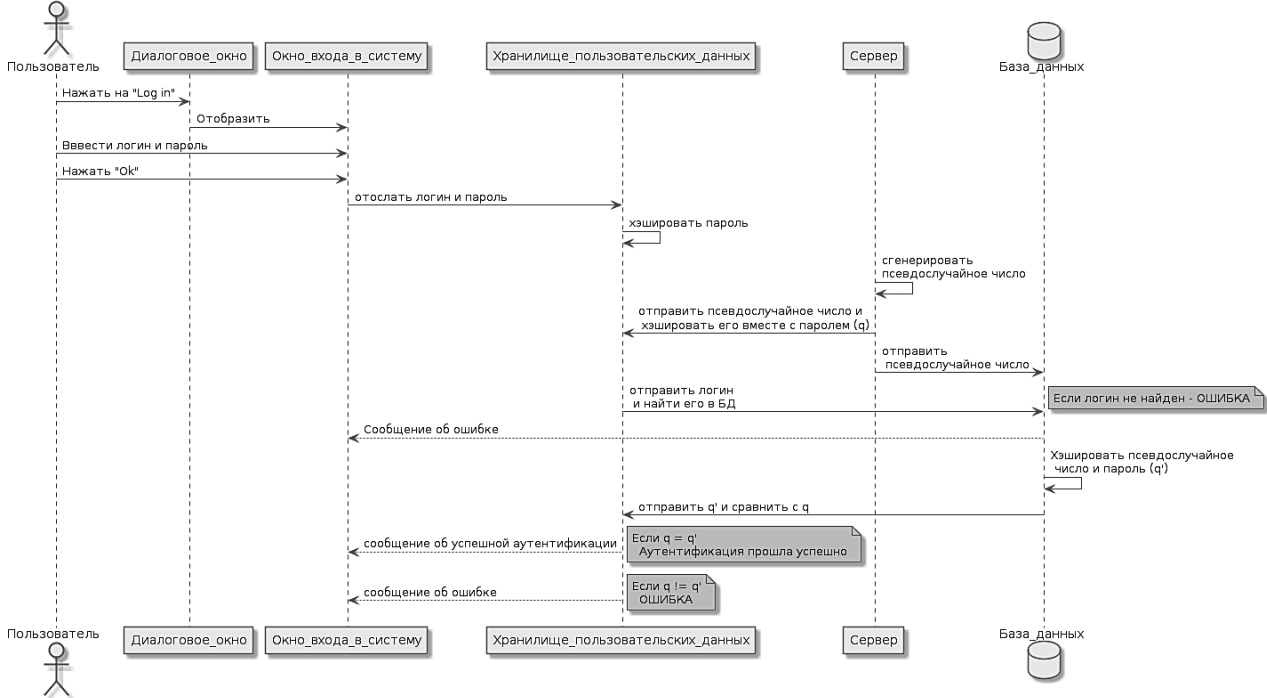


Рисунок 2 – Диаграмма последовательности процесса аутентификации

На рис. 3 представлена диаграмма классов разрабатываемого модуля.

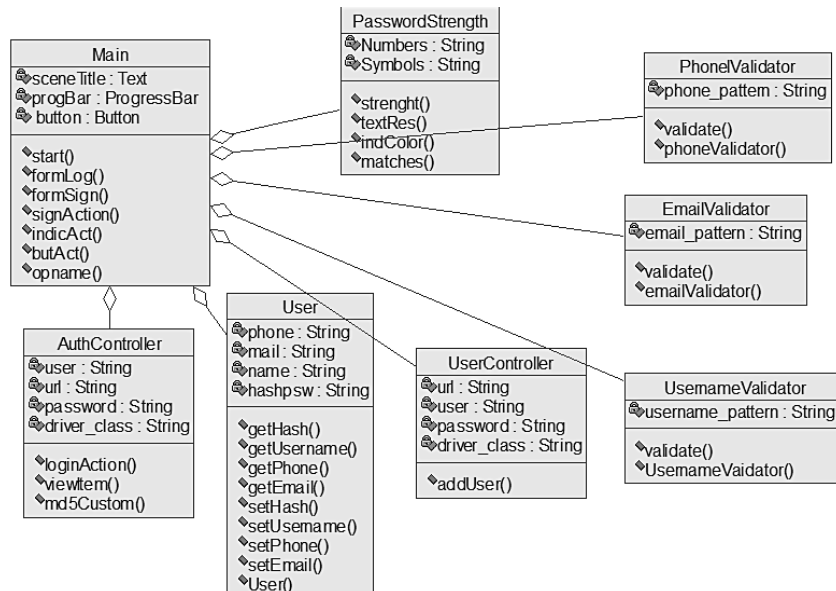


Рисунок 3 – Диаграмма классов разрабатываемого модуля

В классе AuthController реализован процесс аутентификации с обращением к базе данных.

Таблица 1. Структура класса User.

Параметры класса	
privateStringusername	Строка, содержащая логин пользователя
privateStringhashpsw	Строка, содержащая хэш от пароля пользователя
privateStringphone	Строка, содержащая телефон пользователя
privateStringemail	Строка, содержащая емейл пользователя
Методы класса	
publicUser(Stringusername, Stringhashpsw, Stringphone, Stringemail)	Конструктор класса, содержащий все параметры пользователя и передающий их в БД
publicStringgetUsername() publicStringgetHashpsw() publicStringgetEmail() publicStringgetPhone()	Методы, позволяющие получать значения вводимые пользователем
setUsername(Stringusername) setHashpsw(Stringhashpsw) setEmail(Stringemail) publicvoidsetPhone(Stringphone)	Методы, записывающие соответствующие значения в переменные

Класс User содержит в себе данные о пользователе, которые впоследствии заносятся в базу данных. Проверка данных о пользователе осуществляется с помощью регулярных выражений.

Логин пользователя проверяется на валидность с помощью регулярного выражения, которое представлено на рис. 4, т.е. логин должен содержать латинские буквы нижнего регистра, цифры или спецсимволы «-_», количество символов от 3 до 15.

```
USERNAME_PATTERN = "^[a-z0-9_-]{3,15}$"
```

Рисунок 4 – Регулярное выражение для проверки логина

Е-мейл пользователя проверяется с помощью следующего регулярного выражения, которое представлено на рис. 5 [8]. Т.е., адрес должен начинаться с цифры или латинской буквы верхнего/нижнего регистра, затем аналогичный набор из любого количества символов, знак @ и доменное имя.

```
String EMAIL_PATTERN =
    "^[A-Za-z0-9-\\+]+(\\.[A-Za-z0-9-]+)*@"
    + "[A-Za-z0-9-]+(\\.[A-Za-z0-9-]+)*(\\.[A-Za-z]{2,})$";
```

Рисунок 5 – Регулярное выражение для проверки е-мейла пользователя

Номер телефона должен удовлетворять модели (xxx-xxxxxxx), где x – цифры. Регулярное выражение представлено на рис. 6.

```
String Phone_PATTERN = "\\d{3}-\\d{7}"
```

Рисунок 6 – Регулярное выражение для проверки телефона

Процесс регистрации происходит следующим образом: пользователь открывает форму и переходит на вкладку «Signin». В появившихся полях он вводит личные данные (логин, пароль, е-мейл и телефон) и нажимает на кнопку «Signin». Введенные пользователем данные проверяются, ведётся оценка сложности пароля в процентах, которую можно увидеть, нажав на индикатор, если данные прошли проверку на валидность и пароль пользователя относится к степени «Strong» или «Verystrong». В том случае, если пароль состоит только из символов, его стойкость не будет достигать значения «Strong», т.е. чем больше значений алфавита задействовано, тем сильнее стойкость пароля.

Стойкость пароля вычисляется по следующему принципу [6]:

Изначально стойкости (strength) присваивается значение, равное длине пароля, помноженного на 4 (1).

$$Z = l_n * 4, \quad (1)$$

где l_n – длина пароля, Z – стойкость.

После этого осуществляются следующие проверки: в том случае, если в пароле присутствуют латинские буквы верхнего/нижнего регистра, то стойкость вычисляется по формуле 2.

$$Z = Z + l_n - k, \quad (2)$$

где l_n – длина пароля, Z – стойкость, k – количество букв верхнего/нижнего регистра.

Если в пароле присутствуют цифры, то стойкость вычисляется по формуле 3.

$$Z = Z + k_n * 4, \quad (3)$$

где k_n – количество цифр, Z – стойкость.

В том случае, если количество цифр в пароле равно длине самого пароля (пароль состоит из цифр), стойкость вычисляется по формуле 4.

$$Z = Z + l_n - k_n, \quad (4)$$

где l_n – длина пароля, k_n – количество цифр, Z – стойкость.

Если в пароле присутствуют спецсимволы, то стойкость вычисляется по формуле 5.

$$Z = Z + k_c * 6, \quad (5)$$

где k_c – количество спецсимволов, Z – стойкость.

Если в середине пароля присутствуют цифры или спецсимволы (формула 6), то стойкость вычисляется по формуле 7.

$$i > 0 \ \&\& \ i < l_n - 1, \quad (6)$$

где l_n – длина пароля, i – индекс элемента.

$$Z = Z + k_c * 2, \quad (7)$$

где k_c – количество спецсимволов/цифр, Z – стойкость.

В случае, если последовательно находятся два символа верхнего/нижнего регистра или 2 цифры, то стойкость вычисляется по формуле 8.

$$Z = Z - k_{вп} * 2, \quad (8)$$

где $k_{вп}$ – количество встречающихся пар символов/цифр, Z – стойкость.

Если в пароле по порядку или в обратном порядке находятся 3 цифры (123 || 987), стойкость вычисляется по формуле 9.

$$Z = Z - 3, \quad (9)$$

где Z – стойкость.

Далее перечислены условия, которые накапливаются как бонусы, т.е., при совпадении условия бонус наращивается на 1 (изначально его значение равно 0) и впоследствии, основываясь на количестве бонусов, вычисляется значение стойкости.

Бонус наращивается на 1, в случае если длина пароля больше или равна 8 (10).

$$l_n \geq 8 \rightarrow B ++, \quad (10)$$

где l_n – длина пароля, B – бонус.

В случае, если количество символов верхнего/нижнего регистра, цифр или спецсимволов поделенное на длину пароля больше заданного значения (11).

$$\frac{k}{l_n} \geq 0,25 \rightarrow B ++, \quad (11)$$

где l_n – длина пароля, k – количество букв верхнего/нижнего регистра, цифр, спецсимволов, B – бонус.

После вычисления бонусов окончательное значение стойкости вычисляется по формуле 12.

$$Z = Z + B * 2, \quad (12)$$

где Z – стойкость, B – бонусы.

Стойкость может принимать значения от 0 до 100.

В зависимости от полученного значения паролю присваивается соответствующая характеристика его стойкости (Weak, Strong, Verystrong). Результаты представлены в таблице 2

Таблица 2. Классификация стойкости пароля в зависимости от значения

Значение	Стойкость
0-40	Weak
40-85	Strong
85-100	Very strong

Из преимуществ можно отметить то, что разработанный программный модуль аутентификации является расширяемым, т.к. можно добавить свой алгоритм аутентификации

Выводы

В статье проведён анализ существующих САПР с возможностью разграничения прав доступа, представлена структура разрабатываемого программного модуля аутентификации, диаграммы и алгоритмы, которые использовались в ходе реализации программы.

Литература

1. Методы аутентификации [электронный ресурс]. – Режим доступа: <http://www.panasenko.ru/Articles/69/69.html>.
2. SolidWorks-Russia представляет систему SWR-PDM [электронный ресурс]. – Режим доступа: <http://www.sapr.ru/article.aspx?id=7916&iid=320>.
3. Comcon [электронный ресурс]. – Режим доступа: <http://www.sapr.ru/Article.aspx?id=7996>.
4. Компоненты сетевого приложения. Клиент-серверное взаимодействие и роли серверов [электронный ресурс]. – Режим доступа: <http://www.4stud.info/networking/lecture5.html>.
5. MD5 технологий [электронный ресурс]. – Режим доступа: <http://kriptografea.narod.ru/MD5.html>
6. Протоколы идентификации и аутентификации [электронный ресурс]. – Режим доступа: <http://bit.nmu.org.ua/ua/student/metod/cryptology/лекция22.pdf>.
7. Password Strength checker технологий [электронный ресурс]. – Режим доступа: <http://www.dreamincode.net/forums/topic/179364-password-strength-checker/>.
8. How to validate email with regular expression аутентификации [электронный ресурс]. – Режим доступа: <https://www.mkong.com/regular-expressions/how-to-validate-email-address-with-regular-expression/>.

Калинина Е.С., Чернышова А.В. Разработка модуля аутентификации PDM-подсистемы в САД-компонентах САПР с открытым кодом. Предложено описание программного модуля аутентификации с оценкой сложности пароля для существующих САПР с открытым кодом. Исследованы существующие алгоритмы аутентификации и оценки сложности пароля. Программный модуль аутентификации разработан для САПР с открытым кодом, может использоваться для разграничения прав доступа к различного рода информации: группам справочников, отдельным справочникам, классам, объектам и методам расчетов.

Ключевые слова: аутентификация, регистрация, MD5, валидность, оценка сложности пароля

Kalinina E., Chernyshova A. Development of the authentication module PDM-subsystems in CAD-CAD components with open-source code. A description of the software module authentication password complexity assessment for existing CAD systems with open-source code. existing authentication algorithms and evaluation of password complexity. Researched existing authentication algorithms and evaluations of password complexity.

Program module is designed for CAD with the provision of access rights to various kinds of information: directories groups, individual directories, classes, objects and methods of calculation, that will prevent access to undesirable persons and allow entry to legitimate users.

Keywords: authentication, registration, MD5, validity, password complexity rating

Анализ средств JPWL в условиях зашумленности канала передачи данных

С. В. Скороход, О. А. Касьянов
Южный федеральный университет
sss64@mail.ru

Скороход С. В., Касьянов О. А. Анализ средств JPWL в условиях зашумленности канала передачи данных. Рассматривается задача передачи видео в формате jpeg 2000 с применением средств помехоустойчивого кодирования JPWL по зашумленному каналу в условиях возникновения пакетных ошибок. Основная проблема – восстановление потерянного пакета данных средствами JPWL. Поставлена задача проведения экспериментального исследования способности средств JPWL корректировать пакетные ошибки. Методика исследования заключается в моделировании передачи защищенного средствами JPWL видео, состоящего из 1000 кадров одного и того же изображения размером 1024x768 пикселей. Варьируемыми параметрами исследования являются: применяемые для защиты коды Рида-Соломона, количество тайлов изображения, процент потерь RTP (Real-time Transport Protocol) пакетов. Рассматривается стандартный вариант защиты и комбинация стандартного варианта с алгоритмом чередования. Конечными результатами эксперимента выступают среднее значения процента полностью и частично восстановленных тайлов от количества тайлов в кодовом потоке. Описан разработанный для проведения исследования программный комплекс, в который включаются кодер и декодер JPWL, средства разбиения кодового потока на RTP пакеты, средства моделирования пакетных потерь, средства сборки кадров из RTP пакетов. Описаны укрупненная схема функционирования кодера и декодера JPWL. Предложен метод чередования кодового потока, предназначенный для повышения его устойчивости к пакетным ошибкам. В процессе исследования выдвинуты две гипотезы, которые были подтверждены полученными экспериментальными результатами. Гипотеза 1: стандартные средства JPWL не способны корректировать пакетные ошибки. Гипотеза 2: стандартные средства JPWL, применяемые совместно с алгоритмом чередования, могут восстанавливать пакетные ошибки. Побочным результатом исследования является вывод о некотором улучшении восстанавливаемости тайлов с ростом количества тайлов в кодовом потоке.

Ключевые слова: jpeg 2000, JPWL, пакетные ошибки, помехоустойчивое кодирование.

Введение

В работе рассматривается общая проблема передачи видео в формате jpeg 2000 [1] по зашумленному каналу, в частности, по беспроводным сетям. Передача связана с пересылкой сетевых пакетов, несущих фрагменты кодированного видео потока. В результате зашумления канала некоторые пакеты могут быть искажены и отброшены. Следствием таких пакетных ошибок является отсутствие смежных фрагментов видео данных в приемнике, что либо затрудняет их декодирование, либо делает декодирование невозможным. Для защиты кодового потока jpeg 2000 при передаче по беспроводным сетям разработана спецификация ITUT.810 [2]. В ней описан набор средств, применяемых для одиночного изображения, называемых JPWL. Поскольку видео формата jpeg 2000 представляет собой последовательность независимых друг от друга изображений [3], JPWL используется также для защиты видео данных.

Целью данной работы является проведение экспериментальных исследований способности средств JPWL корректировать пакетные ошибки. Под пакетной ошибкой понимается потеря как минимум одного сетевого пакета с данными при передаче от источника к приемнику [4]. Для проведения такого исследования разработан специализированный программный комплекс, который способен выполнять кодирование и декодирование JPWL, разбиение кодового потока JPWL на последовательность сетевых пакетов и обратную сборку кодового потока JPWL из сетевых пакетов с моделированием их потерь. В работе также рассматривается метод чередования, который может существенно повысить способность стандартных средств корректировать пакетные ошибки, что также является предметом экспериментального исследования.

Исследуемая структура кодового потока

Методика исследования опирается на набор средств защиты JPWL, предназначенных для защиты кодового потока видео кадра (Скороход, Артюхова, 2014). При кодировании в формате jpeg 2000 кадр может быть разбит на несколько прямоугольных фрагментов (тайлов), каждый из которых кодируется отдельно. Структура кодового потока изображена на рис. 1 [5].

SOC	Основной заголовок	Заголовок тайла 1	Данные тайла 1	...	Заголовок тайла N	Данные тайла N	EOC
-----	--------------------	-------------------	----------------	-----	-------------------	----------------	-----

Рисунок 1 – Структура кодового потока jpeg 2000

Маркеры SOC и EOC обозначают начало и конец конкретного кадра. Основной заголовок содержит данные об изображении в целом: размер, цветовая схема, количество тайлов и др. Заголовок тайла включает информацию о прямоугольном фрагменте изображения, в том числе о его привязке к координатной сетке. Данные тайла – это собственно закодированный фрагмент изображения.

Средства JPWL могут использовать два способа защиты [2]:

- контрольные суммы CRC-16 или CRC-32 – позволяют только обнаружить присутствие ошибок в данных при использовании минимального объема кодов четности;
- коды Рида-Соломона [6] $RS(n,k)$ (RS-коды) – предназначены не только для обнаружения, но и для исправления $(n-k)/2$ искаженных данных. Здесь n – длина кодового слова в байтах, состоящего из защищаемого фрагмента данных и кодов четности, а k – количество байт защищаемых данных. Код $RS(n,k)$ позволяет исправить до $(n-k)/2$ искаженных байт в кодовом слове. Спецификацией ITUT.810 предусмотрено использование целого семейства RS-кодов от наиболее «слабых» $RS(37,32)$ до наиболее «сильных» $RS(128,32)$ и $RS(160,64)$.

Поскольку контрольные суммы не могут использоваться для защиты от пакетных ошибок, в настоящей работе рассматривается использование только RS-кодов.

Защита кодового потока выполняется добавлением в него специальных сегментов. Предусмотрены четыре вида сегментов.

- EPC – возможность защиты от ошибок. Вставляется только в основной заголовок. Его задача – сообщить декодеру, что в кодовом потоке присутствуют средства JPWL. Содержит описание использованных методов защиты и, при необходимости, их параметры.
- EPB – блок защиты от ошибок. Собственно, реализует защиту данных с использованием RS-кодов. Вставляется в основной заголовок и заголовки тайлов. Содержит параметры применяемой защиты и избыточные коды четности.
- ESD – дескриптор чувствительности. Задаёт степень чувствительности к ошибкам участков кодового потока. Чем больший вклад в формируемое изображение вносит фрагмент кодового потока, тем больше его чувствительность.
- RED – дескриптор остаточной ошибки. Используется только в заголовках декодированного кодового потока. Описывает участки, которые либо полностью восстановлены, либо не были восстановлены декодером JPWL и содержат остаточные ошибки.

Сегмент ESD используется только кодером JPWL с целью реализации более надежной защиты наиболее ценных участков кодового потока. Для декодера JPWL он не несет никакой полезной информации, существенно увеличивая при этом размеры заголовков и всего кодового потока в целом. В связи с этим при передаче видео целесообразно отказаться от использования сегмента ESD [7] и в настоящем исследовании этот сегмент не используется.

Метод и программа исследования

Методика исследования заключается в моделировании передачи защищенного средствами JPWL видео, состоящего из 1000 кадров одного и того же изображения размером 1024x768 пикселей, предварительно закодированного кодером jpeg 2000 системы OpenJpeg 2.1.0 [8]. Полученный кодовый поток загружается в разработанную автором исследовательскую программную систему, в которой он последовательно проходит следующие этапы обработки:

- кодирование JPWL;
- разбиение на RTP-пакеты, соответствующие спецификации RFC 5371 [9, 10];
- моделирование потерь пакетов по заданному проценту потерь;
- сборка кодового потока из оставшихся пакетов;
- декодирование JPWL и подсчет статистики декодирования.

Неизменными параметрами в исследовании являются способ защиты заголовков и размер RTP пакета. Для защиты основного заголовка и заголовков тайлов применяются *RS*-коды, предопределенные в спецификации T.810. Размер RTP пакета равен 1024 байт.

Переменными параметрами исследования являются виды *RS*-кодов, применяемых для защиты данных, процент потери пакетов и количество тайлов, на которые разбивается изображение (определяется выбранным размером тайла). Программа исследования приведена в табл. 1.

Таблица 1. Программа исследования

Количество тайлов (размер тайла)	<i>RS</i> -коды для защиты данных	Процент потерь RTP пакетов
12 (256x256) 48 (128x128) 192 (64x64)	<i>RS</i> (37,32), <i>RS</i> (64,32), <i>RS</i> (96,32), <i>RS</i> (128,32)	1, 3, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50

Статистическими результатами декодирования выступают количество полностью и частично восстановленных тайлов кодового потока.

Следует отметить, что наиболее важной частью кодового потока является основной заголовок. Поэтому для его дополнительной защиты обычно применяются специальные меры, такие как избыточное дублирование пакетов, содержащих его фрагменты, в связи с чем, в данном исследовании пакетные ошибки не будут применяться к пакетам основного заголовка.

Структура и функционирование программного комплекса

Для данного исследования был разработан специализированный программный комплекс, состоящий из двух модулей: подготовительного и моделирующего. Схема функционирования комплекса изображена на рис. 2.

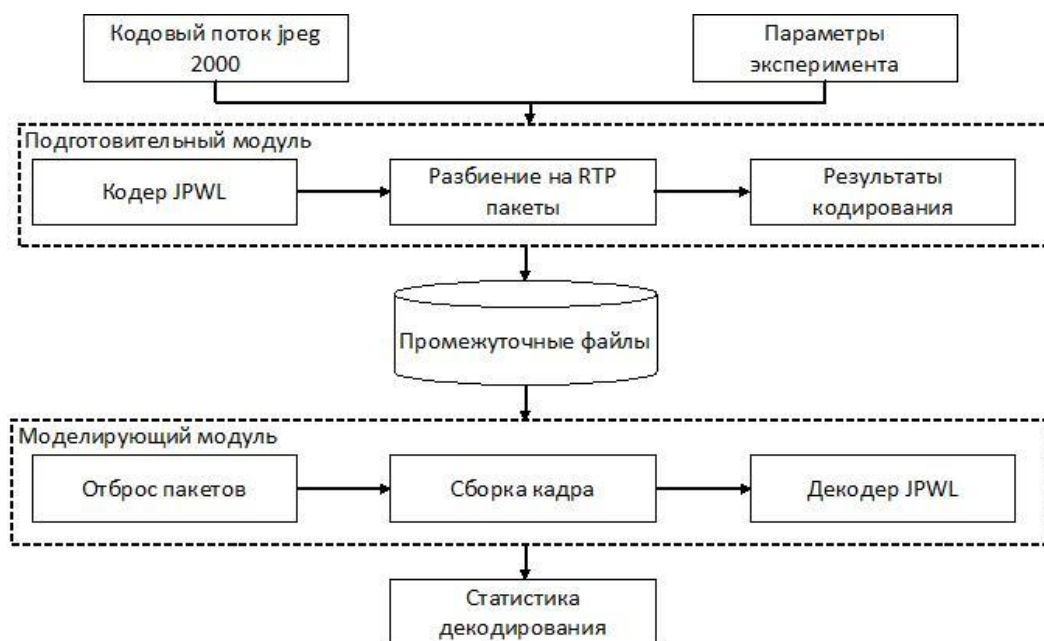


Рисунок 2 – Исследовательский программный комплекс

Исходными данными для подготовительного модуля является кодовый поток jpeg 2000, предварительно создаваемый при помощи OpenJpeg 2.1.0, содержащий 12, 48 или 192 тайла в соответствии с программой исследования, а также набор параметров исследования, в которых задается режим кодирования (с использованием чередования или без) и процент отбрасываемых пакетов. Подготовительный модуль выполняет кодирование JPWL, при необходимости – чередование, разбивает полученный кодовый поток на RTP пакеты в соответствии со спецификацией RFC 5371 и выполняет выгрузку в промежуточные файлы полученную последовательность пакетов, их параметры (количество, длины и т.д.) и параметры исследования.

Моделирующий модуль загружает из промежуточных файлов подготовленную последовательность пакетов и параметры исследования. Далее выполняются циклические итерации отброса пакетов, сборки кадра и его обработки декодером JPWL. Декодер подсчитывает количество полностью и частично восстановленных тайлов. После выполнения 1000 подобных итераций вычисляется среднее арифметическое этих величин,

характеризующее способность декодера восстановить кодовый поток при заданных параметрах кодирования и заданном уровне пакетных ошибок.

Главными звеньями данного комплекса являются кодер и декодер JPWL. В данном исследовании используются библиотеки кодера и декодера, разработанные автором для системы видеотрансляции [11, 12].

Обобщенная схема работы кодера JPWL приведена на рис. 3.

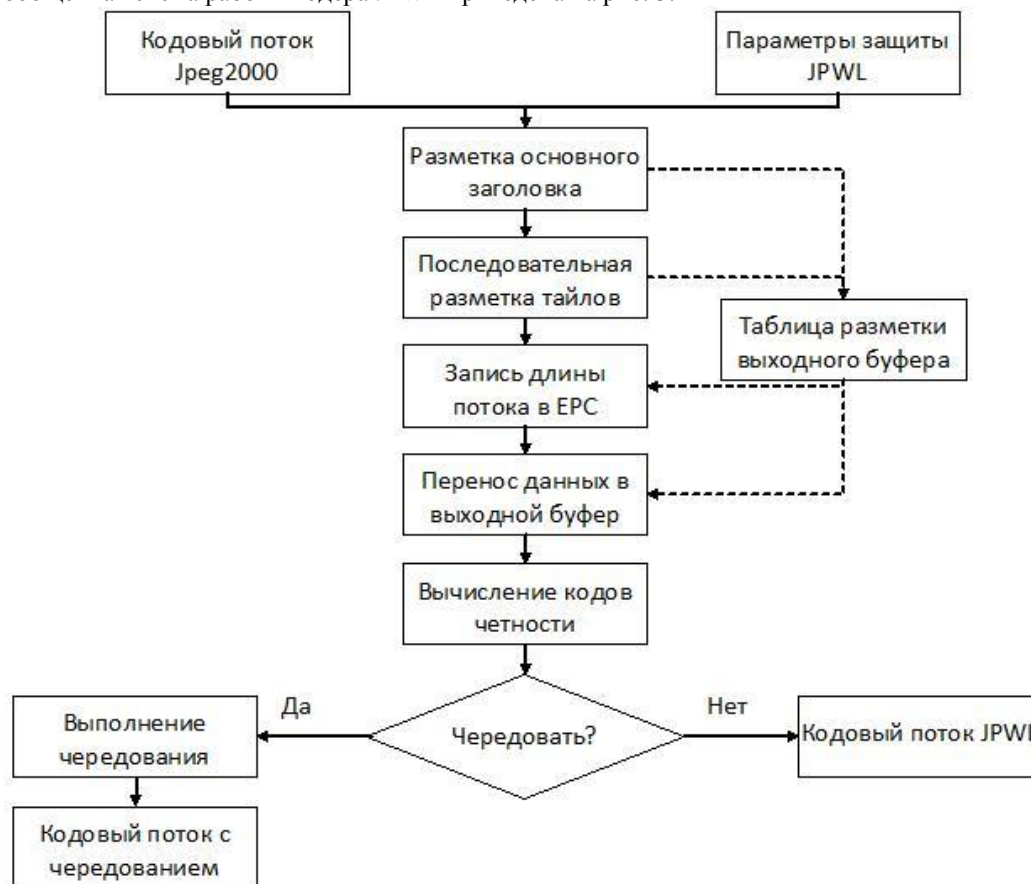


Рисунок 3 – Схема работы кодера JPWL

Входными данными являются собственно кодовый поток jpeg 2000 и параметры используемых средств защиты JPWL. Кодер выполняет разметку основного заголовка и каждого последующего тайла в выходном буфере. При этом данные о параметрах маркеров JPWL и их местоположении заносятся в таблицу разметки выходного буфера. После окончания прохода входного буфера вычисляется длина выходного потока как сумма длины входного потока и длин всех добавляемых в него маркеров. Полученное значение длины помещается в таблицу разметки маркера EPC. Далее выполняется копирование на свои места в выходном потоке исходных данных и параметров маркеров JPWL. Последним шагом кодирования является вычисление и вставка в выходной поток кодов четности маркеров EPB. После этого шага формируется кодовый поток JPWL, удовлетворяющий требованию обратной совместимости, т.е. декодер jpeg 2000 способен декодировать данный поток, если в нем нет искажений. При применении алгоритма чередования дополнительно выполняется чередование, результатом которого является кодовый поток JPWL, удовлетворяющий требованию обратной совместимости с расширением: декодер jpeg 2000 способен декодировать такой поток только при наличии декодера JPWL.

Обобщенная схема работы декодера JPWL изображена на рис. 4. Декодер JPWL выполняет декодирование предварительно собранного кадра, последовательно анализируя основной заголовок и заголовки тайлов. При невозможности восстановления основного заголовка весь кадр отбрасывается. Если основной заголовок успешно восстановлен, то анализ сегмента EPC позволяет определить применяется чередование кодового потока или нет.

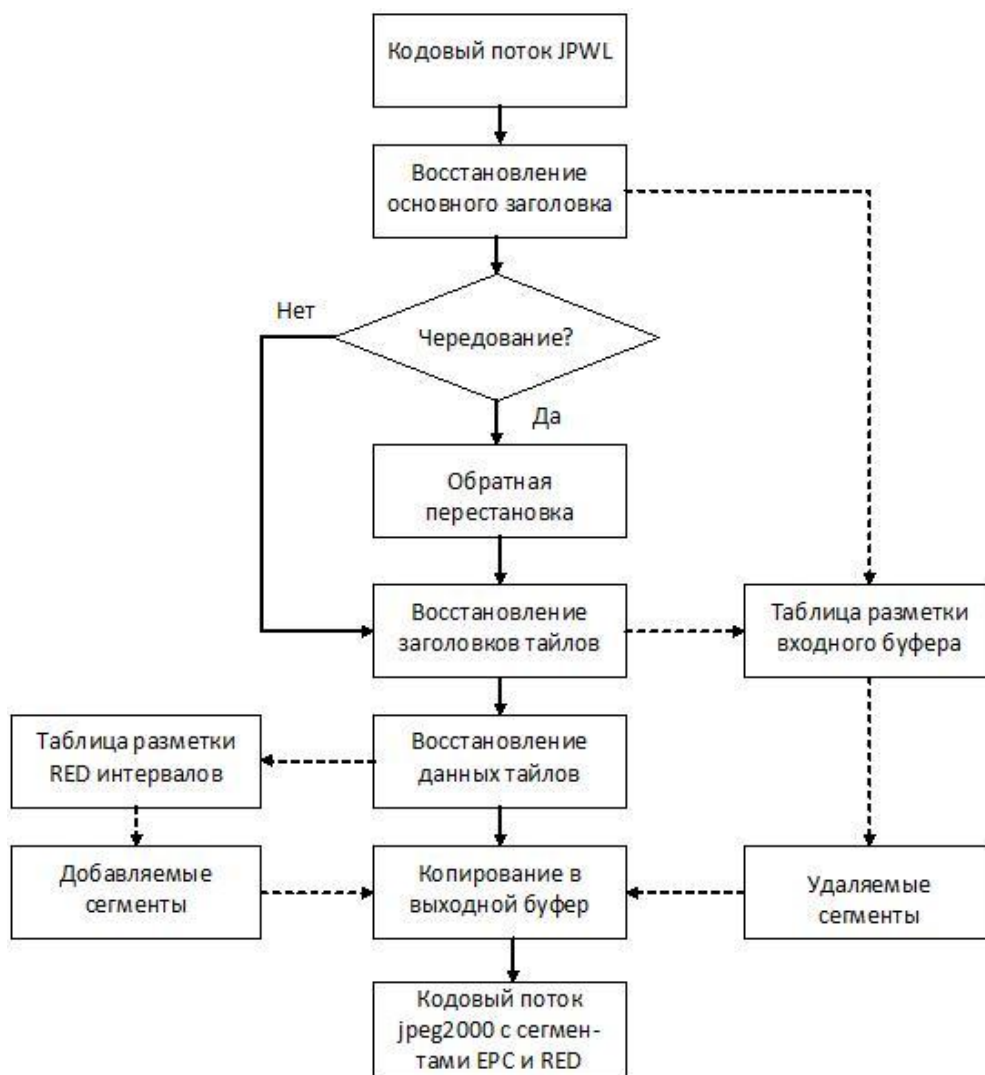


Рисунок 4 – Схема работы декодера JPWL

Если чередование применяется, декодер сначала выполняет обратную перестановку, и только после этого переходит к анализу тайлов. При невозможности восстановления заголовка тайла пропускается весь тайл и ищется заголовок следующего тайла. При этом пропущенный фрагмент помечается в таблице разметки входного буфера как испорченный. В случае успешного восстановления основного заголовка или заголовка тайла в таблицу разметки заносятся параметры и местоположение всех найденных маркеров JPWL. Восстановление данных тайла выполняется только в случае успешного восстановления его заголовка. При этом часть данных тайла может оказаться скорректированной, а часть – нет. В этом случае создаются записи в таблице RED интервалов, которые описывают начало и конец как исправленных, так и не исправленных фрагментов данных.

По завершении просмотра входного буфера выполняется копирование данных в выходной буфер. При этом выполняется пропуск всех входных маркеров JPWL. При наличии записей в таблице RED интервалов в основной заголовок вставляется сегмент EPC, сигнализирующий о присутствии в кодовом потоке сегментов RED, а в заголовки частично восстановленных тайлов добавляются сегменты RED, описывающие адреса восстановленных и не восстановленных участков.

Подсчет статистики декодера JPWL выполняется на основе результатов восстановления заголовков и данных тайлов. Если после декодирования заголовок тайла не содержит сегмента RED, такой тайл считается полностью восстановленным. Если сегмент RED присутствует – тайл восстановлен частично. Наличие сегмента RED ничего не говорит о качестве данных в частично восстановленном тайле, поэтому в данном исследовании этот вопрос не затрагивается.

Метод чередования

Кодер и декодер JPWL поддерживают как стандартный способ защиты, соответствующий спецификации T.810, так и алгоритм чередования, не относящийся к этой спецификации [13]. Сущность алгоритма заключается

в том, что основной заголовок остается без изменений, а остальная часть кодового потока подвергается перестановке. Основной заголовок содержит сегмент EPC, в котором записана общая длина кодового потока. Разность общей длины и длины основного заголовка определяет длину данных, которые должны быть переставлены. Эти данные рассматриваются как прямоугольная матрица, заполненная «по строкам». Данные в этой матрице переставляются так, чтобы она оказалась заполнена «по столбцам».

Пусть L – общая длина кодового потока, M – длина основного заголовка. Тогда согласно [14] количество столбцов и строк матрицы вычисляется по формулам:

$$N_C = \left\lceil \sqrt{L-M} \right\rceil, \quad N_R = \left\lceil \frac{L-M}{N_C} \right\rceil.$$

Следует отметить, что в результате такой перестановки фактическая длина кодового потока несколько увеличивается на стороне кодера, поскольку в матрицу может быть включено до N_R-1 «лишних» элементов. Кодер и декодер JPWL должны учитывать этот факт.

Отличие предлагаемого способа чередования от описанного в [14] заключается в том, что в данном случае чередованию подвергаются весь кодовый поток за исключением основного заголовка. В [14] предлагается перестановка только кодов четности, записанных в сегментах EPB.

Экспериментальное исследование

В исследовании проверяются две основные гипотезы.

Гипотеза 1. Стандартные средства JPWL, соответствующие спецификации T.810, не способны корректировать пакетные ошибки. Критерием, подтверждающим эту гипотезу, является обратная линейная зависимость процента пакетных потерь и суммарного количества полностью и частично восстановленных тайлов.

Проанализируем стандартный способ защиты с применением RS-кодов. Кодовое слово $RS(n,k)$ состоит из двух частей расположенных в разных частях кодового потока. Первую часть составляют избыточные коды четности длиной $n-k$ байт, которые расположены в основном заголовке или заголовках тайлов. Вторая часть – k информационных байт, защищаемых от искажения. При потере пакета возможны три варианта [13].

1. Обе части кодового слова потеряны – восстановление невозможно.
2. Потеряны избыточные коды четности – восстановление невозможно.
3. Потеряны только информационные байты – восстановление возможно только если $k \leq (n-k)/2$. Этому условию соответствуют только коды $RS(160,64)$, $RS(80,25)$, $RS(40,13)$, применяемые для защиты заголовков тайлов, и $RS(96,32)$, $RS(112,32)$, $RS(128,32)$, применяемые для защиты данных тайлов. Остальные 14 предопределенных стандартом кодов не способны произвести восстановление.

Анализируемые в исследовании способы защиты кодами $RS(37,32)$, $RS(64,32)$ не способны скорректировать данные ни в одном из перечисленных вариантов. При применении $RS(96,32)$ и $RS(128,32)$ возможна коррекция данных в варианте 3, однако расплатой за это является значительное увеличение заголовков тайлов, что делает из гораздо уязвимее и может нивелировать возможный положительный эффект.

Гипотеза 2. Стандартные средства JPWL, применяемые совместно с алгоритмом чередования, могут восстанавливать пакетные ошибки. Способность восстановления зависит от используемых RS-кодов и процента пакетных потерь. Критерием, подтверждающим эту гипотезу, является нелинейная зависимость процента пакетных потерь и суммарного количества полностью и частично восстановленных тайлов, при которой до определенного уровня потерь наблюдается стопроцентное восстановление тайлов.

Для проверки гипотезы 1 проведем эксперимент 1. Эксперимент проводится над кодовым потоком jpeg 2000, содержащем 48 тайлов, к которому применяется стандартная защита. Применяемые RS-коды и моделируемые проценты потерь RTP пакетов заданы в программе исследований. Результаты эксперимента приведены в табл/ 2 с округлением до целых величин. Для каждого вида RS-кодов указан размер кодового потока JPWL, средний процент полностью и частично восстановленных тайлов.

Данный результат полностью подтверждает гипотезу 1. Более того, надежды на то, что RS-коды $RS(96,32)$ и $RS(128,32)$ дадут лучший результат по сравнению с $RS(37,32)$ и $RS(64,32)$ не оправдались по причине чрезмерного увеличения заголовков тайлов при применении этих кодов.

В рамках проверки гипотезы 1 проводились также эксперименты 2 и 3, в которых использовался кодовый поток jpeg 2000, состоящий соответственно из 12 и 192 тайлов. Остальные условия соответствовали эксперименту 1. Результаты экспериментов 2 и 3 приведены в табл. 3 и 4.

Таблица 2. Результаты эксперимента 1

Потери пакетов %	RS(37,32) 1025691 байт		RS(64,32) 1767653 байт		RS(96,32) 2647205 байт		RS(128,32) 3526937 байт	
	Полн.	Част.	Полн.	Част.	Полн.	Част.	Полн.	Част.
1	81	18	71	28	69	30	58	40
3	53	44	34	63	35	61	21	76
5	35	60	18	77	16	79	8	85
10	12	78	4	86	4	86	1	87
15	5	81	1	84	1	84	0	81
20	2	78	0	81	0	80	0	76
25	1	75	0	76	0	75	0	71
30	0	69	0	70	0	70	0	65
35	0	64	0	65	0	64	0	60
40	0	59	0	59	0	61	0	54
45	0	55	0	55	0	56	0	50
50	0	50	0	49	0	51	0	43

Таблица 3. Результаты эксперимента 2

Потери пакетов %	RS(37,32) 1003010 байт		RS(64,32) 1733367 байт		RS(96,32) 2599119 байт		RS(128,32) 3464911 байт	
	Полн.	Част.	Полн.	Част.	Полн.	Част.	Полн.	Част.
1	45	54	23	76	25	72	13	83
3	10	87	3	92	2	91	1	87
5	2	91	1	92	0	88	0	82
10	0	90	0	83	0	75	0	68
15	0	87	0	77	0	63	0	54
20	0	79	0	71	0	55	0	45
25	0	76	0	62	0	49	0	36
30	0	70	0	55	0	37	0	27
35	0	63	0	49	0	34	0	19
40	0	60	0	42	0	25	0	10
45	0	55	0	39	0	20	0	3
50	0	51	0	32	0	17	0	0

Таблица 4. Результаты эксперимента 3

Потери пакетов %	RS(37,32) 1003010 байт		RS(64,32) 1733367 байт		RS(96,32) 2647205 байт		RS(128,32) 3526937 байт	
	Полн.	Част.	Полн.	Част.	Полн.	Част.	Полн.	Част.
1	94	5	90	9	90	9	86	13
3	83	14	73	24	74	23	63	34
5	73	22	60	35	60	35	47	48
10	52	38	36	54	36	54	23	67
15	38	47	21	64	22	64	11	74
20	27	53	13	68	13	68	5	75
25	19	56	7	68	7	68	3	72
30	13	57	4	66	5	66	1	68
35	9	56	3	63	2	63	1	64
40	6	53	2	58	1	58	0	60
45	4	51	1	54	1	54	0	55
50	3	47	1	49	0	50	0	50

Полученные зависимости полностью соответствуют критерию, подтверждающему гипотезу 1 во всех экспериментах. Кроме того, можно сделать вывод о том, что с увеличением количества тайлов увеличивается процент их восстановления. Это объясняется тем, что чем больше тайлов, тем меньше их размер, что уменьшает вероятность повреждения конкретного тайла при пакетной ошибке. Наоборот, чем меньше тайлов, тем больше их размер и более уязвимыми для пакетной ошибки они становятся.

Для проверки гипотезы 2 проведем эксперимент 4. Исходные данные для него соответствуют эксперименту 1, но стандартный способ защиты комбинируется с алгоритмом чередования. Результаты

эксперимента приведены в табл. 5.

Таблица 5. Результаты эксперимента 4

Потери пакетов %	RS(37,32) 1027949 байт		RS(64,32) 1770615 байт		RS(96,32) 2649924 байт		RS(128,32) 3529845 байт	
	Полн.	Част.	Полн.	Част.	Полн.	Част.	Полн.	Част.
1	39	61	100	0	100	0	100	0
3	0	100	100	0	100	0	100	0
5	0	100	100	0	100	0	100	0
10	0	100	98	2	100	0	100	0
15	0	100	28	72	100	0	100	0
20	0	100	0	100	85	15	100	0
25	0	98	0	99	7	91	83	15
30	0	92	0	92	0	92	7	84
35	0	72	0	73	0	72	0	70
40	0	37	0	39	0	39	0	35
45	0	11	0	11	0	11	0	9
50	0	1	0	1	0	1	0	1

Анализ табл. 5 подтверждает гипотезу 2. Действительно, при уровне пакетных потерь до 25% любые RS-коды восстанавливают практически все тайлы, изменяется лишь распределение полностью и частично восстановленных тайлов. С дальнейшим ростом пакетных потерь процент восстановленных тайлов начинает снижаться. В эксперименте 4 проявилась еще одна явная зависимость: при применении более «длинных» RS-кодов качество восстановления улучшается, прежде всего за счет большего процента полностью восстановленных тайлов.

На рис. 5 изображен график зависимости суммарного процента полностью и частично восстановленных тайлов (ордината) от процента пакетных ошибок (абсцисса) для эксперимента 4. Полученные зависимости полностью соответствуют критерию, подтверждающему гипотезу 2.

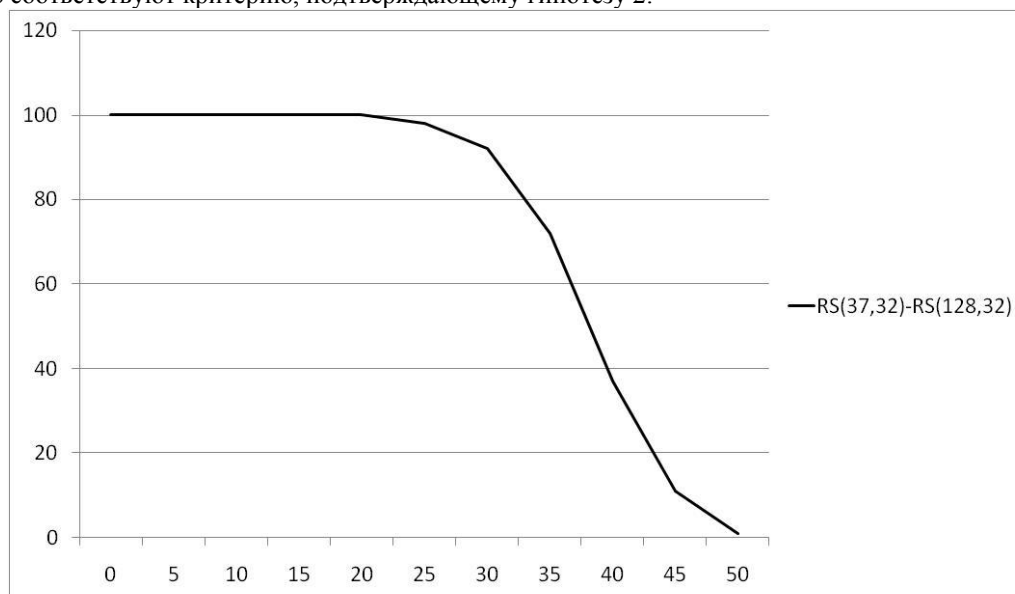


Рисунок 5 – График зависимости для эксперимента 4

В рамках данного исследования были проведены эксперименты 5 и 6, условия которых совпадали с условиями эксперимента 4, но был использован кодовый поток jpeg 2000, содержащий соответственно 12 и 192 тайла. Результаты этих экспериментов во многом совпадают с таблицей 5 и рисунком 4 с некоторым ухудшением при 12 тайлах и улучшением при 192.

Заключение

Данное исследование проведено на программном комплексе, разработанном на языке Си в системе программирования Microsoft Visual Studio 2008. Полученные результаты могут быть использованы для выбора оптимальных параметров JPWL в зависимости от характеристик канала передачи данных.

По результатам проведенных экспериментов можно сделать следующие выводы.

1. Набор стандартных средств JPWL, соответствующих спецификации T.810 является недостаточным для эффективной коррекции пакетных ошибок.
2. Сочетание стандартных средств JPWL с предложенным в работе алгоритмом чередования позволяет успешно корректировать пакетные ошибки.
3. Увеличение количества тайлов незначительно повышает устойчивость кодового потока к пакетным ошибкам.

Направлением дальнейших исследований является экспериментальное исследование влияния пакетных ошибок на качество декодированного изображения в формате jpeg 2000. Для проведения этого исследования требуется разработать декодер jpeg 2000, способный декодировать частично восстановленный кодовый поток, в котором могут отсутствовать некоторые тайлы, а в заголовках, имеющихся тайлов присутствовать сегменты остаточной ошибки RED. К сожалению, известные доступные декодеры, такие как JasPer, OpenJpeg не могут декодировать такой кодовый поток.

Работа выполнена в рамках выполнения базовой части государственного задания в сфере научной деятельности (проект № 3442 "Информационно-алгоритмическое обеспечение систем цифрового управления, автономной высокоточной навигации и технического зрения для перспективных летательных аппаратов: разработка теоретических основ проектирования, алгоритмов, способов эффективной и надежной программной реализации, использование высокопроизводительной вычислительной инфраструктуры для экспериментального моделирования").

Литература

1. Taubman, D.S., Marcellin M.W. JPEG2000: Image Compression Fundamentals, Standards and Practice. – Boston: Kluwer Academic Publishers, 2002. – 773 p.
2. ITU-T Recommendation T.810. Information Technology JPEG2000 Image Coding System: Wireless. – Geneva: ITU, 2007. – 60 p.
3. Acharya T., Tsai P. JPEG 2000 standard for image compression: concepts, algorithms and VLSI architectures.– Hoboken : John Wiley & Sons, Inc., 2005. – 274 p.
4. Скороход С. В., Степанова А. П. Некоторые аспекты технологии сетевого взаимодействия между передатчиком и приемником в процессе видео трансляции // Известия ЮФУ. Технические науки.– №4 (165), 2015.– с. 171–181.
5. ITU-T Recommendation T.800. Information Technology JPEG2000 Image Coding System: Core Coding System. – Geneva: ITU-T Series T: Terminal for Telematic Services, 2003. – 212 p.
6. Морелос-Сагагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение.– М: Изд-во Техносфера, 2005.– 320 с.
7. Skorokhod S.V., Kravchenko P.P., Khusainov N.S. Some aspects of JPWL implementation for streaming video // International Conference «Engineering & Telecommunication En&T 2014». November 26-28, 2014. Book of Abstracts. – Moscow–Dolgoprudny : MIPT, 2014. p. 135-136.
8. An open-source JPEG 2000 codec written in C. Retrieved April 6, 2016, from <http://www.openjpeg.org/>.
9. Network Working Group RFC 5371. RTP Payload Format for JPEG 2000 Video Streams. – The Internet Security (IETF), 2006. – 31 p.
10. Network Working Group RFC 3550. RTP: A Transport Protocol for Real-Time Applications.— Schulzrinne: BlueCoatSystemsInc., 2003.– 89 p.
11. Дроздов С.Н., Жиглатый А.А., Кравченко П.П., Скороход С.В., Хусаинов Н.Ш. Об опыте реализации системы видеотрансляции в формате jpeg2000 и перспективах применения стандарта jpeg2000 для передачи видео и мультиспектральных данных с борта бпла// Известия ЮФУ. Технические науки.– 2014, №7 (156).– с. 161–170.
12. Дроздов С.Н., Жиглатый А. Н., Кравченко П. П., Лутай В. Н., Скороход С. В., Хусаинов Н. Ш. Стандарт JPEG 2000: базовые алгоритмы, примеры реализации и перспективы применения.– Ростов-на-Дону, Изд-во ЮФУ, 2014.– 255 с.
13. Skorokhod S. V. The experimental studies of the jpwl tools ability to correct burst errors in a noisy channel when transmitting video in a jpeg 2000 format // Journal of Theoretical and Applied Information Technology, 10th November, Vol. 81, № 1, 2015. – p. 34-42.
14. ITU-T Recommendation T.810. Information Technology JPEG 2000 image coding system: Wireless. Amendment 1: IP based wireless networks.— Geneva: ITU-T Series T: Terminal for Telematic Services, 2012.–6 p.

Скороход С. В., Касьянов О. А. Анализ средств JPWL в условиях зашумленности канала передачи данных. Рассматривается задача передачи видео в формате jpeg 2000 с применением средств помехоустойчивого кодирования JPWL по зашумленному каналу в условиях возникновения пакетных ошибок. Основная проблема – восстановление потерянного пакета данных средствами JPWL. Поставлена задача проведения экспериментального исследования

способности средств JPWL корректировать пакетные ошибки. Методика исследования заключается в моделировании передачи защищенного средствами JPWL видео, состоящего из 1000 кадров одного и того же изображения размером 1024x768 пикселей. Варьируемыми параметрами исследования являются: применяемые для защиты коды Рида-Соломона, количество тайлов изображения, процент потерь RTP (Real-time Transport Protocol) пакетов. Рассматривается стандартный вариант защиты и комбинация стандартного варианта с алгоритмом чередования. Конечными результатами эксперимента выступают средние значения процента полностью и частично восстановленных тайлов от количества тайлов в кодовом потоке. Описан разработанный для проведения исследования программный комплекс, в который включаются кодер и декодер JPWL, средства разбиения кодового потока на RTP пакеты, средства моделирования пакетных потерь, средства сборки кадров из RTP пакетов. Описаны укрупненная схема функционирования кодера и декодера JPWL. Предложен метод чередования кодового потока, предназначенный для повышения его устойчивости к пакетным ошибкам. В процессе исследования выдвинуты две гипотезы, которые были подтверждены полученными экспериментальными результатами. Гипотеза 1: стандартные средства JPWL не способны корректировать пакетные ошибки. Гипотеза 2: стандартные средства JPWL, применяемые совместно с алгоритмом чередования, могут восстанавливать пакетные ошибки. Побочным результатом исследования является вывод о некотором улучшении восстанавливаемости тайлов с ростом количества тайлов в кодовом потоке.

Ключевые слова: jpeg 2000, JPWL, пакетные ошибки, помехоустойчивое кодирование.

Skorokhod S. V., Kasyanov O. A. The experimental studies of the jpwl tools ability to correct burst errors in a noisy channel when transmitting video in a jpeg 2000 format. The discussion is made on the issue of transmitting video in a JPEG 2000 format using the JPWL tools (Wireless JPEG 2000) for noise-immunity coding through a noisy channel under conditions of burst errors occurrence. It is set out to conduct an experimental study on the ability of JPWL tools to correct burst errors. The methodology of the study consists in modelling a transmission of the JPWL protected video composed of 1,000 frames of one and the same image with a size of 1,024x768 pixels. The variable parameters of the study are as follows: the Reed-Solomon codes applied for protection, the number of image tiles, and the RTP (Real-time Transport Protocol) packet loss ratio. A standard variant of protection and a combination of the standard variant with an interleaving algorithm are considered. The final results of the experiment are the average values of the percentage of fully and partially restored tiles of the number of tiles in a code stream. A software system developed for the conduction of study is described, which includes the JPWL encoder and decoder, the tools for partitioning code streams into RTP packets, the tools for packet loss modelling, and the tools for frame assembly from the RTP packets. A macro flowchart of the JPWL encoder and decoder functioning is described. A method for the code stream interleaving is suggested, which is intended for increasing its resistance to burst errors. During the study process, the two hypotheses were put forward, which were confirmed by the experimental results obtained. The first hypothesis is that the standard JPWL tools are not able to correct burst errors. The second hypothesis is that the standard JPWL tools used in conjunction with the interleaving algorithm can recover burst errors. A side effect of the study is the conclusion on a slight improvement of the recoverability of tiles with the increasing number of tiles in a code stream.

Keywords: JPEG 2000, JPWL, burst errors, noise-immunity coding.

СЕКЦИЯ «КОМПЬЮТЕРНАЯ ГРАФИКА: СИНТЕЗ И ОБРАБОТКА ИЗОБРАЖЕНИЙ»

УДК 004.932.72'1

Распознавание окружностей на изображении с помощью алгоритма ВФОА

О.В. Минков, К.А. Ручкин
Донецкий национальный технический университет
oleg.minkov93@gmail.com

Минков О.В., Ручкин К.А. Распознавание окружностей на изображении с помощью алгоритма ВФОА. Выполнен обзор методов, которые могут быть применены для задачи распознавания многоугольников. Рассмотрена работа алгоритма ВФОА на примере распознавания окружности. Рассмотрена модификация алгоритма ВФОА для распознавания многоугольников.

Ключевые слова: алгоритм, распознавание, ВФОА, изображение, окружность

Введение

Одной из целей построения онтологической модели бионических технологий является создание основы для разработки базы знаний интеллектуальной системы информационной поддержки процессов создания и развития перспектив бионических технологий. Указанная система должна обеспечивать накопление, организацию и использование информационных ресурсов в области бионики.

К настоящему времени разработано несколько алгоритмов оптимизации, объединяемых общим названием роевая оптимизация. Одним из таких алгоритмов является оптимизация передвижения бактерий.

Цель работы

Цель работы – рассмотреть применение эвристических алгоритмов (в частности, алгоритма ВФОА) в задаче распознавания многоугольников на изображении.

Впервые алгоритм ВФОА, известный как алгоритм оптимизации, основанный на передвижении бактерий, был предложен в 2002 году [1]. С алгоритмической точки зрения, он представляет собой многоагентную систему, которая имитирует отдельное и сгруппированное оптимальное поведение бактерий, живущих в кишечнике большинства млекопитающих. С помощью этого алгоритма существует возможность обнаруживать формы, если адаптировать его для задачи распознавания различных форм [2].

Для задачи распознавания многоугольников выбран алгоритм ВФОА, так как для нахождения им различных фигур достаточно заменить функцию определения оптимальности особи и в зависимости от фигуры будет изменено пространство поиска.

Биологические основы

Одной из наиболее изученных бактерий является бактерия E.Coli. Данная бактерия живёт в кишечнике большинства млекопитающих (в том числе и в кишечнике человека). Бактерия E.Coli при соответствующих условиях может самовоспроизводиться (расщепляться) за 20мин. Способность двигаться бактерии E.Coli обеспечивается за счёт шести или более жгутиков, которые вращаются с частотой 100–200 оборотов в секунду, каждый из которых управляется с помощью своего собственного биологического мотора. Бактерия E.Coli чередует свои действия: то перемещается прямолинейно (со скоростью 10–20 мкм/сек.), то кувьркается (изменение направления). Когда жгутики вращаются по часовой стрелке (или против часовой стрелки), они работают как пропеллеры и, следовательно, E.Coli может плыть или кувьркаться.

Хемотаксис – это двигательная реакция бактерии в ответ на появление в среде аттрактанта (аттрактант – вещество, привлекающее бактерии) или репеллента (репеллент – вещество, отпугивающее бактерий).

Понятно, что в естественных условиях аттрактантами являются вещества, полезные для бактерий, а репеллентами – те, которые бактериям вредны.

Примерная схема движения бактерии выглядит следующим образом: когда мотор работает в одну сторону, все жгутики у бактерии складываются и крутятся вместе, и бактерия движется прямолинейно. В конце пробега бактерия останавливается, мотор переключается и начинает работать в другую сторону. Жгутики растопыриваются и "бултыхаются" независимо друг от друга. Бактерия при этом переориентируется в пространстве случайным образом. После этого, мотор опять переключается и начинает работать в ту сторону, в которую жгутики работают вместе, и возникает следующий отрезок прямолинейного движения.

При наличии пространственных изменений концентрации аттрактантов или репеллентов частота кувырканий, а, следовательно, и длина свободного пробега бактерии изменяются. Длина свободного пробега бактерии, плывущей в сторону возрастающей концентрации аттрактанта, увеличивается, а при движении в сторону возрастающей концентрации репеллента уменьшается.

Таким образом, можно выделить следующие хемотаксические действия бактерии E.Coli [3]:

– если бактерия находится в нейтральной среде, то чередуются кувырки и передвижения, за счёт чего выполняется поиск;

– если бактерия перемещается по градиенту аттрактанта, то перемещение продолжается в этом же направлении. Таким образом, обеспечивается поиск более благоприятной окружающей среды.

– если происходит перемещение в направлении противоположном градиенту репеллента, то обеспечивается избегание неблагоприятной окружающей среды.

Таким образом, бактерия может перемещаться по полезным веществам и в то же время избегать опасных веществ. Датчики, используемые E.Coli – белковые рецепторы, которые являются очень чувствительными (незначительное изменение в концентрации полезных веществ может вызвать существенное изменение в поведении бактерии). От рецепторов сигналы поступают на метилакцептирующие белки, которые собирают все сигналы от рецепторов, и результирующий сигнал выходит на мотор жгутика, который управляет движением бактерии в зависимости от соотношения полезных и опасных веществ в окружающей среде.

Бактерии часто умирают или растворяются, и это должно учитываться при моделировании их деятельности. Мутации у E. coli происходят в нормальных условиях приблизительно 10^{-7} в ген и в поколение.

E.Coli могут формировать сложные устойчивые пространственновременные структуры в некоторых полутвердых полезных средах. Бактерии могут поглощать питательные вещества по их радиусу, начиная от внешней границы, заканчивая серединой, даже в случае если бактерии были изначально размещены в центре питательных веществ. Кроме того, при определённых условиях, они могут скрывать притягивающие сигналы от клетки к клетке, за счёт чего они группируются и защищают друг друга. Таким образом, эти бактерии могут группироваться в колонии.

Метод оптимизации на основе моделирования перемещения бактерий (Bacterial Foraging Optimization, BFO)

Данный метод предназначен для нахождения минимума функции $J(X)$, $X \in R^P$ при неизвестном градиенте $\nabla J(X)$, где X – позиция бактерии в пространстве поиска R^P , а с помощью $J(\theta)$ моделируются полезные и вредные свойства среды, т.е. $J(X)$ характеризует, где находятся аттрактанты и репелленты. Таким образом, $J < 0$, $J = 0$, $J > 0$ означает, что бактерия находится в полезной, нейтральной и вредной среде, соответственно.

Пусть $P(j, k, l) = \{ X_i(j, k, l) \}$, $i=1, 2, \dots, S$ описывает позицию каждого члена популяции S бактерий на j -ом хемотаксическом шаге, k -ом шаге воспроизведения и на l -ом событии исключения- рассеивания.

Тогда работу метода оптимизации с помощью моделирования фуражировки бактерий можно представить в виде последовательности выполнения следующих шагов [4].

Шаг 1. Инициализация. Задаются параметры, влияющие на работу метода: S – количество бактерий, N_{re} – количество шагов воспроизведения, N_s – количество шагов-повторений на одном хемотаксическом шаге, N_c – количество хемотаксических шагов, N_{ed} – количество событий исключения-рассеивания, P_{ed} – вероятность рассеивания. Случайным образом распределить начальные значения X_i , $i = 1, 2, \dots, S$ по пространству поиска. Рассчитываются начальные значения целевой функции для каждой бактерии J_i .

Шаг 2. Установить: $l = l + 1$.

Шаг 3. Установить: $k = k + 1$.

Шаг 4. Установить: $j = j + 1$.

Шаг 5. Для каждой бактерии моделируется хемотаксис: кувыркание, перемещение и скольжение.

Шаг 5.1. Установить: $i = i + 1$.

Шаг 5.2. Кувыркание. Моделирование кувыркания достигается за счёт генерации вектора случайных чисел $\phi(j) \in R^P$:

$$\phi = \frac{\Delta}{\sqrt{\Delta^T \Delta}}$$

где Δ – вектор случайных чисел в интервале $[-1; 1]$.

Вектор φ представляет собой множество длин для соответствующих измерений.

Шаг 5.3. Перемещение. Рассчитывается новое положение i -ой бактерии по формуле:

$$X^i(j+1, k, l) = X^i(j, k, l) + C(i)\varphi(j),$$

где $C(i) > 0$ – размер шага в определённом направлении, позволяющий моделировать процесс кувыркания.

Для новой позиции $X^i(j+1, k, l)$ рассчитывается соответствующее значение целевой функции $J(i, j+1, k, l)$.

Шаг 5.4. Скольжение. Если в позиции $X^i(j+1, k, l)$ значение $J(i, j+1, k, l)$ лучше, чем в позиции $X^i(j, k, l)$, то есть выполняется условие: $J(i, j+1, k, l) < J(i, j, k, l)$, тогда производится следующий хемотаксический шаг с тем же вектором φ и в том же направлении (переход к шагу 5.3), и такое повторение может повторяться N_s раз. Если условие не выполняется, то переход к шагу 5.5.

Шаг 5.5 Если $i < S$, то переход к шагу 5.1, в противном случае – переход к шагу 6.

Шаг 6. Если $j < N_s$, то переход к шагу 4, в противном случае – переход к шагу 7.

Шаг 7. Воспроизведение. Менее здоровые бактерии умирают, а остальные, более здоровые, разделяются на две бактерии, при этом новые бактерии размещаются в ту же самую точку пространства поиска. За счёт такого подхода обеспечивается неизменность общего количества бактерий. Для этого все агенты сортируются в соответствии с полученными значениями целевой функции, после чего худшая половина (менее здоровые бактерии) отбрасывается, а лучшая (более здоровые бактерии) – дублируется.

Шаг 8. Если $k < N_{re}$, тогда выполняется переход к шагу 3, в противном случае – переход к шагу 9.

Шаг 9. Исключение и рассеивание. Жизнь популяции бактерий в окружающей среде может меняться либо постепенно (например, путём потребления полезных веществ), либо внезапно в связи с некоторым другим воздействием. Может произойти так, что все бактерии в области погибнут, или колония бактерий будет рассеяна в другую часть окружающей среды. Данный эффект может помешать возможному хемотаксическому прогрессу, но в то же время этот эффект и помогает, поскольку, в случае рассеивания, бактерии могут разместиться около хороших источников с полезными веществами. Исключение и рассеивание помогают понизить вероятность стагнации, то есть заикливания в локальном оптимуме, что часто наблюдается в традиционных методах оптимизации (например, метод Коши, Ньютона, Левенберга-Марквардта и т.п.).

В соответствии с данным подходом каждая бактерия с вероятностью P_{ed} размещается в случайно выбранной точке пространства поиска.

Таким образом, проверяется условие: $U^i < P_{ed}$, где U^i – случайное число в интервале $[0; 1]$ для i -ой бактерии.

Если данное условие выполняется, то бактерия помещается в позицию $X^i(j, k, l)$, полученную случайным образом.

Шаг 10. Если $l < N_{ed}$, тогда выполняется переход к шагу 2, в противном случае – к шагу 11.

Шаг 11. Выбирается и сохраняется лучшее решение J_{best} и соответствующая позиция X_{best} , в которой достигается лучшее решение J_{best} .

Шаг 12. Проверка на окончание поиска. Если были выполнены все циклы для всех агентов, то выполняется переход к шагу 14, в противном случае выполняется перезапуск – переход к шагу 13.

Шаг 13. Перезапуск агентов: выбираются новые случайные позиции для каждого агента X^i , $i = 1, 2, \dots, S$ и рассчитываются соответствующие значения целевой функции J_i , $i = 1, 2, \dots, S$. Счётчики циклов сбрасываются в 0: $j = 0$; $k = 0$; $l = 0$.

Шаг 14. Останов.

Рассмотренный метод оптимизация на основе моделирования перемещения бактерий не учитывал одну из важных особенностей поведения бактерий: бактерии *E.Coli*, как указывалось ранее, способны группироваться, создавать пространственно-временные структуры. В связи с этим в дальнейшем было предложено несколько подходов, позволяющих учитывать в процессе моделирования перемещения бактерий и данную особенность их поведения, основными из которых являются: моделирование сигналов между клетками и применение PSO-оператора (Particle Swarm Optimization, PSO).

Описание работы алгоритма BFOA для распознавания окружности

Каждый предполагаемый круг представлен позицией бактерии, где первые два компонента вектора, x и y – координаты центра этой окружности, а третий компонент r – радиус. Пусть (x_k, y_k, r_k) будет k -м тестовым кругом популяции, где $k = 1, \dots, S$, где S – это размер популяции, т.е. обозначает общее количество тестовых кругов. В фазе инициализации случайные значения в подходящем диапазоне присвоены каждой из трех координат векторов. Давайте рассматривать $2N$ равномерно распределенных демонстрационных точек на круге.

Предположим, что искомая окружность располагается на некотором изображении. Функция $P(x, y)$ равна 1, если пиксель (x, y) является граничным, 0 – иначе. Пусть A будет граничной матрицей. Целевую функцию обозначим как J . Пусть для (x_0, y_0, r_0) тестового круга ее значение будет $J_0 = J(A, x_0, y_0, r_0)$.

Для определения является ли кандидат кругом можно рассмотреть множество кругов с центром в (x_0, y_0) и радиусом, варьирующимся в диапазоне от $r_0 - \delta$ до $r_0 + \delta$. Назовем это множество тестовой полосой. Значение δ можно взять как процент от радиуса, предлагается $\delta = r_0/8$. После этого N тестовых точек берутся для каждого круга в тестовом диапазоне. Тестовые точки расположены на круге на равном расстоянии друг от друга. Пусть i -я тестовая точка на тестовом круге имеет радиус $r_0 + j$ ($-\delta \leq j \leq \delta$) и обозначается (x_i^j, y_i^j) .

$$x_i^j = (r_0 + j) \cos \frac{2\pi}{N} i, \quad (1)$$

$$y_i^j = (r_0 + j) \sin \frac{2\pi}{N} i, \quad (2)$$

где $i = 1, 2, \dots, N; j = -\delta, -\delta + 1, \dots, 0, \dots, \delta - 1, \delta$.

Чтобы измерить степень принадлежности тестовой точки к окружности центрального круга определим функцию расстояния μ . Значение функции расстояния μ_i^j для тестовой точки (x_i^j, y_i^j) определяется как:

$$\mu_i^j = P(x_i^j, y_i^j) \exp\left(-\frac{1}{2\sigma^2} (r + j + r_0)^2\right) \quad (3)$$

Значение μ может изменяться в диапазоне от 0 до 1. Когда $\mu_i^j = 1$, это означает, что выбранная точка граничная и она находится на центральном круге. Если (x_i^j, y_i^j) не граничная точка то $\mu_i^j = 0$.

Целевая функция, соответствующая (x_0, y_0, r_0) для граничной матрицы A определяется по следующей формуле:

$$J(A, x_0, y_0, r_0) = 1 - \frac{1}{(2\delta+1)N} \sum_{i=1}^N \sum_{j=-\delta}^{\delta} \mu_i^j \quad (4)$$

Целевая функция взвешивает каждую граничную точку согласно радиусу круга кандидата. Таким образом, становится возможным с помощью алгоритма WFOA распознать окружность на изображении. Аналогичным образом можно подобрать параметры, которые идентифицируют любую другую фигуру, определить соответствующую этой фигуре целевую функцию и приведенный алгоритм можно будет использовать для поиска этой фигуры на изображении.

Описание работы алгоритма WFOA для распознавания произвольной формы

Представим бактерию позицией предполагаемой формы в виде координат x и y , ее угла поворота a и масштаба s . Форма является массивом вершин относительно ее положения.

Пусть (x_k, y_k, a_k, s_k) будет k -ой тестовой формой популяции, где $k = 1, \dots, S$, а S — это размер популяции, т.е. обозначает общее количество тестовых форм. При инициализации каждой из четырех координат векторам присваиваются случайные значения в подходящем диапазоне. Рассмотрим N демонстрационных точек многоугольника.

Предположим, что искомая форма располагается на некотором изображении. Функция $P(x, y)$ равна 1, если пиксель (x, y) является граничным, 0 – иначе. Пусть A будет граничной матрицей. Целевую функцию обозначим как J . Пусть для (x_0, y_0, a_0, s_0) тестовой формы ее значение будет $J_0 = J(A, x_0, y_0, a_0, s_0)$.

Для определения является ли кандидат искомой формой нужно вычислить целевую функцию. Чем меньше значение целевой функции, тем более точно форма соответствует кандидату. Целевая функция вычисляется на основании суммы расстояний от тестовых точек к ближайшим граничным точкам, если таковые имеются.

Показатель r представляет собой радиус окружности вокруг тестовой точки, где будет производиться поиск граничной точки. В общем случае предлагается брать его следующим образом:

$$r = \max(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}) * 0.1, \quad (5)$$

где $i = 0 \dots N, j = 0 \dots N$.

Чтобы измерить степень принадлежности тестовой точки к форме на граничной матрице, определим функцию расстояния μ . Если существует такое k и l , что $P(x_i + k, y_i + l) = 1$, то значение функции расстояния μ_i для i -ой тестовой точки определяется как:

$$\mu_i = \min\left(\frac{\sqrt{k^2 + l^2}}{r}\right), \text{ что } \square k, l: P(x_i + k, y_i + l) = 1, \quad (6)$$

где $k = -r \dots r, l = -r \dots r$.

Иначе $\mu_i = M$. Также $\mu_i = M$, если результат вычислений по формуле (6) больше, чем r , где M – число больше единицы. Предлагается брать $M = \frac{N}{2} + 1$.

Значение μ может изменяться в диапазоне от 0 до 1 или быть равно 2. Когда μ_i находится в диапазоне от 0 до 1, это означает, что граничная точка находится на расстоянии $\mu_i * r$ от тестовой точки. Если $\mu_i = M$, это означает, что нет граничной точки, которая лежит ближе, чем на расстоянии r от тестовой.

Целевая функция, соответствующая (x_0, y_0, a_0, s_0) для граничной матрицы A определяется по следующей формуле:

$$J(A, x_0, y_0, a_0, s_0) = \frac{1}{N} \sum_{i=1}^N \mu_i \quad (7)$$

Преимущества и недостатки BFO

На основе рассмотренных применений метода BFO можно сделать вывод, что метод BFO хорошо применим для задач непрерывной оптимизации, в которых следует найти минимум некоторой целевой функции.

Проведенные эксперименты по решению практических задач засвидетельствовали эффективность использования метода BFO как базового, так и с группировкой бактерий.

Можно выделить следующие преимущества метода BFO по сравнению с традиционными методами многомерной оптимизации:

- метод не выдвигает абсолютно никаких требований к целевой функции, то есть функция может быть мультимодальной, негладкой, обладать разрывами;
- для нахождения оптимума целевой функции не надо знать градиент целевой функции;
- метод является методом глобального поиска, в то время как традиционные методы в часто являются методами локальной оптимизации;
- метод обладает большим количеством настраиваемых параметров, что позволяет его лучшей адаптации для решения той или иной практической задачи оптимизации;
- метод является мультиагентным методом, за счёт чего достигается снижение времени выполнения поиска, поскольку поиск выполняется некоторым количеством агентов параллельно.
- метод является хорошо расширяемым, то есть в его работу можно легко добавлять новые операторы и тем самым создавать гибридные системы. Одним из перспективных является добавление генетических алгоритмов, которые позволяли бы моделировать более правильные эволюционные процессы, которые наблюдаются в процессе деятельности бактерий.

К недостаткам метода можно отнести:




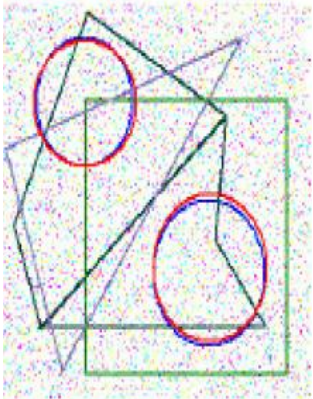
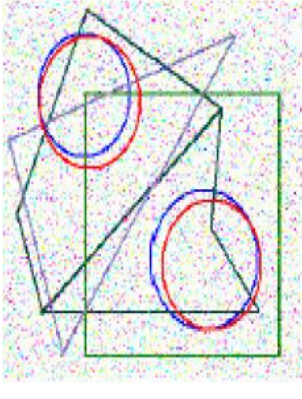
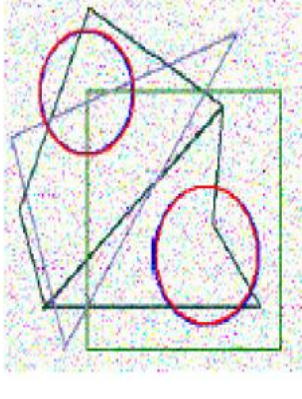
- хоть нахождение оптимума гарантируется, но скорость сходимости носит случайный характер;
- то, что при использовании метода BFO с группировкой бактерий возможно заикливание в локальных оптимумах в случае неправильной настройки параметров, влияющих на группировку бактерий;
- сильная зависимость работы метода от настраиваемых параметров;
- неполнота модели поведения бактерий и может быть доработана, поскольку процедуры воспроизведения и рассеивания ещё не доработаны.

Выводы

Проведен обзор методов, которые могут быть применены для задачи распознавания многоугольников. Выбран метод распознавания для этой задачи. Обоснованы причины выбора. Рассмотрена работа алгоритма BFOA на примере распознавания окружности и рассмотрена адаптированная работа алгоритма для распознавания многоугольников. Как мы видим, целевые функции похожи тем, что в них взвешивается каждая граничная точка согласно радиусу круга кандидата. Отличаются они своими функциями расстояния. В дальнейшем планируется адаптировать алгоритм для распознавания произвольной фигуры.

Сравнения алгоритма BFO с муравьиным и генетическим алгоритмами

Таблица 1. Сравнения алгоритмов.

Алгоритм BFOA	Муравьиный алгоритм	Генетический алгоритм
		
		

Литература

1. Kim D.H., Abraham A., Cho J.H. A Hybrid Genetic Algorithm and Bacterial Foraging Approach for Global Optimization // Information Sciences. – 2007. – № 18 (177). – P. 3918-3937.
2. S. Dasgupta, S. Das, A. Biswas, A. Abraham Automatic circle detection on digital images with an adaptive bacterial foraging algorithm // Soft Comput. – 2010. – P. 1151–1164.
3. Woodward, D. E., Tyson, R., Myerscough, M. R., Murray, J. D., Budrene, E. O., Berg, H. C., Spatio-Temporal Patterns Generated by Salmonella Typhimurium, Biophysical Journal, Vol. 68, pp. 2181–2189, 1995.
4. K.M.Passino, “Biomimicry of bacterial foraging for distributed optimization and control”, IEEE Control System Magazine, pp.52-67, June 2002.

Минков О.В., Ручкин К.А. Распознавание окружностей на изображении с помощью алгоритма BFOA. Выполнен обзор методов, которые могут быть применены для задачи распознавания многоугольников. Рассмотрена работа алгоритма BFOA на примере распознавания окружности. Рассмотрена модификация алгоритма BFOA для распознавания многоугольников.

Ключевые слова: алгоритм, распознавание, BFOA, изображение, окружность

Minkov O.V., Ruchkin K.A. Recognition of circles in an image using BFOA algorithm. A review of techniques that can be applied to recognition problems polygons. Consider working BFOA algorithm on the example of the circle of recognition. A modified algorithm for recognizing BFOA polygons.

Keywords: algorithm, pattern recognition, BFOA, the image, circle

СЕКЦИЯ «МЕТОДЫ И СРЕДСТВА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ»

УДК 004.94

Разработка и исследование виртуальных рабочих мест в среде OpenSim

С.И. Бочков, А.С. Степанов

Ульяновский государственный технический университет
bochkovsam1@rambler.ru, step_al_ul@mail.ru

Аннотация: в статье рассмотрен такой инструмент виртуализации учебного процесса, как виртуальные миры. Приведён краткий обзор мировых разработок в данных областях, рассмотрены особенности популярной платформы OpenSim, также дано описание проектов виртуальных рабочих мест, разрабатываемых в УлГТУ. В качестве резюме приводятся пути совершенствования проектов.

Ключевые слова: виртуальный мир, дистанционное обучение, электронное обучение, обучение навыков, интеллектуальная среда.

Введение

Промышленность часто сталкивается с необходимостью набора новых кадров. Зачастую это только что закончившие ВУЗ молодые специалисты, и сразу же допускать их к работе за реальными рабочими местами было бы неразумно, поскольку используется дорогостоящее оборудование, порождающее побочный эффект в виде опасных факторов производства. К тому же оборудование по неопытности можно вывести из строя. Можно проводить обучение с помощью уже опытного мастера, но ему нужно доплачивать за каждого ученика, что в нынешних условиях финансирования весьма затратно.

Поэтому возникает потребность в виртуальных тренажёрах, позволяющих получить представление о технологических процессах на предприятии и пройти практику перед допуском к реальному рабочему месту.

Предыстория

Разработка и внедрение тренажёрных систем и виртуальных миров в процессы компьютеризованного обучения является в настоящее время актуальной задачей, имеющей большое практическое значение.

Используемые с 1990 в армии и медицине впервые виртуальные миры привлекли внимание масс в 2003, когда Linden Lab запустила знаменитый виртуальный мир Second Life [1].

Бум в этой области пришёлся на 2006-2008 гг., когда в виртуальных мирах появились копии сотен знаменитых музеев и картинных галерей со всего мира, посольства открыли виртуальные двери страждущим проконсультироваться по визовому вопросу, а археологи моделировали прототипы древних поселений.

Однако вскоре Linden Lab сократила штат разработчиков на треть, коммерческие компании начали постепенно закрывать виртуальные представительства. Кризис охватил виртуальные миры в организациях и университетах, когда те стали сокращать бюджеты и подводить итоги первых лет внедрения платформ «дополненной реальности». Так, в 2010 году администрация Техасского Университета, потратившего на виртуализацию 16 кампусов \$250000, не сочла виртуальный мир действенным инструментом достижения его целей – усиления совместной работы, дистанционного образования, виртуальных научных работ [1].

Текущие разработки

Виртуальные миры широко применяются в медицине, где поле для разработок весьма обширно – с помощью виртуальных программ можно тренировать молодых врачей, отрабатывать командные операции. Европейское Космическое агентство работает над созданием системы дополненной реальности для обучения космонавтов оказания помощи и диагностики заболеваний друг друга в космосе.

Прототип системы прошёл успешные испытания в больнице университета Сан-Пьер в Брюсселе [2]. В 2012 году Агентство по исследованиям в области безопасности (Defense Advanced Research Projects Agency) разработала 3D контактные линзы для военных [3].

Обзор платформы OpenSim

OpenSimulator (далее – OpenSim) представляет собой серверную платформу для создания многопользовательских трёхмерных виртуальных миров. Сервер OpenSim обслуживает один или несколько участков виртуальной земли (регионов) и может быть запущен как отдельно (*standalone mode*), так и в составе сети серверов (*grid mode*). Подключённые к серверу пользователи представлены в виде своих 3-х мерных виртуальных образов – аватаров.

Сферы применения OpenSim:

- виртуальные представительства организаций (офисы, образовательные учреждения и др.);
- 3D-моделирование;
- моделирование физических процессов;
- ландшафтный дизайн.

Для программирования сценариев используется язык LSL (Linden Scripting Language), разработанный компанией Linden Labs [4,5]. Также допустимо написание сценариев на языке C# [6]: несмотря на снижение читаемости кода, данный подход позволяет использовать дополнительные библиотеки.

OpenSim написан на языке C#, представляет собой проект с открытым исходным кодом. Поскольку это серверное приложение, для предоставления виртуального мира в графическом режиме требуется клиентское приложение.

Разработка виртуальных рабочих мест

В лаборатории НИР ИДДО УлГТУ была поставлена задача создания виртуальных рабочих мест для повышения эффективности и сокращения сроков обучения и переподготовки по рабочим специальностям. Данная система при этом должна включать в себя модули выдачи ошибок и предупреждений, а также оценивания пользовательских действий.

В настоящее время идёт работа по внедрению электронного обучения по рабочим специальностям на Ульяновском механическом заводе – монтажник, слесарь-сборщик и регулировщик РЭАиП (рис.1).



Рисунок 4 – Общий вид на виртуальную лабораторию

На рис.2 приведена структура виртуального тренажёра. Клиентская среда визуализирует пространство, исходя из данных, хранящихся в БД OpenSim. В свою очередь, OpenSim передаёт команды ввода-вывода на серверную часть. Последняя отвечает за хранение истории пользовательских действий, параметров состояния рабочих столов, а также сценариев ввода-вывода в указанные файлы.

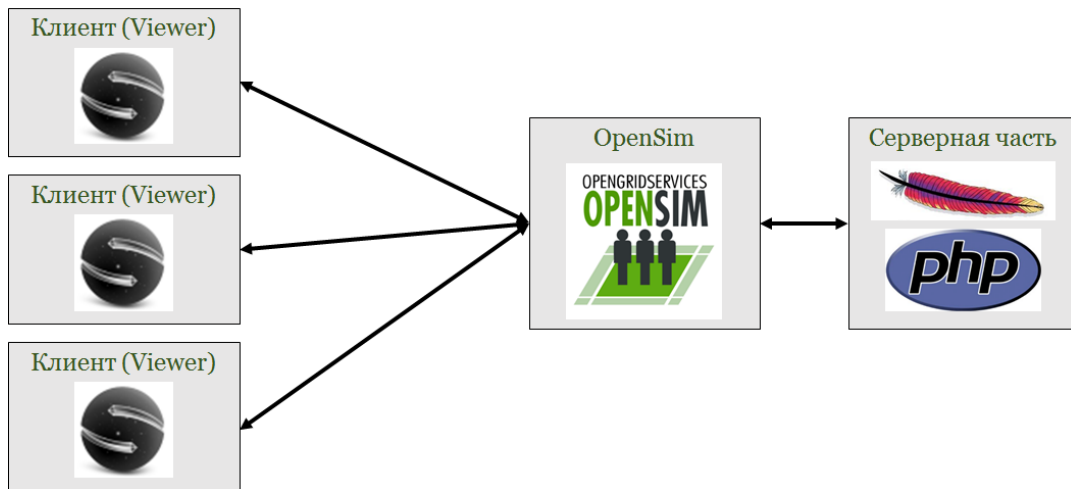


Рисунок 5 - Организация виртуального пространства

В качестве примера рассмотрим виртуальное рабочее место монтажника РЭАиП. Особое внимание уделяется процессам составления алгоритмов обучения. Так, для монтажника проектируются два сценария:

- пайка элементов;
- пайка проводов.

Пайка элементов включает в себя подсценарии припаивания следующих радиоэлементов:

- резистор С2-33Н;
- микросхема 1533ЛА3;
- конденсаторы К52-1 (электролитический), К10-176 (керамический).

Для каждого элемента предусмотрены свои параметры пайки, такие как посадочное место на печатной плате, время и температура пайки.

В каждое виртуальное рабочее место встроена система анализа ошибок. Если обучающийся совершил какое-либо неверное действие, система пытается вернуться к предыдущему состоянию, предварительно сигнализируя об ошибке с помощью текстовых сообщений.

Для написания сценариев преимущественно используется язык LSL. В листинге, приведенном ниже, показан образец сценария. Указанный программный код описывает поведение камеры при наведении и нажатии левой кнопки мыши на объекты рабочего места.

```

default
{
    touch_start(integer num) {
        if (isTriggerOn(atKey)) {
            llSitTarget(positionOnChair, ZERO_ROTATION);
            llSetClickAction(CLICK_ACTION_SIT);
            llRequestPermissions(llDetectedKey(0), PERMISSION_CONTROL_CAMERA);
        }
        else {
            llInstantMessage(llDetectedKey(0), "Наденьте халат!!!");
        }
    }
    changed(integer change) {
        llClearCameraParams();
        if (llAvatarOnSitTarget() == NULL_KEY) {
            readRequest = sendRequest("grade.php", "user="+agent+"&val=READ");
            osSetPrimitiveParams(ztKey, [PRIM_COLOR, ALL_SIDES, <1,0,0>, 1]);
            osSetPrimitiveParams(stKey, [PRIM_COLOR, ALL_SIDES, <1,0,0>, 1]);
            llSetAlpha(1, ALL_SIDES);
            llSetClickAction(CLICK_ACTION_TOUCH);
            agent = "";
        }
        else {
            osSetPrimitiveParams(stKey, [PRIM_COLOR, ALL_SIDES, <0,1,0>, 1]);
        }
    }
    vector camPos;
    vector focusOffset;
    float distance;
    string val = Float2String(llGetAlpha(ALL_SIDES) / llGetNumberOfSides(), 2, TRUE);
}
  
```

```

listen(integer channel, string name, key id, string message) {
    list lst = llParseString2List(message, [" "], [1]);
    integer zoom = llList2Integer(message, 0);
    vector camPos;
    vector focusOffset;
    float distance;
    if (zoom) {
        camPos = llList2Vector(lst, 1);
        focusOffset = llList2Vector(lst, 2);
        distance = llList2Float(lst, 3);
    } else {
        camPos = idealTopPosition;
        focusOffset = <0.8, 0.062, -1.1>;
        distance = 0.7;
    }
    moveCamera(camPos, focusOffset, distance);
}

```

В ходе обучения монтажник должен приобрести следующие знания и навыки:

- температурный режим пайки;
- время пайки;
- последовательность выполнения действий при монтаже радиоэлементов типа «микросхема», «конденсатор»;
- зачистка и облуживание проводов;
- пайка проводов.

На рис.3 показан процесс пайки. Для лучшей видимости припой на паяльнике выделен красным цветом (1). Курсор мыши установлен на контактной площадке, о чём свидетельствует всплывающее сообщение (2), при этом время пайки не соответствует требуемому (3).

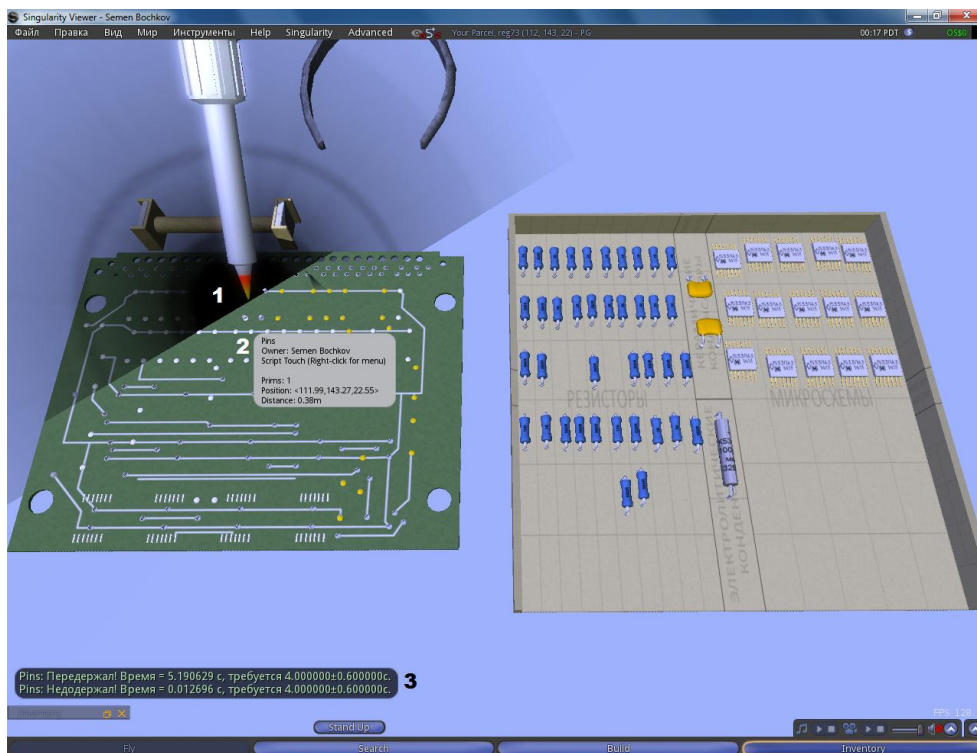


Рисунок 6- Процесс пайки

Заключение

Последние исследования показали, что среди взрослой аудитории виртуальные «я» есть лишь у 4% людей. Сегодня виртуальные представительства компаний на публичных платформах больше похожи на «города-призраки», однако для решения внутренних нужд продолжают создаваться «закрытые» площадки виртуальной реальности [7-9].

К таким закрытым площадкам относятся виртуальные рабочие места, разрабатываемые в Ульяновском государственном техническом университете. Её отличие от виртуальных лабораторных площадок многих других университетов состоит в том, что она с высокой точностью и приближённостью к действительности моделирует рабочее место сотрудника завода.

Практическим результатом данного проекта является система формирования навыков у обучающегося при выполнении типичных рабочих операций.

Перспективными направлениями в области виртуализации являются:

- совершенствование проектируемых виртуальных рабочих мест;
- внедрение экспертной системы, оценивающей действия пользователя;
- интеграция системы виртуальных рабочих мест с LMS Moodle;
- освоение новых областей в сфере науки и промышленности для применения технологий виртуализации обучения.

Литература

1. Виртуальные миры: практическое применение / Блог компании Гарс Телеком / Хабрахабр [Электронный ресурс] – Режим доступа: <https://habrahabr.ru/company/garstelecom/blog/140081/>, свободный.
2. Diagnosing Medical Problems in Space Using Augmented Reality - The Atlantic [Электронный ресурс] – Режим доступа: <http://www.theatlantic.com/health/archive/2012/02/diagnosing-medical-problems-in-space-using-augmented-reality/252880/>, свободный.
3. <http://www.businessinsider.com/virtual-reality-contact-lenses-will-give-troops-a-tactical-edge-on-the-battlefield-2012-3> US Defense Agency Develops Virtual Reality Contact Lenses - Business Insider. – Режим доступа: <http://www.businessinsider.com>
4. LSL Portal - Second Life Wiki [Электронный ресурс] – Режим доступа: http://wiki.secondlife.com/wiki/LSL_Portal, свободный.
5. OpenSim [Электронный ресурс] – Режим доступа: www.opensimulator.org, свободный.
6. C# scripting in OpenSim [Электронный ресурс] – Режим доступа: <http://justincc.org/blog/2008/11/21/c-scripting-in-opensim/>, свободный.
7. Бахарева В.А., Захарова У.С., Сербин В.А., Фещенко А.В. Технологии виртуальной и дополненной реальности в образовательной среде вуза. // Открытое и дистанционное образование. Томск, 2015. №4 (60). – с. 12-20.
8. ИНОБР > Проекты > Компьютерные тренажерные системы [Электронный ресурс] – Режим доступа: http://misis.ru/inobr/kts_inobr, свободный.
9. Трухин А.В. Анализ существующих в РФ тренажерно-обучающих систем // Открытое и дистанционное образование. Томск, 2008. №1 (29). – с. 32-40.

***Аннотация:** в статье рассмотрен такой инструмент виртуализации учебного процесса, как виртуальные миры. Приведён краткий обзор мировых разработок в данных областях, рассмотрены особенности популярной платформы OpenSim, также дано описание проектов виртуальных рабочих мест, разрабатываемых в УлГТУ. В качестве резюме приводятся пути совершенствования проектов.*

***Ключевые слова:** виртуальный мир, дистанционное обучение, электронное обучение, обучение навыков, интеллектуальная среда.*

***Abstract:** in the paper such tool as virtualization of educational process is considered. Review of world famous projects in this field is provided, features of OpenSim platform are considered. Also description of designed in UlSTU virtual workplaces is given. The article is concluded with the ways of upgrading these projects.*

***Keywords:** virtual world, distance education, skills development, intellectual environment.*

Интегрированная инструментальная подсистема генетического поиска

Л.В. Курейчик, В.В. Курейчик, Вл.Вл. Курейчик
Южный федеральный университет
kureychik@sfedu.ru

Курейчик Л.В., Курейчик В.В., Курейчик Вл.Вл. Интегрированная инструментальная подсистема генетического поиска. В работе рассматривается Интегрированная инструментальная подсистема генетического поиска. Описана структурная схема подсистемы генетического поиска. Рассмотрен перечень задач, возникающие перед пользователем при проектировании генетического алгоритма.

Ключевые слова: Инструментальная подсистема, генетический алгоритм, поиск, среда, схема.

Введение

Подсистема генетического поиска предназначена для решения задач с использованием одного из методов эволюционного моделирования – генетического поиска. Рассмотрим перечень задач, которые возникают перед пользователем при проектировании генетического алгоритма [1,2].

1. Создание архитектуры макрогенетического алгоритма (описание структуры, кодирующей решение задачи; механизмов кодирования, декодирования и оценки приспособленности решений; определение набора генетических операторов и последовательность их выполнения; параметров эволюции популяции; критериев прекращения эволюционного поиска).

2. Создание архитектуры метагенетического алгоритма (определение количества и характеристик макрогенетических алгоритмов, входящих в состав метагенетического алгоритма, параметров эволюции отдельных популяций, направлений и механизмов миграции особей между отдельными популяциями, критериев прекращения эволюционного поиска).

Поиск решения оптимизационной задачи, для которой в среде подсистемы генетического поиска был сконструирован генетический алгоритм. Подавая на вход подсистемы генетического поиска описание генетического алгоритма и исходные данные решаемой задачи, пользователь получает решение оптимизационной задачи в формате, предназначенном для его использования в практических целях. В случае решения задачи, относящейся к области автоматизированного проектирования, – в формате, совместимом с промышленными САПР.

Описание задач

Задачи, для которых предназначена подсистема генетического поиска, условно можно разделить на три группы. Первая группа – это проектирование и отладка разнообразных генетических алгоритмов. Вторая – анализ эффективности и иных показателей генетических операторов и алгоритмов. Третья – решение практических задач с помощью генетических алгоритмов, спроектированных в среде интегрированной инструментальной подсистемы генетического поиска.

Описанное деление задач определяет режимы функционирования, которые должны быть реализованы в подсистеме генетического поиска: режим проектирования; режим анализа; режим исполнения.

Рассмотрим названные режимы функционирования подсистемы генетического поиска подробнее.

Проектирование и отладка генетических алгоритмов нацелено на выработку новых проблемно-ориентированных методов решения конкретного круга задач, и их практическое использование. Особенности предметной области задачи, решаемой с помощью генетического алгоритма, влияют на выбор морфологической и генетической структуры особи, кодирующей её решение. Особенности представления данных определяют круг генетических операторов, которые могут быть применены при проектировании алгоритма. На данном этапе решаются задачи, связанные с применяемыми стратегиями селекции особей, допускаемых к размножению, правил оценки приспособленности особей (целевой функции), а также правил выживания в популяции (стратегия отбора).

Генетический алгоритм, построенный с помощью интегрированной инструментальной подсистемы генетического поиска, должен отвечать критериям корректности, сходимости, функциональности, полноты и непротиворечивости.

Сходимость генетического алгоритма предполагает возможность получения решения задачи с помощью данного алгоритма за конечное число шагов или за ограниченное время [1].

Понятие непротиворечивости алгоритма имеет смысл для алгоритмов, относящихся к классу разветвляющихся [3], то есть таких, которые предусматривают выбор одной или нескольких альтернативных последовательностей действий в зависимости от исходных и промежуточных данных. Генетические алгоритмы относятся к данному классу, поскольку содержат как минимум два логических блока, определяющих ветвления и реализуемых в операторах селекции и отбора. Генетический алгоритм является непротиворечивым, если для каждой ветви генетического алгоритма существует хотя бы один набор исходных данных, при котором выполняется данная ветвь алгоритма [1].

Корректным (правильным) будем считать генетический алгоритм, который на любом допустимом для данной задачи наборе исходных данных (входных параметров) заканчивает свою работу и выдает результат, удовлетворяющий требованиям задачи.

Соответствие разработанного генетического алгоритма трем названным критериям проверяется путем верификации алгоритма.

Подсистема генетического поиска должна предоставлять инструменты статистической обработки, носящие универсальный характер. Более того, результаты, полученные в ходе исследования эффективности генетических операторов и алгоритмов, должны быть непротиворечивыми и сравнимыми с результатами исследования других генетических операторов и алгоритмов.

Следует отметить, что введение в генетический алгоритм дополнительных блоков не влияет на достоверность показателей производительности исследуемого генетического оператора или алгоритма, поскольку учет затрат ресурсов вычислительной системы ведется отдельно для каждого оператора.

Поскольку целью любого алгоритма является получение приемлемых результатов за минимальное время, в режиме исполнения из структуры генетического алгоритма исключаются все блоки (операторы), предназначенные для сбора и обработки вспомогательной информации в режимах отладки и анализа данного алгоритма.

Подсистема генетического поиска формирует среду исполнения генетического алгоритма. При этом выполняются следующие действия: создание переменных окружения; загрузка подключаемых модулей, содержащих исполняемый код генетических операторов; загрузка исходных данных генетического алгоритма, а также проблемно-ориентированных знаний о решаемой задаче; создание функциональных объектов генетического алгоритма и управление их жизненным циклом; вывод результатов работы генетического алгоритма.

Такой подход очень удобен для интеграции созданного генетического алгоритма в различные системы автоматизированного проектирования [3].

Подсистема генетического поиска

При разработке интегрированной инструментальной подсистемы генетического поиска используются приемы объектно-ориентированного проектирования программных систем [4]. На рис. 7. приведена структурная схема подсистемы генетического поиска.

В данной подсистеме выделяются две основные части:

1. Подсистема взаимодействия с пользователем, предоставляющая дружественные средства ввода и отображения информации. Данная подсистема реализована с использованием графического интерфейса пользователя (ГИП) [5].

2. Подсистема моделирования и анализа реализует возможность применения ППП для решения практических задач. На данную подсистему возложены следующие задачи: верификация, сборка и исполнение описанного генетического алгоритма; подключение внешних модулей, содержащих исполняемый код оператора; сбор, накопление и анализ статистической информации; ввод исходных данных и выгрузка результатов работы генетического алгоритма; преобразование данных из одного формата в другой; обеспечение безопасности функционирования подсистемы генетического поиска и пр. В дальнейшем данную подсистему будем называть ядром подсистемы генетического поиска (Ядром ППП).

Подсистема взаимодействия с пользователем реализует оконный интерфейс. Инструменты ППП подразделяются на следующие группы:

- инструменты проектирования: мастер выбора структуры особи предназначен для определения морфологической и генетической структуры особи, способа декодирования и оценки приспособленности особи;
- мастер выбора архитектуры ГА;
- построитель схем включает панель компонентов, окно свойств компонента (оператора), дизайнер схемы алгоритма, редактор описания, панель связей, панель верификации схемы алгоритма, мастер сохранения схемы алгоритма;

- мастер детализации описания ГА позволяет выполнить преобразование обобщенного описания ГА в детализированное;
- мастер экспорта решения позволяет выбрать формат данных, в который будет экспортировано найденное решение;
- редактор кода ГО;
- инструменты анализа, обеспечивающие выбор объекта исследования, выбор исследуемых характеристик и алгоритмов анализа, выбор варьируемых параметров, а также отображение результатов (графики, гистограммы таблицы и пр.) и сохранение результатов.

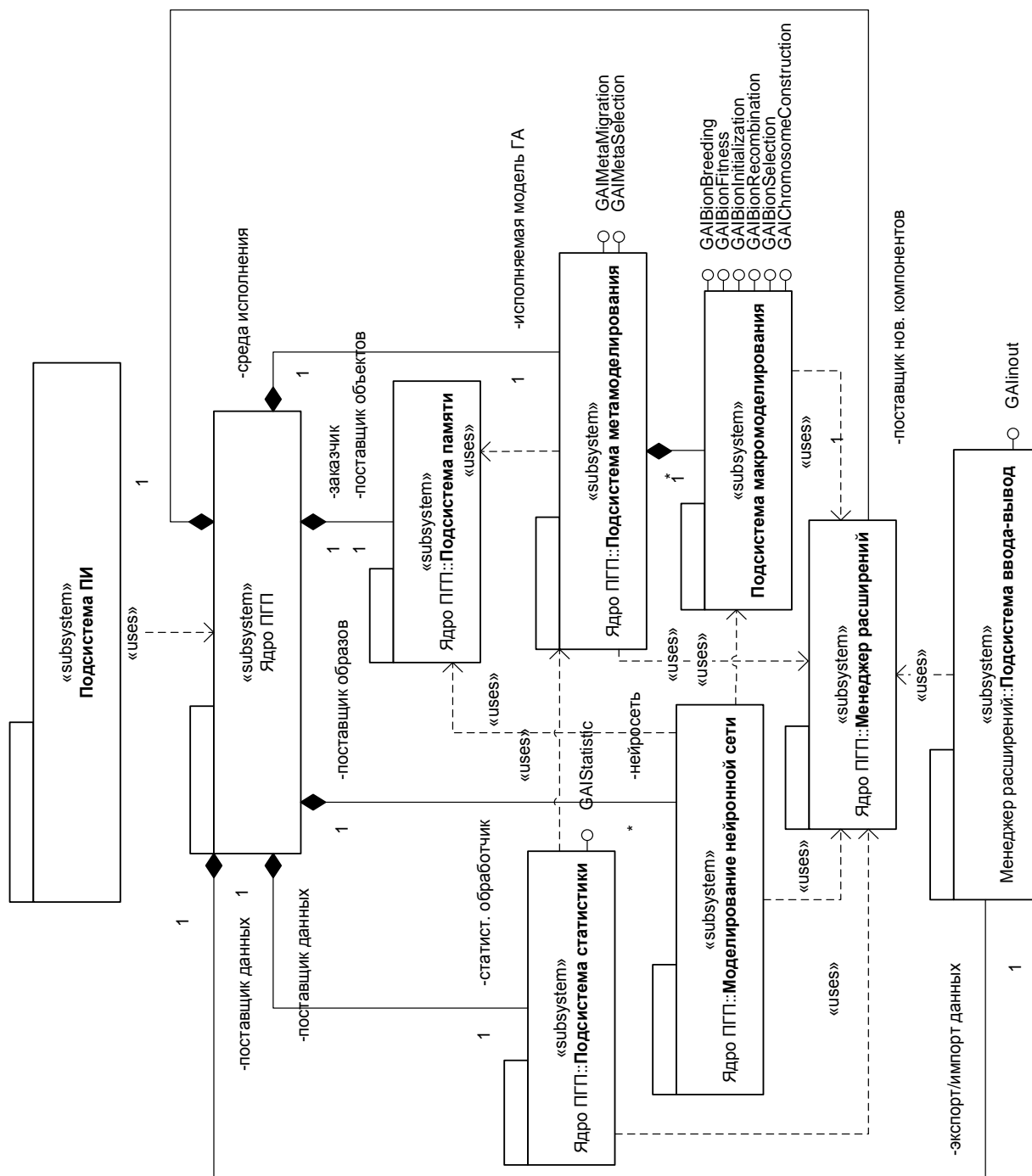


Рисунок 8 - Структурная схема подсистемы генетического поиска

Рассмотрим ядро подсистемы генетического поиска подробнее. Многообразие функций, возложенных на ядро ППП, определяет его сложную внутреннюю структуру. Прежде всего, следует выделить подсистему эволюционного моделирования, назначение которой состоит в интерпретации описания макро- и метагенетических алгоритмов, в создании соответствующей данному описанию структуры данных, а также в

выполнении генетического поиска, с целью получения решения поставленной оптимизационной задачи. В связи с открытостью архитектуры подсистемы генетического поиска другой важной частью ядра ППП является менеджер расширений, основанная задача которого состоит в интеграции в ППП различных операторов. Данная подсистема взаимодействует со всеми структурными элементами ядра ППП, расширяя их функциональность.

Единство операционной среды, формируемой ядром ППП, обеспечивается подсистемой памяти, назначение которой состоит в создании всех внутренних объектов ядра ППП, а также в управлении их жизненным циклом [5].

Подсистема генетического поиска должна обеспечивать совместимость с существующими системами автоматизированного проектирования на уровне форматов файлов. Экспорт, импорт и преобразование данных из одного формата в другой осуществляется подсистемой ввода-вывода [5]. Совместимость с существующими системами достигается за счет возможности интеграции новых алгоритмов преобразования (конвертации) данных с помощью менеджера расширений.

Предлагается организация подсистемы эволюционного моделирования в виде двухуровневой иерархической структуры.

Нижний уровень иерархии отвечает за макроэволюционное моделирование и может содержать один или несколько независимых генетических алгоритмов. Элементы данного уровня в ППП реализованы в виде объектов подсистемы макроэволюционного моделирования (GAMakroEvolutionEngine), каждый из которых содержит собственный набор данных:

- множество альтернативных решений (популяция особей);
- описание генетического алгоритма;
- набор управляющих параметров генетического алгоритма.

Верхний иерархический уровень предназначен для моделирования метаэволюционного поиска (GAMetaEvolutionEngine). В ППП создается только один экземпляр подсистемы метаэволюционного моделирования, который содержит следующие данные:

- описание метагенетического алгоритма, а также правил отбора и путей миграции особей между популяциями;
- набор описаний макроэволюционных алгоритмов;
- набор управляющих параметров генетического алгоритма.

Заключение

Определены особенности функционирования интегрированной инструментальной подсистемы генетического поиска. На основе их анализа были выделены режимы функционирования ППП: проектирования; анализа; исполнения.

Декомпозиция ППП на составляющие части: позволит реализовать базовые алгоритмы и инструменты эволюционного моделирования. Она определяет возможности расширения подсистемы генетического поиска. В основу организации ППП положены две концептуальных модели – модель представления решения задачи и модель представления генетического алгоритма. Особенно важно, что конечный пользователь ППП может определить собственный способ кодирования решения поставленной задачи. Это позволит повысить эффективность и качество результатов поиска оптимальных решений.

Литература

1. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. М.: Физматлит. 2006.
 2. Курейчик В.В., Нужнов Е.В. «Возможности организации интегрированной инструментальной среды поддержки процедур генетического поиска и оптимизации решений» // Прикладные информационные технологии и интеллектуальные системы №4 Науч. изд. Таганрог, 2003.
 3. Норенков И.П. Основы автоматизированного проектирования. –М.: Изд-во МГТУ имени Н.Э.Баумана, 2004. – 336с.
 4. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. «Приемы объектно-ориентированного проектирования: паттерны проектирования» – СПб: Питер, 2001 – 368с.
 5. Бакало М.А., Курейчик В.В. Концепция построения системы поддержки генетических алгоритмов // Прикладные информационные технологии и интеллектуальные системы №4 Науч. изд. Таганрог, 2006.
- Kureychik, L.V., Kureychik V.V., Kureychik V.I. Integrated tool subsystem genetic research. The paper deals with Integrated Tool subsystem genetic search. We describe a block diagram of a subsystem of genetic research. We consider the list of problems faced by the user in the design of genetic algorithm.*
- Keywords:** Instrumental subsystem, genetic algorithm, search, environment scheme.

Использование метода муравьиной колонии для построения кратчайших связывающих сетей

Б.К. Лебедев, О.Б. Лебедев, Е.М. Лебедева
Южный федеральный университет
lebedev@sfedu.ru

Лебедев Б.К., Лебедев О.Б., Лебедева Е.М. "Использование метода муравьиной колонии для построения кратчайших связывающих сетей". В работе представлен алгоритм построения кратчайших связывающих сетей на основе метода муравьиной колонии. При разработке алгоритма были предложены новые технологии, принципы и механизмы решения задачи построения кратчайших связывающих сетей, основанные на моделировании процессов адаптивного поведения муравьиной колонии. В соответствии со спецификой поставленной задачи были объединены в целое математические методы, в которых заложены принципы природных механизмов принятия решений. Проведены экспериментальные исследования и получены результаты, подтверждающие в целом теоретические разработки.

Ключевые слова: Муравьиная колония, кратчайший путь, сеть, метод, СБИС.

Основные положения

Проблема построения кратчайших связывающих сетей одна из наиболее сложных при автоматизированном проектировании внутрисхемных соединений электронных устройств. Поскольку при машинной трассировке соединений используется ортогональная опорная сетка, интерес представляет построение связывающих деревьев Штейнера в ортогональной метрике [1]. В настоящее время резкое повышение функциональной сложности СБИС стимулирует разработку новых эффективных методов и средств их проектирования.

Отдельные попытки [1-3] применения эволюционного моделирования к задаче построения деревьев Штейнера были достаточно успешными. Однако предложенные структуры ГА фактически являются "слепыми" поисковыми структурами с присущими им недостатками: генерация большого количества подобных решений; генерация большого количества "плохих" решений; генерация решений с нарушениями, что требует дополнительного контроля.

В последние годы интенсивно разрабатывается научное направление с названием «Природные вычисления» (Natural Computing), объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений [Курейчик, 1-4]. К таким методам можно отнести, прежде всего, методы моделирования отжига [5], методы эволюционного моделирования [2], генетические алгоритмы [4-7], эволюционной адаптации [8], алгоритмы роевого интеллекта [7-8] и муравьиные алгоритмы (Ant Colony Optimization - ACO) [8]. Идея муравьиного алгоритма - моделирование поведения муравьев, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи. Основу поведения муравьиной колонии составляет самоорганизация, обеспечивающая достижения общих целей колонии на основе низкоуровневого взаимодействия, благодаря которому, в целом, колония представляет собой разумную многоагентную систему. Особенности являются наличие непрямого обмена, который и используется в муравьиных алгоритмах. В работе используется представление задачи построения связывающих деревьев Штейнера в виде адаптивной муравьиной системы, на основе сочетания принципов самообучения, самоорганизации и коллективной адаптации.

Формулировка проблемы и подходы к построению дерева Штейнера

Известная проблема Штейнера состоит в следующем [8]. Дано множество точек на плоскости: $P = \{p_i | i=1, 2, \dots, n\}$. Требуется найти дерево $S=(X, U)$, с множеством вершин X и множеством ребер U таких, что $P \subset X$, и суммарная длина ребер U минимальна. В случае ортогональной метрики расстояние между точками с координатами (x_i, y_i) и (x_j, y_j) определяется по формуле $d_{ij} = |x_i - x_j| + |y_i - y_j|$.

Отличительной особенностью дерева Штейнера является то, что оно кроме основных вершин P содержит некоторое множество дополнительных вершин $Q = X \setminus P$ [13].

Проведем через множество точек $P=\{p_i | i=1,2,\dots,n\}$, расположенных на плоскости линии, параллельные осям X и Y . Образуется ортогональная сетка. Обозначим через V множество точек пересечений линий сетки и назовем их узлами сетки. Сопоставим ортогональной сетке граф $G=(V,E)$. Ряд теорем [Курейчик В.В., 2009] доказывает, что существует дерево Штейнера, все дополнительные вершины которого принадлежат V . Другими словами, при отыскании дополнительных точек можно ограничиться узлами ортогональной сетки $G=(V,E)$, а само дерево Штейнера полностью укладывается в ортогональной сети. При этом число этих точек $Q \leq n-2$.

Исходная задача эквивалентна задаче отыскания в графе $G=(V,E)$ дерева $G^*=(V^*,E^*)$, имеющего минимальный суммарный вес ребер и включающего заданный набор вершин P графа G , т.е. $P \subset V^* \subset V$, $E^* \subset E$.

Назовем цепь в графе $G=(V,E)$, связывающую две основные вершины, s -маршрутом.

Задача построения МДШ сводится к задаче построения и выбора $(n-1)$ -го s -маршрутов, связывающих n основных вершин. К настоящему времени сложились два основных подхода к построению МДШ [5]. Первый подход базируется на идеях волнового алгоритма Ли и заключается в последовательной реализации s -маршрутов.

Второй подход базируется на комбинаторных принципах. Задача решается в два этапа. На первом этапе формируется набор альтернативных вариантов s -маршрутов, заведомо большей размерности, чем $(n-1)$, покрывающий МДШ. На втором этапе из сформированного набора выбирается $(n-1)$ s -маршрутов, покрывающих МДШ.

Рассмотрим процесс формирования альтернативных вариантов s -маршрутов [8-15]. На множестве вершин P алгоритмом Прима строится минимальное связывающее дерево (МСД) $R=\{R_l | l=1,2,\dots,n\}$, $R_l=(p_i, p_j)$ – ребро МСД. Для каждого ребра R_l на графе G строится набор $S_l=\Gamma(R_l)$ альтернативных вариантов s -маршрутов, равный $S_l=\{s_lz | z=1,2,\dots,m\}$. $S=\{S_l | l=1,2,\dots,n\}$ – множество наборов. Формирование S_l осуществляется исходя из следующих посылок: длина s_lk должна быть минимальна; варианты различных s -маршрутов должны обеспечивать максимально возможное совпадение друг с другом.

В общем случае в набор S_l для ребра $R_l=(p_i, p_j)$ включаются все маршруты минимальной длины, соединяющие p_i и p_j в ортогональном графе G .

В работе рассматриваются два подхода к построению МДШ последовательный и комбинаторный и их комбинация, но все они базируются на методе муравьиной колонии.

Механизмы построения дерева Штейнера на основе муравьиной колонии

Для поиска решения задачи используется ортогональный граф $G=(V,E)$, построенный по выше приведенной методике. Построение МДШ осуществляется на базе предварительно построенного алгоритмом Прима МСД $R=\{R_l | l=1,2,\dots,n\}$.

При использовании последовательного подхода к построению МДШ за каждым ребром $R_l=(p_i, p_j)$ закрепляются два муравья a_{l1} и a_{l2} , причем муравей a_{l1} помещается в вершину p_i , а муравей a_{l2} в вершину p_j . Задачей каждого муравья является построение на графе G s -маршрута, связывающего вершины p_i и p_j . В общем случае за каждым ребром может быть закреплено несколько пар муравьев. Задачей колонии муравьев является построение МДШ. Каждый муравей откладывает на ребрах построенного s -маршрута феромон. На начальном этапе на всех ребрах графа G откладывается одинаковое (небольшое) количество феромона, равное Q/m , где $m=|E|$.

Формирование каждым муравьем s -маршрута осуществляется последовательно (пошагово). Изначально для каждого муравья, расположенного в вершине p_i , задаются координаты вершины p_j к которой прокладывается s -маршрут и определяются возможные направления его перемещения по ребрам ортогонального графа $G=(V,E)$, обеспечивающие построение s -маршрута минимальной длины. Так, например, для муравья, прокладывающего s -маршрут из вершины p_i в p_j возможные направления его перемещения вниз и влево. А для муравья, прокладывающего s -маршрут из вершины p_j в p_i возможные направления его перемещения вверх и вправо. При этом муравей не должен выходить за границы прямоугольника, диагональ которого связывает вершины p_j и p_i . На каждом шаге t муравей формирует список $Kl(t)$ ребер - кандидатов для включения в s -маршрут в соответствии возможными направлениями перемещения из вершины, в которой он расположен, и ограничениями. В s -маршрут включается одно из ребер списка. На конечном шаге $t=n_l$ l -м агентом будет сформирован s -маршрут s_l .

При использовании комбинаторного подхода к построению МДШ изначально для каждого ребра R_l , строится набор $S_l=\Gamma(R_l)$ альтернативных вариантов s -маршрутов $S_l=\{s_lz | z=1,2,\dots,m\}$. За каждым ребром $R_l=(p_i, p_j)$ закрепляются один муравей. В общем случае за каждым ребром может быть закреплено несколько муравьев. На каждой итерации каждый муравей a_l выбирает вариант s -маршрута из набора $S_l=\{s_lz | z=1,2,\dots,m\}$.

Моделирование поведения муравьев в задаче связано с распределением феромона на ребрах графа G . При этом вероятность включения на шаге $t+1$ ребра $e_j \in Kl(t+1)$ в формируемый отдельным муравьем маршрут $s_l(t)$, пропорциональна количеству феромона на ребре e_j .

Чем больше феромона будет отложено на ребрах маршрута, следовательно, большее количество муравьев будет включать эти ребра в собственные маршруты. Для избегания преждевременной сходимости используется отрицательная обратная связь в виде испарения феромона.

В работе впервые используется двухуровневая система непрямого обмена информации, использующая для этих целей в качестве носителя ортогональный граф $G=(V,E)$. Непрямой обмен – стигмерги (stigmergy), представляет собой разнесённое во времени взаимодействие, при котором одна особь оставляет информацию на ребрах графа, а другие используют эту информацию позже, когда в неё попадают. Каждый муравей имеет свой цвет. При своём движении муравей метит путь феромоном и своим цветом, и эта информация используется другими муравьями для выбора пути. Концентрация феромона на пути определяет предпочтительность движения по нему.

Процесс поиска решений итерационный. Каждая итерация включает три этапа. На первом этапе каждый муравей находит решение (строит маршрут), на втором этапе откладывает феромон и цвет, на третьем этапе осуществляется испарения феромона. Для хранения информации о цвете используется индикатор цвета - вектор $D_j=\{d_{jl}|l=1,2,\dots,n\}$, формируемый в процессе работы алгоритма, где $d_{jl}=1$, если по ребру e_j проходил муравей типа (цвета) l . Изначально $d_{jl}=0$. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромон и цвет откладываются агентом на ребрах после полного формирования решения.

На первом этапе каждой итерации каждый l -ый муравей формирует свой собственный маршрут sl .

При последовательном подходе первый этап осуществляется следующим образом. Процесс построения маршрута sl пошаговый. На каждом шаге агент применяет вероятностное правило выбора следующей вершины для включения ее формируемый маршрут sl . Пусть к данному моменту сформирован маршрут $sl(t)$, Агент определяет множество ребер $Kl(t+1)$ кандидатов для включения в свой маршрут. $Kl(t+1)$ формируется по методике, рассмотренной выше. Для каждого ребра $e_j \in Kl(t+1)$ определяется количество размещенного на нем

феромона - h_j . По индикатору цвета на ребре e_j определяется число типов муравьев $u_j = \sum_l d_{jl}$, прошедших через ребро e_j . Рассчитывается параметр $w_j = (u_j - 1) \cdot l_j$, где l_j – длина ребра e_j .

По формуле (3.1) – при мультипликативной свертке, либо по формуле (3.2) – при аддитивной свертке определяется потенциальная стоимость C_j каждого ребра $e_j \in Kl(t+1)$.

$$C_j = (h_j)^\alpha \cdot (w_j)^\beta \quad (3.1)$$

$$C_j = \alpha \cdot (h_j) + \beta \cdot (w_j), \quad (3.2)$$

где α, β - управляющие параметры, которые подбираются экспериментально.

Вероятность P_j включения ребра $e_j \in Kl(t+1)$ в формируемый маршрут $sl(t)$, определяется следующим соотношением

$$P_j = C_j / \sum_{j|e_j \in Kl(t+1)} C_j \quad \text{для } (j|e_j \in Kl(t+1)). \quad (3.3)$$

Агент с вероятностью P_j выбирает одно из ребер $e_j \in Kl(t+1)$, которое включается в маршрут $sl(t)$, маршрут уже обозначается как $sl(t+1)$. На конечном шаге $t=nl$ l -м агентом будет сформирован s -маршрут sl .

При комбинаторном подходе l - агентом на каждой итерации не строится, а выбирается из набора $Sl = \{slz|z=1,2,\dots,m\}$ маршрут sl . Пусть Elz – множество ребер графа G , входящих в slz .

Вероятность Plz выбора маршрута slz , определяется следующим соотношением

$$Plz = Clz / \sum_{z|slz \in Elz} C_{lz}, \quad (3.4)$$

где Clz потенциальная стоимость маршрута slz , $Clz = \sum_{j|e_j \in Elz} C_j$ для $(j|e_j \in Elz)$.

Показатель w_j используется исходя из следующих соображений. Рассмотрим два маршрута $s1$ и $s2$. Пусть $l1$ длина маршрута $s1$, а $l2$ длина маршрута $s2$. В худшем случае общая длина сети $l = l1 + l2$. Если же маршруты $s1$ и $s2$ используют общие ребра, то общая длина сети будет меньше на суммарную длину общих ребер. Таким образом, чем больше общих ребер используемых множеством маршрутов и их суммарная длина, тем меньше общая длина сети.

После формирования за n шагов муравьями маршрутов (каждый муравей - свой маршрут) определяется длина сети (дерева Штейнера), построенного муравьями F_l .

На втором этапе итерации, каждый l -ый муравей метит путь феромоном и своим цветом, т.е. откладывает феромон и заносит информацию в индикаторы цвета на каждом ребре построенного им маршрута.

Количество феромона Δt_l , откладываемое l -ым муравьем на каждом ребре построенного им маршрута sl , определяется следующим соотношением

$$\Delta t_l = Q \setminus F, \quad (3.5)$$

где Q – базовое количество феромона, откладываемое l -ым муравьем на ребрах маршрута.

После того, как каждый агент сформировал решение и отложил феромон, на третьем этапе происходит общее испарение феромона на ребрах полного графа G в соответствии с формулой (3.6).

$$h_j = h_j \cdot (1 - \rho), \quad (3.6)$$

где ρ – коэффициент обновления (0.93–0.99).

После выполнения всех действий на итерации находится длина F сети, построенной агентами на данной итерации, которая запоминается. Далее осуществляется переход на следующую итерацию. Временная

сложность этого алгоритма зависит от времени жизни колонии l (число итераций), количества вершин графа n и числа муравьев u , и определяется как $O(l \cdot n^2 \cdot u)$. В работе исследовалась комбинация подходов. На первом этапе на нескольких итерациях (не более 10) поиск решения осуществлялся на основе комбинаторного подхода. На последующих итерациях поиск решения осуществлялся на основе последовательного подхода. Проанализировав экспериментальные исследования можно сделать следующие выводы.

По сравнению с существующими алгоритмами достигнуто улучшение результатов на 8-12%. В среднем запуск программы обеспечивают нахождения решения, отличающегося от оптимального менее чем на 0,5%, т.е. пяти запусков программы со случайными начальными расположениями цепей было достаточно для нахождения оптимума. Перспективными путями улучшения муравьиных алгоритмов являются различные адаптации параметров с использованием базы нечетких правил и их гибридизация с генетическими алгоритмами. Как вариант, такая гибридизация может состоять в обмене через определённые промежутки времени текущими наилучшими решениями.

Литература

- 1 Курейчик, В.В. Концепция эволюционных вычислений, инспирированных природными системами [Текст] / В.М. Курейчик, С.И. Родзин // Известия ТРТУ. Тематический выпуск "Интеллектуальные САПР". - Таганрог: Изд-во ТТИ ЮФУ, 2009. -№4(93). -С. 16-25.
- 2 Курейчик, В.В. Роевой алгоритм в задачах оптимизации [Текст] / В.В. Курейчик Д.Ю. Запорожец // Известия ЮФУ. Технические науки. – Таганрог: Изд-во ТТИ ЮФУ, 2010. –№7 (108). –С. 28–32.
- 3 Лебедев, В.Б. Размещение с учетом критериев трассируемости [Текст] / В.Б. Лебедев, С.А. Степаненко // Известия ТРТУ. Тематический выпуск "Интеллектуальные САПР". -Таганрог: Изд-во ТРТУ, 2006. -№8(63). -С. 287-289.
- 4 Лебедев, О.Б. Трассировка в канале методом муравьиной колонии [Текст] / О.Б. Лебедев // Известия ЮФУ. Технические науки. Тематический выпуск «Интеллектуальные САПР». – Таганрог: Изд-во ТТИ ЮФУ, 2009. – № 4 (93). – С. 46 – 52.
- 5 Лебедев, Б.К. Глобальная трассировка на основе роевого интеллекта [Текст] / Б.К. Лебедев, В.Б. Лебедев // Известия ЮФУ. Технические науки. – Таганрог: Изд-во ТТИ ЮФУ, 2010. –№7 (108). –С. 32–39.
- 6 Курейчик, В.В. Обзор эволюционных методов оптимизации на основе роевого интеллекта [Текст] / В.В. Курейчик, А.А. Полупанов // Известия ЮФУ. Технические науки. – Таганрог: Изд-во ТТИ ЮФУ, 2011. – №7 (120). –С. 7–12.
- 7 Лебедев, Б.К. Процедуры канальной трассировки на основе гибридизации роевого интеллекта с генетическим поиском [Текст] / Б.К. Лебедев, В.Б. Лебедев // Сб. трудов IV Всероссийской научно-технической конференции «Проблемы разработки перспективных микро- и наноэлектронных систем – 2010 (МЭС-2010)». – М.: ИППМ РАН, 2010. – С. 214 – 221.
- 8 Тарасов, В.Б. Эволюционное проектирование: на границе между проектированием и самоорганизацией [Текст] / В.Б. Тарасов, А.В. Голубин // Известия ТРТУ. Тематический выпуск «Интеллектуальные САПР». – 2006. –№8(63). – С.77 – 82.
- 9 Лебедев, Б.К. Интеллектуальные системы [Текст] / Б.К. Лебедев, О.Б. Лебедев и др. // Таганрог: Изд-во ТТИ ЮФУ, 2008. – 195 с.
- 10 Кажаров, А.А. О некоторых модификациях муравьиного алгоритма [Текст] / А.А. Кажаров, В.М. Курейчик // Известия ЮФУ. Технические науки. Тематический выпуск "Интеллектуальные САПР". -Таганрог: Изд-во ТТИ ЮФУ, 2008. -№4(81). С. 7-12.
- 11 Лебедев, Б.К. Эволюционные механизмы трассировки в канале [Текст] / Б.К. Лебедев, В.Б. Лебедев, О.Б. Лебедев // Известия ЮФУ. Технические науки. Тематический выпуск "Интеллектуальные САПР". - Таганрог: Изд-во ТТИ ЮФУ, 2008. -№9(86). С. 92-100.
- 12 Курейчик, В.М. Использование роевого интеллекта в решении NP-трудных задач [Текст] / В.М. Курейчик, А.А. Кажаров // Известия ЮФУ. Технические науки. –Таганрог: Изд-во ТТИ ЮФУ, 2011. –№7(120). –С. 30–37.
- 13 Лебедев, Б.К. Интеллектуальная процедура построения дерева Штейнера на основе процедур отсечки и сужения [Текст] / Б.К. Лебедев // Известия ТРТУ. –Таганрог: Изд-во ТРТУ, 2000. -№ 1. - С. 89 – 94.
- 14 Лебедев, Б.К. Эволюционные процедуры решения комбинаторных задач на графах [Текст] / Б.К. Лебедев, О.Б. Лебедев // Труды десятой национальной конференции по искусственному интеллекту с международным участием. -М.: Физматлит, 2006. –С. 416-424.
- 15 Курейчик, В.В. Композитные методы разбиения графовых моделей [Текст] / В.В. Курейчик, П.В. Сороколетов. –Таганрог: Изд-во ТРТУ, 2006. -140с.

Лебедев Б.К., Лебедев О.Б., Лебедева Е.М. "Использование метода муравьиной колонии для построения кратчайших связывающих сетей". В работе представлен алгоритм построения кратчайших связывающих сетей на основе метода муравьиной колонии. При разработке алгоритма были предложены новые технологии, принципы и механизмы решения задачи построения кратчайших связывающих сетей, основанные на моделировании процессов адаптивного поведения муравьиной колонии. В соответствии со спецификой поставленной задачи были объединены в целое математические методы, в которых заложены принципы природных механизмов принятия решений. Проведены экспериментальные исследования и получены результаты, подтверждающие в целом теоретические разработки.

Ключевые слова: Муравьиная колония, кратчайший путь, сеть, метод, СБИС.

Lebedev B.K., Lebedev O.B., Lebedev E.M. "The use of ant colony to build shortest connecting networks." This paper presents an algorithm for constructing shortest connecting networks using the method of ant colony. In the development of the algorithm proposed new technology, principles and mechanisms of solving the problem of constructing shortest connecting networks, based on the modeling of processes of adaptive behavior of an ant colony. In accordance with the specifics of the problem were integrated into the whole mathematical methods, which lays down the principles of natural decision-making mechanisms. Experimental studies and obtained results confirming the overall theoretical developments.

Keywords: ant colony, the shortest path network, method, VLSI.

Разработка виртуальной лаборатории для изучения и моделирования архитектур процессорных элементов

Р.В. Мальчева, О.А. Авксентьева
Донецкий национальный технический университет
raisa_malcheva@donntu.org

Мальчева Р.В., Авксентьева О.А. Разработка виртуальной лаборатории для изучения и моделирования архитектур процессорных элементов. Рассмотрены основные функциональные элементы виртуальной лаборатории для изучения принципов работы и основ проектирования процессорных компонентов. Даны примеры разработки моделей различных уровней.

Ключевые слова: функциональная модель, демонстрация, Flash-технология, проектирование, моделирование.

Введение

Глубокое понимание принципов построения и функционирования сложных систем (каковыми являются устройства микропроцессорной техники) достигается лишь при практической работе с такими системами или их моделями. Казалось бы, при сравнительной дешевизне и широком распространении устройств микропроцессорной техники достаточно легко создать стенды для изучения принципов функционирования этих средств вычислительной и управляющей техники. Однако отсутствие доступа к большинству компонентов микропроцессорных устройств (регистров, сумматоров, вентиляемых схем и т. п.), расположенных внутри одной или нескольких БИС, приводит к необходимости использования для этих целей специально разработанных моделей: физических или функциональных [1].

Физическая модель – это стенд со структурой изучаемого устройства микропроцессорной техники, построенный на интегральных схемах малой и средней степени интеграции и снабженный средствами для управления режимами работы исследуемого устройства, а также индикации состояний любых его компонентов. В связи с быстрой эволюцией средств микропроцессорной техники массовый выпуск таких специализированных стендов нецелесообразен. Экономически невыгодно и их мелкосерийное производство.

Функциональная модель – это микроЭВМ, в которую введена программа, имитирующая работу исследуемого устройства микропроцессорной техники под воздействием приказов, подаваемых со специально разработанного пульта управления или, например, со штатной клавиатуры микроЭВМ. Функциональные модели чрезвычайно универсальны: одни и те же технические средства позволяют производить изучение процесса преобразования информации в любых цифровых устройствах и с любой степенью детализации. Для исследования нового устройства надо только сменить моделирующую программу, программу вывода на экран изображения структуры моделируемого устройства и программу отображения состояния компонентов этого устройства. Основным недостатком функциональных моделей – невозможность исследования работы оригинала при реальных входных сигналах, нагрузках и помехах – присущ и физическим моделям, так как последние строятся на других (по сравнению с оригиналом) элементах, в другом конструктивном исполнении и к ним подключаются лишние устройства (схемы индикации состояний компонентов оригинала). В соответствии со своим назначением функциональная модель должна правильно и полно отражать состояния лишь индицируемых компонентов оригинала и только в индицируемые моменты времени [1].

Следовательно, при создании функциональной модели надо в первую очередь заботиться о способах ввода управляющих сигналов и вывода информации о состояниях моделируемого устройства, т. е. о факторах, определяющих необходимый уровень и полноту исследования. Та же часть программного обеспечения модели, с помощью которой анализируются текущее ее состояние, наличие сигналов, инициирующих переход в следующее состояние, и определяется новое содержимое индицируемых компонентов оригинала, может быть построена любым удобным (или доступным) для разработчика способом.

Анализ технического и методического обеспечения

Для обеспечения лабораторного практикума по дисциплинам, связанным с изучением архитектуры компьютеров, используется [1] два пути:

- 1) создание учебных лабораторных комплексов на микропроцессорных комплектах;
- 2) создание стендов для моделирования ЭВМ и любых цифровых устройств.

Первый путь позволял проводить физическое исследование микроЭВМ и цифровых устройств, но требовал периодической замены лабораторной базы. Так с 1983 года, когда были введены в строй учебные лабораторные комплексы на микропроцессорных комплектах K584, K589, K580, было создано несколько новых лабораторных комплексов и истрачено достаточно много средств. Кроме того, эти комплексы хороши для обучения специалистов, бакалавров и магистров по направлению подготовки «Информатика и вычислительная техника», но сложны и избыточны для обучения студентов других специальностей и направлений подготовки.

Второй путь предполагал создание устройства отображения информации и программной модели, исследуемой ЭВМ или цифрового устройства, реализуемой на любой ЭВМ. Такой моделью стала персональная ЭВМ, на которой легко изменять как структуру, так и программу изучения функционирования ЭВМ и (или) любого цифрового устройства [1].

Такой подход – проектирование виртуальных ЭВМ для изучения с их помощью архитектуры компьютеров, с одной стороны, и использование самого процесса разработки виртуальных ЭВМ как средство обучения, с другой стороны, применяется на кафедре «Компьютерная инженерия» уже более 30 лет. Однако с появлением новых технологий изменились презентационные свойства моделей [2].

Разработка демонстрационных роликов для пояснения выполнения отдельных операций

Современные мультимедийные средства существенно облегчают процесс обучения через реализацию одного из фундаментальных его методов – наглядности. Наиболее удобной для создания демонстраций является технология Flash, которая поддерживает не только все презентационные возможности Microsoft Power Point, но и позволяет создавать динамические демонстрации, т.е. демонстрировать работу некоторого процесса шаг за шагом с заданной частотой смены кадра [3].

Как показывает практика подготовки специалистов направления компьютерной инженерии, особое затруднение у студентов вызывает понимание машинного (внутреннего) представления данных и их обработки. Для повышения эффективности преподавания дисциплин, связанных с изучением основ информатики и особенностей выполнения машинных команд (в кодах или на ассемблере), был подготовлен значительный объем демонстрационных роликов [4]. В качестве примера на рис.1 приведены слайды демонстрационного ролика выполнения операции «циклический сдвиг вправо на 1 разряд».

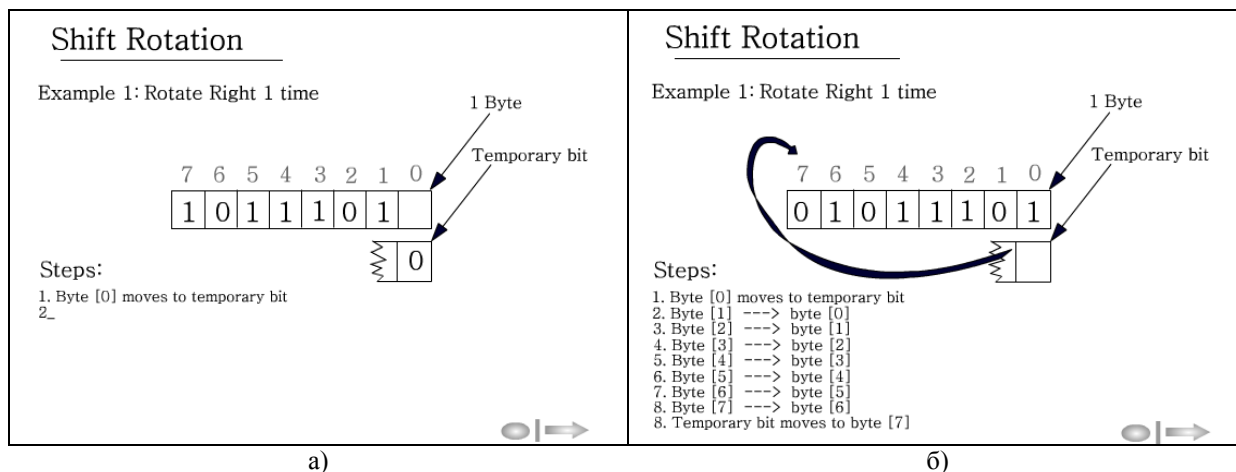


Рисунок 1- Фрагменты демонстрационного ролика выполнения операции «циклический сдвиг вправо на 1 бит»: а) результат 1-го шага; б) результат выполнения операции

Анимационные ролики используются не только во время чтения лекций. Разработана Web-версия учебных дисциплин, со страниц которой студенты могут просматривать демонстрационные ролики, а также проходить самотестирование изученного материала.

Создание Flash-фильма для демонстрации работы виртуальной машины

На кафедре «Компьютерная инженерия» ДонНТУ, так же как и в других университетах [1, 5], в качестве заданий по курсовому проектированию применяется разработка студентами виртуальных компьютеров. При этом используются упрощенные модели наиболее популярных промышленных процессоров.

Для ознакомления студентов с принципом работы одно- двух- или трехадресной (рис.2) виртуальной машины созданы специальные Flash-фильмы.

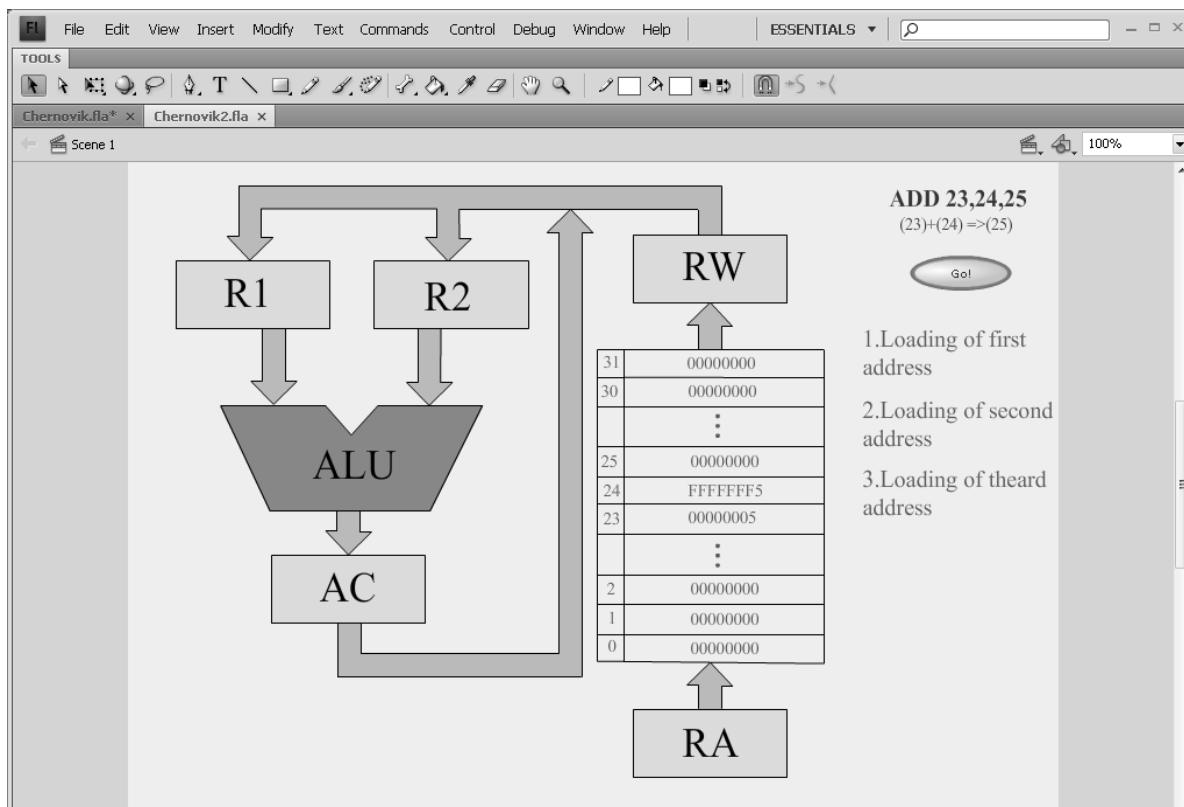


Рисунок 2 – Вид структурной схемы трехадресной машины при выполнении операции сложения

Проектирование устройств на языках VHDL и Verilog

Более детальное проектирование составных элементов виртуальных компьютеров выполняется с использованием таких языков описания аппаратуры, как HDL, Verilog и др. [6-9].

Модели, созданные на языках проектирования VHDL или Verilog, можно не только запускать в средах проектирования, но и загрузить в FPGA плату и подключить к нему устройства ввода и вывода. Существует множество отладочных FPGA стендов, которые уже имеют встроенные устройства ввода, такие, как кнопки и переключатели; и вывода – световые индикаторы, экран. Также в таких стендах имеются различные порты ввода/вывода, к которым можно подключить периферийные устройства.

Рассмотрим пример проектирования устройства умножения операндов, заданных в ПН («положительном нуле») [10, 11]. Оно реализовано в виде VHDL модели и состоит из блока умножения, элемента И-2-ИЛИ, D-триггера, DC-триггера (рис.3).

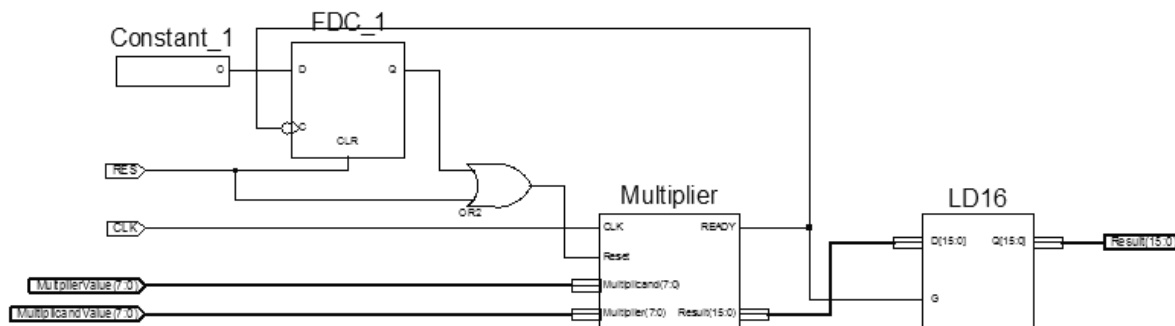


Рисунок 3 - VHDL модель устройства умножения

При нажатии кнопки Reset, логический «0» поступает на DC триггер и на блок умножения (рис.4), происходит обнуления регистра AC, записывается множимое и множитель, происходит умножение. D-триггер срабатывает по сигналу ready блока умножения, он необходим для буферизации данных.

Блок умножения состоит из:

- MC - 8 разрядный регистр множимого. Загрузка множимого осуществляется по команде Cload_mc с управляющего автомата.
- MR - 8 разрядный регистр множителя. По команде Cload_mr в регистр загружается множитель, по команде Cshift_right_mr происходит логический сдвиг вправо. Нулевой разряд является признаком коррекции произведения.

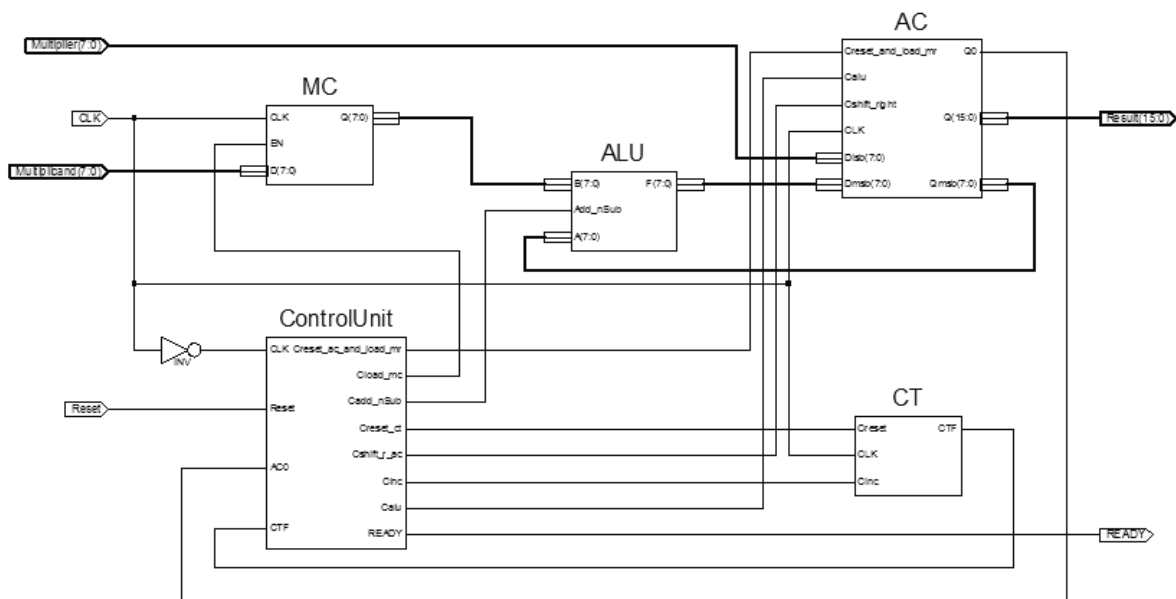


Рисунок 4 – VHDL модель блока умножения

- AC - 16 разрядный регистр результата. Имеет логический сдвигатель вправо (Cshift_right_ac), команду сброса (Creset), команду загрузки результата с АЛУ (Calu).
- ALU - 8 разрядное арифметико-логическое устройство, содержит сумматор (Cadd/sub := 1) и вычитатель (Cadd/sub := 0).
- CT - счетчик, предназначенный для подсчета операций. Имеет признак переполнения CTF. Команда Creset_ct сбрасывает счетчик в исходное состояние, а Cinc - инкрементирует содержимое счетчика.
- Control Unit - управляющий автомат.

VHDL модель MC:

```

process (CLK)
begin
if (rising_edge(CLK)) then           //если восходящий вход CLK
  if (EN='1') then                  //и если сигнал Enable
    Q <= D;                          //то записать данные.
  end if;
end if;
end process;

```

Структура блока АЛУ представлена на рис.5.

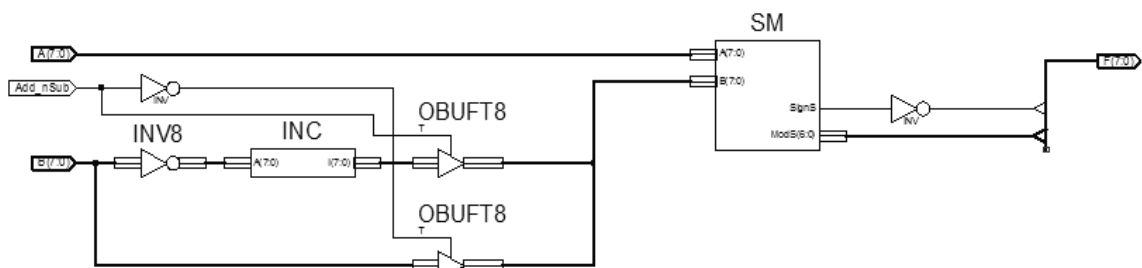


Рисунок 5 – VHDL модель блока АЛУ

Блок АЛУ включает в себя:

- инкрементор:

```

process (A)
begin
  I <= A+1;
end process;

```


- инвертор;
- сумматор:

```
process (A,B)
  variable S : STD_LOGIC_VECTOR (7 downto 0);
begin
  S := A+B;
  ModS <= S(6 downto 0);
  SignS <= S(7);
end process;
```

Для произведения используется обычный сумматор, а после инвертируется знаковый разряд. Для выполнения операции вычитания необходимо поменять знак у второго операнда (B) и провести сложение. Смена знака производится путем инверсии операнда, а затем инкрементации полученного значения. Счетчик реализован в виде VHDL модели, включает в себя 4-х разрядный счетчик и элемент 3И и реализует следующую сущность:

```
entity CB4CE_HXILINX_CT is
port (
  Q0 : out STD_LOGIC; Q1 : out STD_LOGIC;
  Q2 : out STD_LOGIC; Q3 : out STD_LOGIC;
  C : in STD_LOGIC; CE : in STD_LOGIC;
  CLR : in STD_LOGIC
);
end CB4CE_HXILINX_CT;
```

Реализация счетчика выглядит следующим образом:

```
architecture Behavioral of CB4CE_HXILINX_CT is
  signal COUNT : STD_LOGIC_VECTOR(3 downto 0) := (others => '0');
  constant TERMINAL_COUNT : STD_LOGIC_VECTOR(3 downto 0) := (others => '1');
begin
  process(C, CLR)
  begin
    if (CLR='1') then
      COUNT <= (others => '0');
    elsif (C'event and C = '1') then
      if (CE='1') then
        COUNT <= COUNT+1;
      end if;
    end if;
  end process;
  Q3 <= COUNT(3); Q2 <= COUNT(2);
  Q1 <= COUNT(1); Q0 <= COUNT(0);
end Behavioral;
```

По сигналу CRL значение счетчика сбрасывается в исходное состояние. Нарастивание счетчика происходит по фронту сигнала CE. Для данной задачи необходим 3-х разрядный счетчик, поэтому сигнал переполнения добавляется логическим элементом 3И.

Регистр АС реализован в виде следующего кода:

```
process (CLK,RESULT)
begin
  if (rising_edge(CLK)) then
    if (Creset_and_load_mr = '1') then
      RESULT(15 downto 8) <= "10000000"; //Сброс старшей части
      RESULT(7 downto 0) <= Dlsb; //Записываем множимое в младшую часть
    end if;
    if (Calu = '1') then
      RESULT(15 downto 8) <= Dmsb;
    end if;
    if (Cshift_right = '1') then //логический сдвиг
      RESULT(15) <= RESULT(15);
      RESULT(14) <= not (RESULT(15));
      RESULT(13 downto 0) <= RESULT(14 downto 1);
    end if;
  end if;
  Qmsb <= RESULT(15 downto 8); //Множимое
  Q0 <= RESULT(0); //Младший бит
  Q <= RESULT; // Запись результата
end process;
```

Control Unit представлен в виде VHDL-модели и реализует алгоритм операции умножения в положительном нуле. В общем виде модель выглядит следующим образом:

```

process (CLK,Reset)
begin
if (Reset = '1') then //Сигнал сброса
state <= RST;
elseif rising edge(CLK) then //Если фронт CLK
case state is //Проверка текущего состояния
when RST => ... //состояние сброса
when BIT_CHECK => ... //проверка нулевого бита множителя
when SHIFT => ... //сдвиг регистра AC
when CTF_AND_SIGNBIT_CHECK => ... //проверка переполнения и проверка знакового бита,
после переполнения
when LAST_SHIFT => ... //сдвиг, если знаковый бит равен «1»
when RDY => ... //состояние готовности
end case;
end if;
end process;

```

Временная диаграмма умножения чисел «-6» и «-2» в положительном нуле представлена на рис.6. В multiplicand [7:0] – записано множимое = -6. В multiplier [7:0] – записан множитель = -2. В result [15:0] записывается результат умножения = 12. Далее на диаграмме изображены управляющие сигналы.

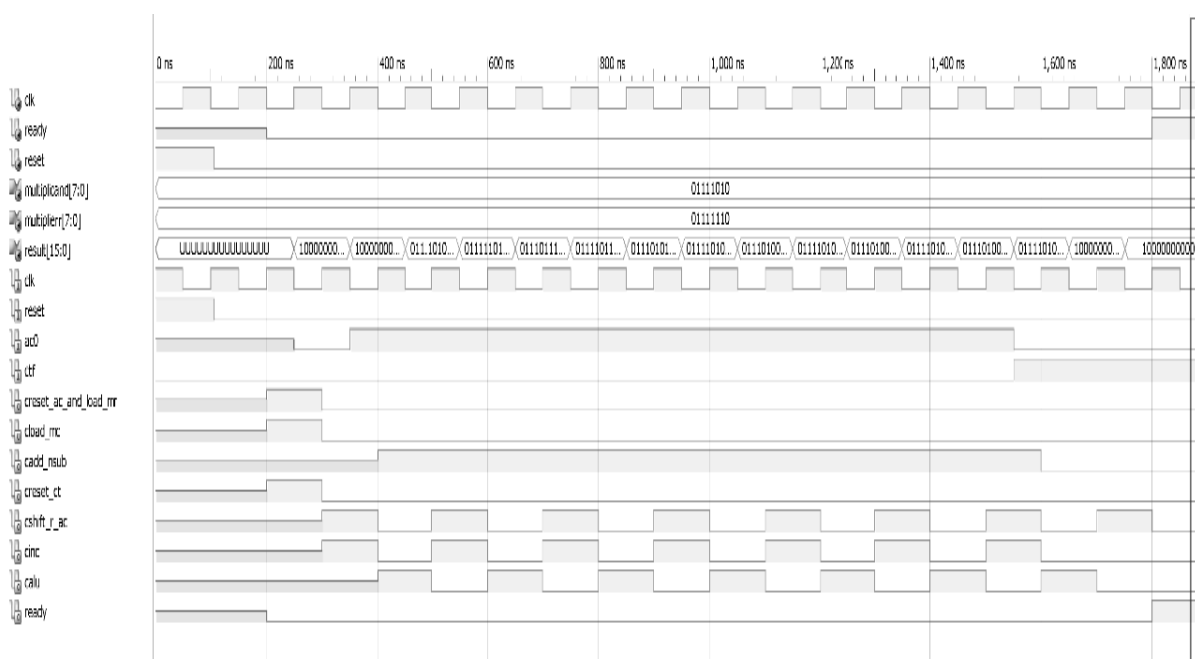


Рисунок 6 – Временная диаграмма умножения чисел в ПН

Выводы

Для технического обеспечения дисциплин, связанных с изучением архитектуры компьютеров, необходимо применять широкий спектр демонстрационных и обучающих средств. Для их создания используются различные современные технологии, такие как Power Point и Flash. Для изучения основ проектирования элементов ЭВМ и выполнения, соответствующих курсовых и дипломных проектов необходимо использовать языки описания аппаратуры, такие как HDL, Verilog и др. Особое место в процессе постановки учебного процесса занимает создание простых гипотетических машин, обладающих типичными чертами многих конкретных ЭВМ. Знание принципов построения и функционирования такой ЭВМ является хорошей базой для освоения процессорных систем любых типов и моделей.

Литература

1. Кириллов В.В. Архитектура базовой ЭВМ. – СПб: СПбГУ ИТМО, 2010. – 144 с.
2. Malcheva R. Applying Internet technologies to improve the perception of lectures // Proceedings of 3d Congress EE. - Glasgow, 2002. - PP.348-349.
3. Macromedia Flash MX 2004 ActionScript 2.0. Справочник разработчика: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. - 896 с.

4. Мальчева Р.В. Использование видеороликов и игр при обучении иностранных студентов направления "Компьютерная инженерия" // Збірка праць V науково-методичної конференції "Проблеми і шляхи вдосконалення науково-методичної та навчально-виховної роботи в ДонНТУ". – Донецьк: ДонНТУ, 2013.

5. Baranov S. Digital System Design. In book "Design of Digital Systems and Devices". Series: Lecture Notes in Electrical Engineering. – Springer-Verlag Berlin Heidelberg, Vol. 79, 2011. – P. 3- 41.

6. Rawski M., Tomaszewicz P., Borowik G. Logic Synthesis Method of Digital Circuits Designed for Implementation with Embedded Memory Blocks of FPGAs. In book "Design of Digital Systems and Devices". Series: Lecture Notes in Electrical Engineering. - Springer-Verlag Berlin Heidelberg, Vol.79, 2011.– P.121-144.

7. Хаханов В.И., Хаханова И.В., Литвинова Е.И., Гузь О.А.. Проектирование и верификация цифровых систем на кристаллах. Verilog & System Verilog. – Харьков: ХНУРЭ, 2010. – 528 с.

8. Malcheva R., Naaem H. Development of the Data Transferring System Using SoC // European Scientific Journal, 2014. - Vol.10, N7. – PP.168-172.

9. Grout I. Digital Systems Design with FPGAs and CPLDs. – Amsterdam: Elsevier, 2008. – 406 pp.

10. Денисов Д.В., Артюхин В.А., Седенков М. Ф. - Аппаратное обеспечение вычислительных систем. Под ред. Д.В. Денисова. – М.: Маркет ДС, 2007. – 184 с.

11. Базарова С. Б-М., Чемерисюк А.С., Тулохонов Э.А., Гомбоев Е. Ш., Варфоломеев А. В. – Выполнение арифметических операций в АЛУ для чисел с фиксированной запятой. Часть I: Практическое пособие. –Улан-Удэ: ВСГТУ, 2006. – 77 с.

Мальчева Р.В., Авксентьева О.А. Разработка виртуальной лаборатории для изучения и моделирования архитектур процессорных элементов. Рассмотрены основные функциональные элементы виртуальной лаборатории для изучения принципов работы и основ проектирования процессорных компонентов. Даны примеры разработки моделей различных уровней.

Ключевые слова: функциональная модель, демонстрация, Flash-технология, проектирование, моделирование.

Malcheva R., Avksentieva O. Development of virtual laboratory to study and simulate of processor elements' architectures. The main functional elements of a virtual laboratory to study of the processing in processor's components and foundations of their developing are discussed. Examples of developing of the simulation models of different levels are presented.

Keywords: functional model, demonstration, Flash-technology, developing, simulation.

Формальные грамматики в семиотической концептуальной модели предметной области

А.В. Григорьев

Донецкий национальный технический университет
grigorievalvl@gmail.com

Григорьев А.В. Формальные грамматики в семиотической концептуальной модели предметной области. Рассмотрен комплекс методов работы с формальными грамматиками, определенных в рамках концептуальной модели предметной области инструментальной оболочки для автоматизации построения интеллектуальных САПР ограниченного класса. Работа носит итоговый характер.

Ключевые слова: *формальные грамматики, концептуальная модель предметной области, семиотическая модель, САПР.*

Введение

САПР относятся к важнейшим информационным системам. Развитие производства и общества в целом не мыслимо без применения САПР. Лавинообразное увеличение числа предметных областей (ПрО), где требуется создание САПР, ставит проблему автоматизации их создания. Методы и средства систем искусственного интеллекта (СИИ), проникая, в том числе, и в область создания САПР, позволяют сделать решение этой проблемы реальностью. Данная тенденция, к сожалению, пока не получила своего полного воплощения на практике. Т.о., *актуальным* является построение унифицированных средств и методов построения САПР, адаптируемых на технологии проектирования в требуемой ПрО, обеспечивающих более высокую эффективность процесса создания и функционирования как новых интеллектуальных САПР (И САПР), так и модификации существующих САПР до уровня гибридных. Проведенный ранее анализ [] тенденций развития САПР и СИИ в современных условиях позволяет сделать вывод, что *актуальной становится задача* разработки комплексной КМ ПрО САПР как основы для создания специализированной инструментальной оболочки (ИО), предназначенной для построения интеллектуальных САПР уровня АРМ, учитывающей тенденции развития САПР и СИИ и являющейся средством предметной и проблемной адаптации в современных условиях.

При этом данная КМ должна обеспечивать следующие возможности предметной и проблемной адаптации: 1) ИО должна быть способна настраиваться на любую техническую или не техническую ПрО, обладающей достаточно формализованным описанием объекта и методами построения объекта с требуемыми характеристиками; 2) Быть ориентирована на эксперта в ПрО, выполняющего адаптацию САПР, как источник знаний при построении моделей объекта проектирования и проектных процедур и быть способной адаптироваться на специфический уровень его квалификации; 3) Адаптироваться на данную ПрО с характерными для нее типами фазовых переменных и координат взаимодействия; 4) Быть ориентированной на построение набора требуемых типов изделий в данной ПрО; 5) Формировать состав модельных АУ требуемой полноты; 6) В рамках каждого АУ формировать набор моделей, проектных процедур и критериев требуемой полноты; 7) ИО должна при построении САПР на каждом АУ, задавая набор процедур, моделей и критериев, обеспечить способность задать иерархические, регулярные структуры, реализующие регулярные же, иерархически организованные функции, специфичные для данной ПрО; 8) Иметь возможность адаптироваться на достигнутый уровень воплощения методик проектирования для существующих САПР в данной ПрО, т.е. – быть ориентирована либо на построение нового автономного интеллектуального САПР (И САПР) в случае отсутствия П САПР, либо – интеллектуальной надстройки над существующим САПР, добавив в нее те или иные проектные процедуры, повысив тем самым уровень воплощения методик проектирования; 9) ИО должна обеспечивать построение либо всех процедур САПР целиком в форме ЭС, либо - построение САПР как гибридной.

Построение данной КМ выполнено автором в рамках работ [1-30]. Данная КМ является основой построения соответствующего инструментального комплекса по автоматизации построения интеллектуальных САПР некоторого ограниченного класса – мета-эвристической оболочки (МЭО).

Наибольшее влияние на методы построения КМ оказала выбранная форма ее представления – семиотическая модель. А так как главный компонент СМ – это формальные грамматики, то без преувеличения можно сказать, что суть КМ – это набор методов работы с формальными грамматиками, определенными в рамках СМ.

Цель работы: описать структура средств и методов работы с формальными грамматиками в рамках С КМ ПрО МЭО.

1. Общие принципы построения С КМ ПрОб МЭО

Ранее в работах [1,2] был определен ряд принципов построения инструментальной мета-эвристической оболочки (МЭО), предназначенной для построения интеллектуальных САПР. К главным особенностям МЭО относятся:

- 1) работа с моделями [3,4,5] сложных объектов различной степени недоопределенности.
- 2) ориентация при обучении базы знаний комплекса на ограниченное количество имеющихся в наличии апробированных на практике моделей сложных объектов данной предметной области;

Ограниченное число моделей сложных объектов предполагает необходимость автоматизации синтеза возможных решений с различной степенью их определенности с последующим отсевом выбором имеющих смысл решений с целью ускорить их ввод в базу знаний САПР.

Структура МЭО и технология работы с МЭО проиллюстрированы на рис. 1.

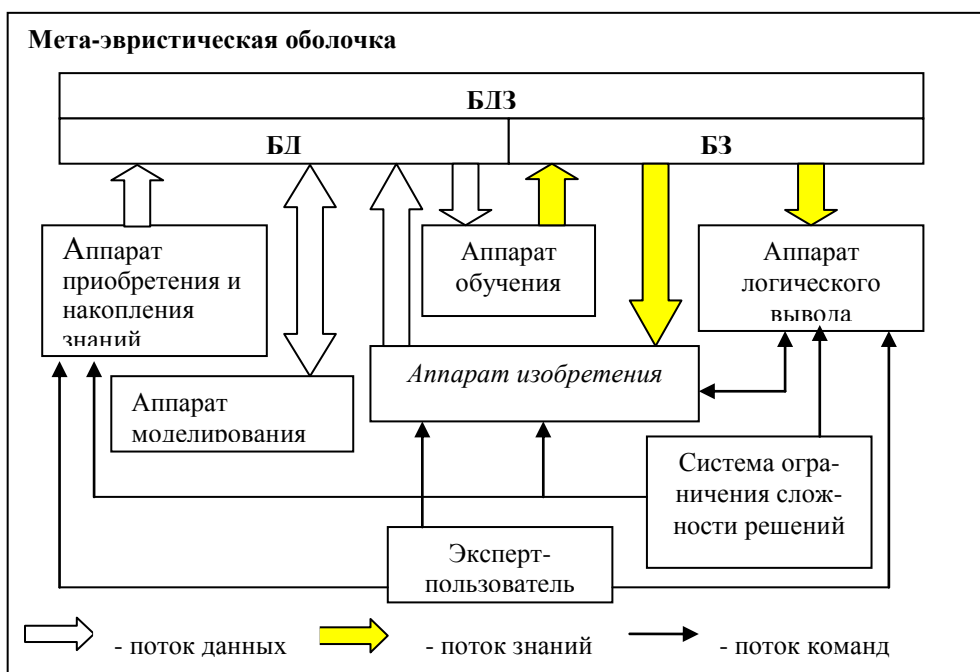


Рисунок 1 - Структура мета-эвристической оболочки

Кратко опишем назначение некоторых элементов комплекса:

1) Аппарат моделирования – обеспечивает нахождение недоопределенной информации предметной области, путем рассмотрения данной задачи на динамических недоопределенных вычислительных моделях [31,32,33].

2) Аппарат изобретения – обеспечивает создание новых объектов некоторого заданного типа.

3) Система оценки и ограничения сложности моделей – оценивает и ограничивает сложность объектов для любых предметных областей (ПрОб) как при вводе апробированных (достоверных) объектов их проблемно-ориентированных САПР или при прямом вводе объектов пользователем, а так же при синтезе (изобретении) объектов. Запрещает рассмотрение не имеющих смысла или слишком сложных для анализа данным пользователем объектов, повышая тем самым скорость процесса изобретения.

Общие принципы построения С КМ ПрОб МЭО:

1. Построение САПР решения типичных задач проектирования, где имеет место большое, но ограниченное множество возможных решений;
2. Физическая семантика ПрОб;
3. И-ИЛИ-дерево как форма контекстно-независимой грамматики описания множества решений структур или функций объекта некоторого типа;
4. Семиотическая модель как форма представления КМ ПрОб;

Рассмотрим данные принципы детальнее.

1.1. Построение САПР решения типичных задач проектирования

Существует ряд определений САПР, что связано со сложностью этого понятия. Примеры определений САПР: как комплекс средств, как совокупность проектных процедур, как набор этапов разработки документации и т.д. [34]. Однако, в качестве главного определения можно рассматривать процедурное определение САПР. В соответствии с ним САПР представляет собой ряд моделей объекта проектирования возрастающих уровней абстракции (АУ), связанных проектными процедурами, модифицирующими данные модели под управлением соответствующих критериев (см. рис. 2).

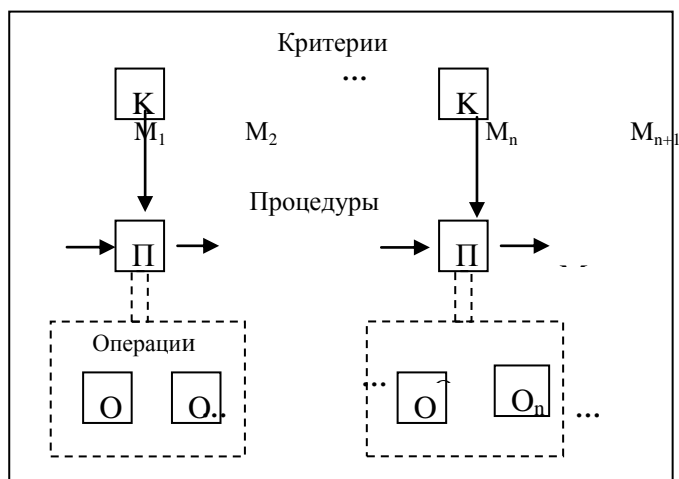


Рисунок 2 - Структура САПР

Совокупность критериев составляет в целом ТЗ на изделие. Набор процедур определяет методику проектирования. Каждая процедура есть комплекс проектных операций. Каждому АУ соответствует некоторое подмножество процедур и связанных с ними форм представления моделей объекта проектирования. Типичные проектные процедуры, относящиеся к любому АУ, это: синтез (выбор) модели, редактирование, верификация (моделирование), документирование. Типичные формы представления моделей объекта проектирования: система уравнений, документ, графический образ, таблица, текст на некотором языке программирования и т.д. В САПР используются такие АУ моделей проектирования и, соответственно, уровней наборов проектных процедур:

- структурный (полное отсутствие модели функций в модели объекта проектирования);
- системный (уровень потоков заявок как модели функций объекта);
- функционально-логический (уровень дискретных, логических уравнений, составляющих модель объекта);
- количественный, включая подуровень макро-моделей, т.е. моделей с сосредоточенными параметрами (уравнений Кирхгофа) и подуровень микро-моделей, т.е. моделей с рассредоточенными параметрами (уровень дифференциальных уравнений в частных производных как моделей объекта).

Перечисленные АУ уровни упорядочены по возрастанию сложности и отличаются:

- полнотой представления модели пространства и времени (наличием или отсутствием, дискретным или непрерывным представлением);
- полнотой отражения совокупности фазовых переменных (поток и потенциал) и координат взаимодействия (емкость, индуктивность и т.д.);
- полнотой состава дополнительных свойств и отношений (функций), отделенных в модели объекта (топологические, функциональные, энергетические и т.д.);
- полнотой представления законов сохранения (энергии, вещества и т.д.), выраженных посредством набора отношений.

На практике это приводит к тому, что САПРы могут решать два класса задач, отличающихся размером пространства поиска решений проектных процедур:

- задачи типичного проектирования, т.е. – выбора подходящего решения из ряда известных;
- задачи поискового конструирования, т.е. - поиска неизвестного ранее решения (изобретения).

1.2. Физическая семантика КМ ПрОб МЭО

Содержательная формулировка КМ ПрОб МЭО, соответствующая физической семантике ПрОб:

- 1) жизненный цикл объекта представляется как совокупность линейных векторных пространств, соответствующих числу необходимых тактов времени;
- 2) линейные векторные пространства связаны набором "временных" связей;
- 3) элементами линейных векторных пространств являются пространственные координаты, связанные совокупностью "пространственных" связей, определенных внутри "временных" связей;
- 4) пространственная координата может рассматриваться как физическая точка (ФТ), в случае, если состав и значения ее свойств отличается от "неопределено";
- 5) свойства ФТ возникают только как факт отражения существования "простых" связей между ФТ внутри "пространственных" связей;
- 6) "собственное" свойство - идентификатор ФТ есть некоторая фазовая переменная данной ПрОб, задающая потенциал точки для некоторой субстанции; задается посредством "кольцевого" отношения над ФТ;
- 7) совокупность "собственных" свойств ФТ определяется уровнем представления модели и ПрОб;
- 8) "чужое" свойство ФТ - задание факта наличия связи - отношения, задающего меру влияния потенциала другой ФТ на потенциал данной ФТ; задается посредством "простой" связи над парой ФТ;
- 9) модель функции ФТ задается табличным образом как множество возможных комбинаций значений свойств-потенциалов ФТ (по всем временным, пространственным и "простым" связям), когда-либо имевшим место на практике;
- 10) отношение зависимости связанной пары потенциалов "свой" - "чужой", выделенное внутри модели функции ФТ, определяет меру влияния "чужого" потенциала на "свой" потенциал ФТ и задает поток, направленный на уравнивание потенциалов во времени; данное отношение задается как функция "простой" связи, определенной внутри "пространственной" связи;
- 11) отсутствие в наборе свойств некоторой ФТ1 "чужого" свойства ФТ2 говорит о невозможности влияния потенциала ФТ2 на потенциал ФТ1 (отсутствующая или однонаправленная связь);
- 12) функция "простой" связи определяется таблично как совокупность историй поведения ее во времени и косвенно задает параметры связи (емкость, сопротивление и т.д.);
- 13) модель функции ФТ строится в соответствии с законами сохранения движения и энергии;
- 14) модели функций ФТ и "простых" связей задают функциональные и энергетические связи, простые" связи задают структурные и причинно-следственные связи, пространственные связи задаются явно (см. выше);
- 15) вещественные связи, связанные с переносом вещества, определяются как изменение ФТ своих пространственных координат и строятся в соответствии с законом сохранения массы;
- 16) переход на новый уровень представления модели предполагает не только декомпозицию избранного набора ФТ (блоков) и свойств на более мелкие, но и увеличение числа этих точек, т.е. изменение (расширение) вышележащих уровней декомпозиции.

1.3. И-ИЛИ-дерево как форма контекстно-независимой грамматики описания множества решений структур или функций объекта некоторого типа;

1. И-ИЛИ-дерево как аппарат представления БЗ

Опишем принятый в среде мета-эвристической оболочки метод представления знаний [1]. Основой данного метода является использование И-ИЛИ-дерева как аппарата представления БЗ.

Базой знаний в системе является:

- И-ИЛИ-дерево, то есть упорядоченное множество вершин по «И» и по «ИЛИ», при этом все вершины имеют свой адрес в И-ИЛИ-дереве;
- Множество семантических двунаправленных зависимостей между ИЛИ-вершинами, описывающими их совместимость между собой, то есть множество правил вывода или продукций.

Представление данных в виде И-ИЛИ-дерева необходимо для выполнения теоретико-множественных операций над БЗ системы в процессе обучения и вывода. Поиск решений в системе ведется по И-ИЛИ-дереву, которое состоит из единиц информации, представленных некоторой структурой.

Под И-ИЛИ-деревом понимается некоторый связный граф, не содержащий циклов и имеющий иерархическую многоуровневую структуру. Вершинам И-ИЛИ-дерева может быть дана следующая трактовка, выполненная с точки зрения терминов теории формальных грамматик [2]. Любая вершина рассматривается либо как терм либо как синтерм. Терм определяется как элементарный терминальный символ множества и включает оригинальный фрагмент описания некоторого объекта(ов) на некотором языке. Термы соединяются между собой только посредством операции "И" (&). Синтерм, т.е. нетерминальный символ, задается как имя множества, которое может в дальнейшем раскладываться. Элементами разложения могут быть как термы, так и синтермы, соединенные посредством операции "И" (&) или "ИЛИ" (V). Синтермы всегда записываются в соответствии с нотацией Бекуса-Наура только в угловых скобках "<>". Очевидно, что термы будут являться только листьями дерева, которые, не могут иметь "сыновей" в данном дереве.

Для избранного подхода к представлению И-ИЛИ-деревьев справедливы следующие утверждения:

- Каждая вершина или узел дерева представляет собой описание фрагмента объекта;
- Каждая полная цепочка от вершины дерева до некоторого листа есть объект, состоящий из фрагментов, которые представлены узлами входящими в данную цепочку;
- Синтермы, имеющие одного «отца», являются альтернативами по отношению к друг другу;
- Отношения между ИЛИ связывает те термы, комбинация которых принадлежат некоторому непустому множеству семантически верных (проверенных) объектов, имеющих место в И-ИЛИ-дереве;
- Прототипы или верифицированные объекты есть основа построения И-ИЛИ-дерева;
- И-ИЛИ-дерево есть средство:
 - компактной записи множества известных объектов;
 - порождения гипотез о возможных новых объектах;
- И-ИЛИ-дерево – это множество синтаксически правильных выражений;
- Продукции определены над И-ИЛИ-деревом и задают правила вывода, которые в совокупности позволяют вычлени из И-ИЛИ-дерева семантически верное подмножество, то есть те же самые аксиомы-объекты.

1.4. СМ как форма представления КМ ПрОб;

Средства построения формальной модели.

Открытый характер базы знаний САПР требует использовать для ее построения семиотическую модель (СМ). СМ представляет собой открытую формальную систему и имеет форму восьмерки [36]:

$F = \langle T, C, A, P, r, b, g, d \rangle$, где

T - множество базовых элементов системы, на которых строятся все выражения в F ;

C - множество правил построения синтаксически правильных формул, определяющих среди всех возможных выражений из базовых элементов те, которые синтаксически правильны;

A - множество аксиом F , образующее подмножество в множестве синтаксически правильных формул, которым априорно присваивается статус истинности;

P - множество правил вывода, или семантические правила, (позволяющие получать из аксиом новые синтаксически правильные формулы, которым можно приписывать статус истинности);

r, b, g, d - правила изменения, соответственно для T, C, A и P .

Для конструктивности семиотической модели требуется реализация следующих классов процедур:

$P1$ - определения принадлежности данного элемента множеству T ;

$P2$ - идентификации различия элементов множества T ;

$P3$ - определения синтаксической корректности элементов, построенных посредством правил C .

Процедуры $P1, P2$ и $P3$ должны быть конструктивными, т.е. завершаться через определенное число шагов.

Конструктивная СМ является разрешимой, если существует конструктивная процедура $P4$, дающая однозначный ответ на вопрос - является ли данный синтаксически корректный элемент семантически верным.

СМ может рассматриваться как форма представления концепции "возможных миров" Крипке. Проблема построения разрешимой СМ в общем случае пока не решена.

2. Общая структура средств и методов работы с формальными грамматиками в рамках СМ

Общая структура включает в себя следующие методы и средства работы с формальными грамматиками в рамках СМ:

1) Методы построения **системы вложенных формальных проблемно-независимых языков спецификаций** на базе физической семантики ПрОб с привлечением аппарата НЕ-факторов;

2) Комплекс **методов задания семантических зависимостей** над контекстно-свободными грамматиками, адаптированные к различным условиям их применения;

3) Комплекс алгоритмов **выполнения теоретико-множественных операций** над грамматиками, адаптированные к различным условиям их применения;

4) **Методы обучения**, т.е. – построения базы знаний, на базе ТМО, адаптированные к различным условиям их применения;

5) Методы организации **логического продукционного вывода** на базе ТМО, адаптированные к различным условиям их применения;

6) Метод **оценки сложности** выполнения теоретико-множественных алгоритмов над грамматиками различных типов;

7) Методы преобразования грамматического описания объекта, имеющего **не допустимую когнитивную сложность** представления, к виду, имеющую допустимую форму представления когнитивной сложности;

8) **Изобретение новых решений** – как инструмент сужения числа аксиом грамматики, построенной путем обобщения прототипов;

9) Построение системы **интерфейсов** «Язык предметной области ↔ Язык формальных спецификаций соответствующего уровня абстракции» с целью обеспечения.

Рассмотрим их детальнее перечисленные компоненты.

3. Методы построения системы вложенных формальных проблемно-независимых языков спецификаций на базе физической семантики ПрОб с привлечением аппарата НЕ-факторов

СМ КМ ПрОб включает 6 обязательных уровней: 1) исходной модели; 2) задания времени как блока и свойства; 3) значений свойства времени и моделей пространств; 4) пространственных точек и их идентификаторов; 5) «простых» свойств и внутренних функций блоков; 6) значений "простых" свойств и "кортежей" функций. Все уровни описываются, исходя из общих аксиом, сформулированных в [10]. Рассмотрим последовательно все перечисленные уровни. Правила преобразования семантически верных выражений (правила **g** для **A**), соответствующих **Ti**-му отношению сигнатуры, будем обозначать как **Gi** и формулировать их по мере описания системы уровней.

3.1. Уровень исходной модели

Исходная модель предмета уровня 1 включает блок с именем, но без внутренней структуры, имеющим единственное недоопределенное свойство без имени и структуры, единственную "круговую" связь, замыкающую блок сам на себя. Зададим состав семантически верных отношений исходного уровня (аксиомы ЛА1-ЛА4), задающих описание глобальной аксиомы - *прототипа*, описывая одновременно синтаксис отношений:

ЛА1. Единственный недоопределенный блок – модель предмета, имеет идентификатор "**П**", но не имеет структуры - $B = \{ P \}$.

ЛА2. Единственное недоопределенное свойство не имеет идентификатора и структуры: $D = \{ Nil^D \}$.

ЛА3. Внешняя граница блока **П**: $G[P] = \{ P.Nil^D \}$, где точка обозначает отношение принадлежности свойства блоку.

ЛА4. Среда или множество *связей* уровня (рис. 1) имеет вид: $L_P = \{ l_1^P \}$. Структура связи: $l_1^P = P.Nil^D \Leftrightarrow P.Nil^D$, где \Leftrightarrow обозначает двунаправленное отношение передачи информации между блоками через свойства их внешних границ. Будем обозначать связь l_1^P как **С0**.

СВЯЗЬ "С0" ПО "NIL"

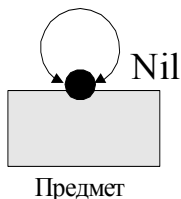


Рисунок 3 - Изначальное представление модели предмета

Принятая выше грамматика описания отношений в ЛА1-ЛА4 входит в *множество синтаксически верных отношений*, составляющих прототип.

ЛА5. Модель прототипа уровня 1 не имеет альтернативных форм представления.

ЛА6. Перечисленный состав блоков, свойств, границ и связей задает набор системообразующих элементов модели данного уровня.

3.2. Уровень задания времени - блока и свойства

Уровень 2 предполагает декомпозицию исходного свойства Nil^D блока **П** на три подсвойства - обратный элемент, время **T** и неопределенность. Одновременно выполняется декомпозиция блока **П** на обратный элемент, блок **Время** (в дальнейшем просто "**B**") и неопределенность (рис. 4). Тут **B** - собственное свойство блока **T**.

Множество связей $L_B = \{l_k^B\}_{k=1}^6$, является декомпозицией связи l_1^B . Структура связи l_1^B предполагает связь подблока B с внутренней границей блока Π (т.е. $\underline{\Pi}$), затем связь между внутренней и внешней границами блока Π и затем связь блока Π самого с собой (внутри "старой" связи C_0):

$$l_1^B = \Pi(B).(Nil_1^D(T)) \Leftrightarrow \Pi(\underline{\Pi}).(Nil_1^D(T)) \Leftrightarrow \Pi.(Nil_1^D(T)) \Leftrightarrow \Pi.(Nil_1^D(T)) \Leftrightarrow \Pi(\underline{\Pi}).(Nil_1^D(T)) \Leftrightarrow \Pi(B).(Nil_1^D(T)) \quad (1)$$

Упустив промежуточные этапы, выражение (1) можно переписать как:

$$l_1^B = \Pi(B).(Nil_1^D(T)) \Leftrightarrow \Pi(B).(Nil_1^D(T)). \quad (2)$$

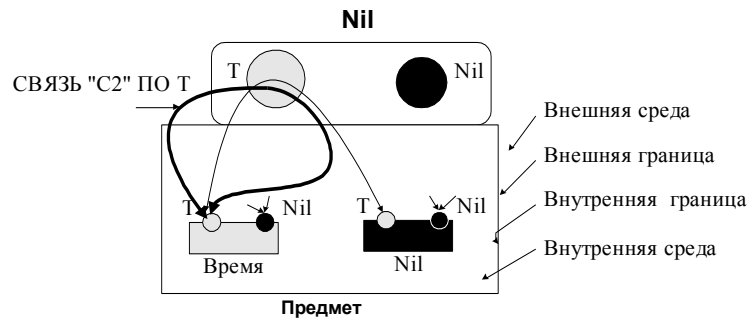


Рисунок 4 - Уровень ввода времени как свойства и блока

3.3. Уровень значений свойства времени и моделей пространства

Уровень 3 предполагает декомпозицию свойства T на совокупность дискретных значений времени и одновременную декомпозицию блока B на совокупность моделей пространств, соответствующих отдельным значениям времени (рис. 5).

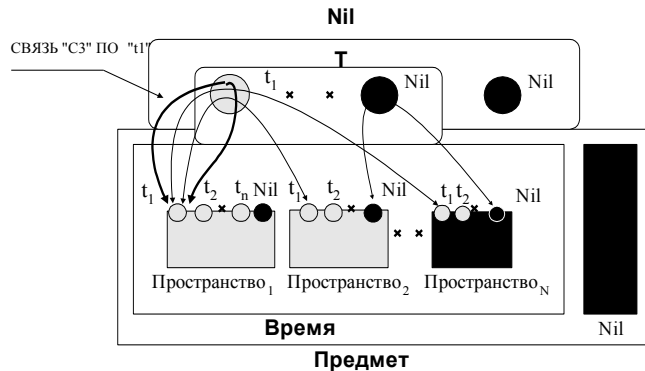


Рисунок 5 - Уровень определения моделей пространств

3.4. Уровень пространственных точек и их идентификаторов

Уровень 4 предполагает перевод всех значений свойства T в разряд новых свойств и дальнейшую их декомпозицию на ряд собственных значений, в качестве которых выступают идентификаторы физических точек пространства. Одновременно выполняется декомпозиция моделей пространств P на блоки - физические точки пространств (рис. 6).

Структура пространственной связи:

$$l_{kv}^{Tx} = P_{i1}^k(X_{r1}^k).t_c^k(x_u^v) \Leftrightarrow P_{i2}^k(X_{r2}^k).t_c^k(x_u^v); \forall r1, r2.$$

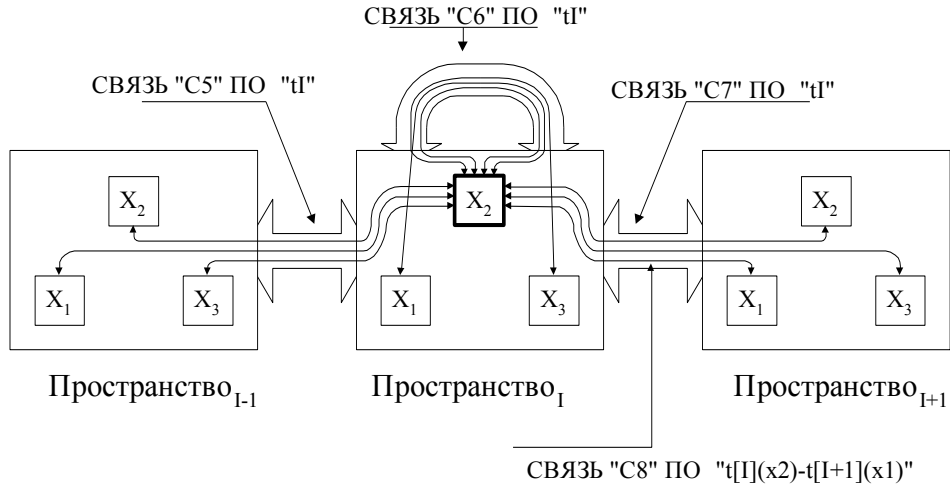


Рисунок 6 - Фрагмент совокупности связей ПТ

3.5. Уровень «простых» свойств и внутренних функций блоков

Уровень 5 предполагает перевод всех идентификаторов пространственных координат из разряда значений в разряд новых свойств и дальнейшую их декомпозицию на ряд собственных значений - идентификаторов "простых" свойств физических точек пространства, а так же декомпозицию моделей блоков - физических точек пространств на множество блоков - носителей простых свойств или внутренних функций (ВФ). Каждое "простое" свойство есть потенциал ПТ в данной предметной области (гидро, тепло и т.д.). ВФ наследует все свойства и связи ПТ. Совокупность свойств внешней границы отдельной ПТ определяется множеством ее связей (табл. 1).

Таблица 1. Пример состава свойств ПТ X_2^0 .

t_{-1}^1						t_0^2						t_{+1}^3					
x_1^1		x_2^2		x_3^3		x_1^4		x_2^5		x_3^6		x_1^7		x_2^8		x_3^9	
s_1^1	s_2^2	s_1^3	s_2^4	s_1^5	s_2^6	s_1^7	s_2^8	s_1^9	s_2^{10}	s_1^{11}	s_2^{12}	s_1^{13}	s_2^{14}	s_1^{15}	s_2^{16}	s_1^{17}	s_2^{18}

В этом примере верхние индексы нумеруют свойства в пределах всего множества свойств данного уровня, а нижние индексы - в пределах вышележащей структуры данных. Т.о., внешняя граница ВФ:

$$\forall i, j, c, u : G[P_i (X_j^i (F_z^{ij}))] = \&\{ P_i (X_j^i (F_z^{ij})) t_c^j (x_u^c (s_h^u)) \}_{h=1}^{N_{ijcu}^{ij}} ;$$

$$\{ t_c^{ij} (x_u^c (s_h^u)) \}_{h=1}^{N_{ijcu}^{ij}} \subseteq t_c (x_u^c).$$

G4) Множество «простых» связей внутри пространственных связей: $l_{kv}^{Tx} = \{ l_{kve}^{Txs} \}_{e=1}^{N_e^{kv}}$. Каждая отдельная "простая" связь имеет вид:

$$l_{kve}^{Txs} = P_{i1}^k (X_{r1}^k) t_c^k (x_u^v (s_b^e)) \Leftrightarrow P_{i2}^k (X_{r2}^k) t_c^k (x_u^v (s_b^e)); \forall r1, r2.$$

3.6. Уровень значений "простых" свойств и "кортежей" функций

Выполняется перевод всех идентификаторов "простых" свойств ПТ из разряда значений в разряд новых свойств и дальнейшую их декомпозицию на ряд собственных значений, т.е. идентификаторов значений "простых" свойств, а так же декомпозиция ВФ на блоки - носители значений "простых" свойств - "кортежи".

3.7. Выводы по разделу

Предложенная система моделей фактически задает систему вложенных языков формальных спецификаций описанного выше вида.

4. Комплекс методов задания семантических зависимостей над контекстно-свободными грамматиками, адаптированные к различным условиям их применения

В данный комплекс входят:

- атрибутивные грамматики, где каждый терм или синтерм грамматики имеет список атрибутов, задающих его принадлежность к именованному прототипу решения; используются для задания множеств прототипов малой мощности;

- отношения совместности-несовместности над ИЛИ-узлами контекстно-свободной грамматики, моделирующие в обобщенном виде атрибутивные грамматики, и косвенно моделирующих атрибутивные грамматики; используются для задания множеств прототипов средней мощности;

- контекстно-свободные грамматики с явным использованием НЕ-фактора, идентифицирующего неопределенный компонент грамматики, играющие совместно с предикатом «Существует» и возможным отрицанием роль посылок и выводов в явно заданных продукционных зависимостях, так же косвенно моделирующих атрибутивные грамматики; используются для задания множеств прототипов большой мощности.

Конкретные примеры некоторых грамматик будут показаны в рамках последующих разделов.

5. Комплекс алгоритмов выполнения теоретико-множественных операций над грамматиками, адаптированные к различным условиям их применения

Автором предложен целый комплекс алгоритмов выполнения теоретико-множественных операций над грамматиками, адаптированные к различным условиям их применения.

Условия применения включают:

3.1) Форму грамматик:

- атрибутивные грамматики;
- грамматики с отношениями совместности-несовместности;
- грамматики с НЕ-фактором;

3.2) Семантику грамматик:

- грамматики с неизвестной семантикой;
- грамматики с известной семантикой;

Ниже, в качестве примера, приведен алгоритм выполнения ТМО над грамматиками с неизвестной семантикой и синтаксисом.

5.1. Специфика алгоритм выполнения ТМО над грамматиками с неизвестной семантикой и синтаксисом

Специфика предлагаемого подхода к представлению знаний в специализированной инструментальной оболочке для создания интеллектуальных САПР – мета-эвристической оболочке (МЭО) описана ранее в [10-22] и кратко может быть охарактеризована следующим образом. База знаний представляет собой И-ИЛИ-дерево с определенными отношениями (продукциями) над ИЛИ-синтермами. Цель вывода в базе знаний - обеспечение выбора требуемого прототипа по техническому заданию (ТЗ) как подмножеству значений ИЛИ-синтермов, т.е.:

1) Отношения между ИЛИ связывают те термы, комбинация которых принадлежит некоторому непустому множеству семантически верных (проверенных) прототипов, имеющих место в И-ИЛИ-дереве;

2) Аксиомы, или прототипы есть основа построения И-ИЛИ-деревя;

3) И-ИЛИ-дерево есть средство для компактной записи множества известных прототипов и порождения гипотез о возможных новых прототипах;

4) И-ИЛИ-дерево – это множество синтаксически правильных выражений;

5) Продукции определены над И-ИЛИ-деревом и задают правила вывода, которые в совокупности позволяют вычлнить из И-ИЛИ-деревя семантически верное подмножество, т.е. те же самые аксиомы-прототипы.

При работе в открытой базы знаний в среде мета-эвристической оболочки возникают следующие задачи:

- обобщение множеств прототипов по структуре и по функциям в пределах отдельного модуля знаний, т.е. создание И-ИЛИ-деревя;

- аппарат вывода в смысле теории сложности САУ, по базе знаний – как целевого пространства систем (ЦПС) [35] на базе ТЗ, аппарат преобразования И-ИЛИ-деревьев.

Как в том, так и ином случае основу аппарата составляют теоретико-множественные операции (ТМО) над И-ИЛИ-деревом, и, в частности, такие операции, как пересечение, объединение, разность и дополнение множеств прототипов, хранящихся в И-ИЛИ-деревьях.

Все прототипы как модели объектов имеют в МЭО следующую иерархию описаний:

1) внешнее описание прототипа - ТЗ; назначение - внешняя идентификация, выбор прототипа из множества всех прототипов; интерфейс с малоквалифицированным пользователем;

2) описание на языке внутреннего представления, имеющего определенную грамматику: блоки, массивы блоков, свойства блоков, значения свойств, массивы свойств, связи, массивы связей; назначение - явное текстовое писание модели; интерфейс с высококвалифицированным пользователем;

3) табличную форму записи и хранения модели; назначение - внутренне, инструментальное представление прототипов в форме, позволяющей обеспечить оптимальную форму для хранения и преобразования; интерфейс с пользователем в этом случае - не предусматривается.

Табличная форма записи прототипов есть преобразованная форма представления на языке внутреннего представления.

И-ИЛИ-дерево как форма обобщения прототипов имеет место на любом уровне представления моделей. Форма Бэкуса-Наура (БНФ) может выступать как форма представления И-ИЛИ-дерева, свойственная грамматическому характеру описаний моделей на уровнях 1 и 2. С другой стороны, БНФ всегда можно соотнести с И-ИЛИ-деревом в любой форме представления, в том числе и табличной, свойственной уровню 3. Т.о. средства выполнения ТМО над БНФ может рассматриваться как универсальный аппарат работы с И-ИЛИ-деревьями в любой форме представления - от уровня 1 до уровня 3.

Т.о. необходимо создать аппарат ТМО над БНФ. Для чего необходимо определить:

- класс грамматик, представленных в форме БНФ;
- возможные ограничения на алгоритм, т.е. область его применения и возможности;
- выбрать оптимальный набор инструментальных средств его представления, определяемый спецификой задачи.

И-ИЛИ-деревья, представленные в виде БНФ, которые могут в зависимости от способа решения задачи обобщения и изобретения иметь две альтернативных формы представления:

1) Всякая альтернатива в ИЛИ-синтермах связана напрямую со списком идентификации ряда прототипов (исходная версия). Множество прототипов есть подмножество множества синтаксически верных выражений СМ. Грамматическая трактовка: вариант контекстной зависимости в грамматиках, заданных идентификацией прототипов.

2) Всякая альтернатива в ИЛИ-синтермах не связана напрямую со списком идентификации ряда прототипов. Множество прототипов совпадает с множеством синтаксически верных выражений СМ. Грамматическая трактовка: вариант контекстной независимости в грамматиках.

В работе [13] сделан обзор существующих и возможных методов решения данной задачи, определен класс грамматик, представленных в форме БНФ, введены возможные ограничения на алгоритм, т.е. область его применения и возможности и доказана возможность построения необходимого алгоритма.

Опишем метод решения задач выбора оптимального набора инструментальных средств его представления, определяемый спецификой задачи и выполняется собственно построение алгоритма для второй формы представления И-ИЛИ-деревьев, предполагающей контекстную независимость в грамматиках.

5.2. Выбор средств реализации алгоритма

Для задания алгоритма возможно применение различных прогрессивных форм представления из среды CASE-технологий или объектно-ориентированного программирования.

Наиболее оптимальным путем есть применение идей и средств R-технологии автоматизации проектирования программ (Вельбицкий И.В.) по причине:

- 1) явного включения средств, ориентированных на работу с текстом;
- 2) удобная графическая форма представления алгоритма.

Пусть дано два множества текстов описания моделей, заданных БНФ. Необходимо сформировать алгоритм выполнения над ними теоретико-множественных операций (пересечение, объединение, разность, дополнение).

Предлагается следующий алгоритм сравнения двух множеств, порождающий на выходе результаты указанных теоретико-множественных операций (пересечение, объединение, разность, дополнение).

5.3. Основные определения, соглашения и ограничения

5.3.1. Основные определения

Определение 1.

Термом называется элементарный символ множества. Термы соединяются между собой только посредством операции "И" (&).

Определение 2.

Синтермом называется имя множества, которое может раскладываться. Элементами разложения могут быть как термы, так и синтермы, соединенные посредством операции "И" (&) или "ИЛИ" (V). Синтермы всегда записываются только в угловых скобках "<>".

Определение 3.

Если два множества совпадают по имени, то это означает, что они эквивалентны и по структуре, т.е.

одно и то же имя означает одно и то же множество.

Определение 4.

Если два множества совпадают по структуре, то это значит, что имена у них разные, а подмножества и способ их объединения одинаковый, т.е. одна и та же структура может иметь много разных форм записей, но при полном разложении этих форм записи мы в результате получим одно и то же.

5.3.2. Основные соглашения

Соглашение 1.

Элементы каждого отдельного множества должны соединяться только по "И", или только по "ИЛИ". Совместное использование этих двух знаков операций при записи разложения отдельного множества не допустимо.

Соглашение 2.

Не существует двух разных путей, порождающих одну и ту же цепочку термов.

Соглашение 3.

Каждый элемент по "ИЛИ" можно рассматривать как символ и как множество цепочек символов, начинающихся с этого символа. Если "ИЛИ" выступает как символ, признаки помещаются в графе символа "СИМВ", в противном случае - в графе множества "МН".

Соглашение 4.

Если при движении по некоторому пути выяснилось, что данный символ "ИЛИ" представляет собой полностью просмотренное множество символов, то при движении по любому другому пути, приводящего к этому же символу, множество цепочек, начинающихся с этого символа, будет иметь тот же признак.

Соглашение 5.

Каждая строка вновь вводимого множества должна заканчиваться признаком конца "%". Знак "%" в конце строки играет служебную роль и в конечный текст не входит.

Соглашение 6.

Признаком "ЗНАК" (Z) помечаются все знаки множества как в основном и во вспомогательном стеках.

Соглашение 7.

Признаком "НЕОТРАБОТАН" (NO) помечаются все не отработанные элементы множества, соединенные по "ИЛИ" во вспомогательном стеке.

Соглашение 8.

Признаком "РАСКЛАДЫВАЕТСЯ" (R) помечаются все синтермы во вспомогательном стеке, которые подверглись разложению.

Соглашение 9.

Признаком "СОВПАДЕНИЕ" (S) помечаются все элементы множества (термы или синтермы) во вспомогательном стеке, которые совпали со сравниваемыми элементами другого множества.

Соглашение 10.

Признаком "НЕСОВПАДЕНИЕ" (NS) помечаются все элементы множества (термы или синтермы) во вспомогательном стеке, которые не совпали со сравниваемыми элементами другого множества.

Соглашение 11.

При описании алгоритмов использованы сокращения:

ZAKON - регистр, куда вписывается правая часть множества $S = \text{"операция"}(a_1, a_2, a_3)$,

ТОРМ - таблица определения идентификаторов множеств (синтермов), каждая строка по структуре имеет вид $S = \&(a_1, a_2, a_3)$,

ТННА - таблица счетчиков номеров амперсендов;

СТЕК - основной стек, имеется два основных стека СТЕК1 и СТЕК2, предназначенных для левого и правого множества,

VСТЕК - вспомогательный стек, имеется два вспомогательных стека VСТЕК1 и VСТЕК2, предназначенных для левого и правого множества, элементы - имена IM1 или IM2, имеющие:

- признаки: "NO" - неотработан;

- графу "МН" (множество), где возможны признаки - "NS", если оно не совпало и "S" - если совпало;

- графу "СИМВ" признак "R" - развернут, признак "S" - свернут;

PRST - признак стека, для левого стека принимает значение "L", для правого - "P";

PRS - признак сворачивания или несворачивания стека, принимает значение "1" и "0" соответственно;

RSR - регистр, хранящий синтерм из VСТЕК с признаком "NS" в графе "МН", если он не совпал и "S" - если совпал;

IM - регистр для хранения идентификатора множества, выбираемого из стека для сравнения, имеется IM1 и IM2 исходя из типа стека;

TCH - текущее значение счетчика "амперсендов" - промежуточных служебных множеств - синтермов, порождаемых алгоритмом; имеет начальное значение 1;

"@четный номер" и "@нечетный номер" - имена "амперсендов"; тут "четный номер" и "нечетный номер" - значения TCH; синтермы @четный и @нечетный позволяют объединять и пересекать множества,

имеющие разные синтермы, но одни и те же базовые термы, т.е. собственно порождаемое описание.

Соглашение 12.

В одном выражении можно использовать совместно термы и синтермы.

5.4. Основные ограничения

Ограничение 1.

Запрещено определять синтермы через самих себя.

Ограничение 2.

Порядок просмотра термов и синтермов при последовательном разложении БНФ одинаков как в первом так и во втором множестве.

5.5. Алгоритм ТМО над БНФ

5.5.1. Алгоритм записи в стек

Вход: Правая часть множества, записанная в регистр ZAKON. Выход: STEK.

Метод: При записи правой части множества в стек каждый терм записывается в отдельную ячейку стека и совокупность термов накрывается знаком "&". Каждый синтерм записывается так же в отдельную ячейку стека. Знак операции, посредством которого соединены подмножества - в вершине стека.

5.5.2. Алгоритм разложения

Вход: Элемент множества. Выход: TOPM.

Метод: Поиск элемента множества в TOPM. Если нашли, то запись правой части в STEK в соответствии с алгоритмом записи в стек, затем дозапись STEK в STEK1 или STEK2, а синтерма, который разложился - в VSTEK1 или в VSTEK2 соответственно, с признаком "R".

Если не нашли, то выдача сообщения, что данное имя - терм.

5.5.3. Алгоритм сворачивания (восстановления)

Вход: STEK, TCH, TOPM, PRST, VSTEK.

Выход: STEK, THNA.

Метод:

1) Если PRST = L, то наращиваем TCH на 1 (получаем @нечетный номер), затем наращиваем TCH еще на 1 (получаем @четный номер). Таким образом, имеем два регистра: @четный номер и @нечетный номер. Смотрим в STEK:

- идут подряд два знака, то первый знак поднимаем, а начиная со второго, анализируем;

- стоит один знак, начинаем анализ этого знака:

а) знак "&" - анализируем признаки группы идентификаторов до следующего знака:

- если все идентификаторы имеют признак "S", то эта группа уничтожается, синтерм в RSR приобретает признак "S" в графе "MH", и помещается в STEK под первый знак, если он есть или в STEK, если его там нет;

- если встретился хоть один идентификатор с признаком "NS" в графе "MH", то группа идентификаторов уничтожается, синтерм в RSR приобретает признак "NS" в графе "MH" и помещается в STEK под первый знак или в STEK.

б) знак "V":

- если встречаем идентификаторы с признаком "S", то засылаем их в @четный номер. Они должны соединяться посредством знака операции "V", и когда группа идентификаторов просмотрена, то цепочка этих идентификаторов должна заканчиваться признаком конца "%";

- если встречаем идентификаторы с признаком "NS", то засылаем их в @нечетный номер аналогично.

После того, как группа "ИЛИ" в STEK1 иссякла, анализируем регистры с амперсендами:

- если @четный номер пуст, а @нечетный номер полон, то уничтожает эти "ИЛИ", присваиваем синтерму из RSR признак "NS" в графе "MH" и засылаем в STEK1 под знак, или в вершину стека;

- если @четный номер полон, а @нечетный номер пуст, то

уничтожаем эти "ИЛИ", присваиваем синтерму из RSR признак "S" в графе "MH" и засылаем в STEK1 под первый знак, или в вершину стека, если знак отсутствует;

- если оба амперсенды не пусты, то записываем в TOPM сформировавшиеся четные и нечетные амперсенды, а затем

<имя синтерма> = @нечетный номер V @четный номер %,

где <имя синтерма> - синтерм, который разложился на совпадающие и несовпадающие "ИЛИ", попавшие в соответствующий амперсенд.

Затем эти амперсенды записываются в STEK1 со следующими признаками в графе "MH":

- @четный - с признаком "S";

- @нечетный - с признаком "NS".

Значение счетчиков записываются в таблицу хранения номеров амперсенда: сначала нечетный, а затем четный.

2) Если PRST = P, то

выбираем первый элемент из вершины VSTEK.

Если это знак, то засылаем его в основной стек STEK и выбираем следующий элемент из VSTEK.

Если это не синтерм, или синтерм, который не раскладывался (т.е. не имеет признака "R"), то:

- если $PRS = 1$, то поднимает знак, находящийся в вершине STEK и переписываем в него все идентификаторы из VSTEK до первого знака или до синтерма с признаком "R";

- если $PRS = 0$, то все идентификаторы до первого знака или до синтерма с признаком "R" переписываем в STEK.

Все идентификаторы в STEK переписываем с признаками, присущими им в VSTEK. Если производится сворачивание, признаки уничтожаются.

Если это синтерм с признаком "R", то:

- если $PRS = 0$, дописываем этот синтерм в STEK и заканчиваем работу;

- если $PRS = 1$, смотрим в VSTEK:

а) идут подряд два знака: первый знак поднимаем, а начиная со второго уничтожаем все идентификаторы до следующего знака, а вместо них под поднятый знак помещаем элемент из VSTEK, который раскладывался;

б) стоит один знак: тогда уничтожаем все идентификаторы, начиная с этого знака до следующего. Следующий знак поднимаем и под него записываем синтерм; конец работы.

5.5.4. Алгоритм прямого хода.

Вход: STEK1, STEK2, VSTEK1, VSTEK2, ТОРМ.

Выход: STEK1, STEK2, VSTEK1, VSTEK2.

Метод: Сравниваем имена множеств.

Если $IM1 = IM2$, то множества совпали полностью.

K0:

Выбираем сообщения, что множество 1 совпало с множеством 2 и оканчиваем работу.

Если $IM1 \neq IM2$, то раскладываем $IM2$ и анализируем знак в вершине STEK2:

- знак & - идем на RIM1;

- знак V -

K1:

выбираем первый идентификатор из вершины STEK2, засылаем его в $IM2$ и сравниваем:

1) если $IM1 = IM2$, то на K0;

иначе раскладываем $IM2$ и анализируем знак в STEK2:

K2:

- если & - сворачиваем цепочку по "И" в соответствии с алгоритмом сворачивания и засылаем синтерм в VSTEK2, и на K1;

- знак V - на K1;

2) если $IM2$ - терм, то анализируем знак в STEK2:

- знак & - на K2;

- знак V - $IM2$ засылаем в VSTEK2 с признаком "NO" и выбираем следующий идентификатор по "ИЛИ".

Если STEK2 опустел, а $IM1 \neq IM2$, то восстанавливаем STEK2, сворачивая VSTEK2, выбираем $IM2$ из STEK2 и раскладываем.

RIM1:

раскладываем $IM1$ и анализируем знаки в STEK1 и STEK2:

если & - & - на YMN4,

иначе на INACH.

YMN:

выбираем идентификаторы из STEK1 и STEK2 и запоминаем их соответственно в регистрах $IM1$ и $IM2$ и сравниваем:

если $IM1 = IM2$, то засылаем их с признаками "S" в VSTEK1 и VSTEK2 соответственно и на YMN,

иначе

M3:

раскладываем $IM2$

- если $IM2$ - синтерм, то выбираем из STEK2 первый идентификатор, засылаем его в $IM2$ и на M0;

- если $IM2$ - терм, то смотрим, какой знак в вершине STEK2:

1) знак & - идем до первого "ИЛИ", или пока STEK2 не станет пуст, выбрасывая все встретившиеся идентификаторы в VSTEK2 с признаком "NS", заслав туда предварительно $IM2$ с признаком "NS".

Если нашли "ИЛИ", выбираем первый идентификатор по "ИЛИ" и на M0,

иначе

ZV:

восстанавливаем VSTEK2 до первого "S", или пока он не станет пуст (сворачивая). Затем выбираем

идентификатор из $??\text{TEK2}$, засылаем его в IM2 , раскладываем IM1 и IM2 и на YMN .

2) знак V - засылаем этот идентификатор в VSTEK2 с признаком "NS" и выбираем следующее "ИЛИ". При этом все встретившиеся "И" засылаем в VSTEK2 с признаком "NO". Если "ИЛИ" уже нет, то на ZV .

M0:

если $\text{IM1} = \text{IM2}$, то

M1:

выбрасываем в VSTEK2 все не просмотренные "ИЛИ" с признаком "NO" до первого знака, или пока STEK2 не станет пуст, затем содержимое IM2 с признаком "S" и все остальные "ИЛИ" до первого "И".

Из STEK1 также выбрасываем в VSTEK1 все не просмотренные "ИЛИ" с признаком "NO" до первого знака, а остальные со своими признаками, затем содержимое IM1 с признаком "S" и все остальные "ИЛИ" до первого знака и на YMN .

Если $\text{IM1} \neq \text{IM2}$, то на $M3$.

При выборе идентификаторов из стеков в поиске "И" могут возникнуть ситуации:

- 1) STEK1 пуст и STEK2 пуст - работает алгоритм обратного хода (идем на OBHOD);
- 2) STEK1 пуст, а STEK2 не пуст - для STEK1 работает алгоритм сворачивания по "И", а в STEK2 сворачиваем все до идентификатора с признаком "S" (также работает алгоритм сворачивания);
- 3) STEK1 не пуст, а STEK2 пуст - идем на M ;
- 4) STEK1 не пуст и STEK2 не пуст - продолжаем работу.

M : засылаем несовпавший по "И" идентификатор под знак в STEK1 и выбираем идентификатор из VSTEK1 :

- если это терм, то поднимаем знак в вершине STEK1 и помещаем терм туда;
- если это синтерм с признаком "R", то работает алгоритм сворачивания;
- если это знак:

знак $\&$ - засылаем его в STEK1 ;

знак V - первому идентификатору по "ИЛИ" присваиваем признак "NS" в графе "МН" и засылаем его в STEK1 , после чего начинает работать алгоритм обратного хода (идем на OBHOD).

При поиске этого первого "ИЛИ" (оно имеет признак "S" в графе "СИМВ"), работает алгоритм сворачивания.

INACH : выбираем идентификаторы из STEK1 и STEK2 , засылаем их в IM1 и IM2 соответственно и сравниваем:

- если $\text{IM1} = \text{IM2}$, то идем на $M1$;
- иначе идем на $M3$.

Примечание: после того, как выбрали очередной идентификатор из под знака, нужно проверять : если следующий знак, то выбранный знак не засылаем снова в STEK , а засылаем в VSTEK после последнего выбранного из STEK идентификатора.

5.5.5. Алгоритм обратного хода

Вход: STEK1 , STEK2 , VSTEK1 , VSTEK2 , TOPM , THNA .

Выход: TOPM .

Метод:

OBHOD: Идем по VSTEK1 в поисках первого идентификатора с признаком "NO" после первого встретившегося идентификатора с признаком "S".

Идентификаторы по "И" с признаком "S" в графе "СИМВ" переписываем в STEK1 с признаком "S" в "МН". При этом, отыскиваем такие же идентификаторы с признаком "S" в VSTEK2 , сохраняя признак, а все предшествующие ему идентификаторы в том порядке, в котором анализировались, восстанавливаем в STEK2 . Если встречаем синтерм с признаком "R", сворачивающий цепочку по "И", то сворачиваем в соответствии с алгоритмом сворачивания.

Если встретили знак "ИЛИ", переписываем все идентификаторы по "ИЛИ" из VSTEK1 до первого "S" в STEK1 . Причем, идентификаторы, имеющие признак "S" в графе "СИМВ", приобретают "S" в графе "МН", а имеющие "NS" в графе "СИМВ", приобретают "NS" в графе "МН". Такие же идентификаторы с признаками "S" или "NS" из VSTEK2 засылаются в STEK2 , попутно перебрасывая встретившиеся на пути к ним идентификаторы в STEK2 . Идентификаторы с признаком "S" в "СИМВ" приобретают такие же признаки в графе "МН".

Когда встретили первый идентификатор с признаком "NO" после того, как первый встретившийся "S" засылали в STEK1 , записываем его и все остальные до первого знака или синтерма с признаком "R" в STEK1 , а в правом стеке переписываем все идентификаторы с сохранением присущих им признаков в STEK2 до первого идентификатора с признаком "S", или пока VSTEK2 не станет пуст. Если не нашли больше "S" в VSTEK2 , то идем на POISK .

Затем выбираем первый идентификатор с "NO" из STEK1 и засылаем в IM1. Из STEK2 выбираем первый идентификатор с "NO" для сравнения, а все остальные, которые предшествуют ему, восстанавливаем в VSTEK2.

1) Если $IM1 = IM2$, то в левом стеке переписываем все идентификаторы по "ИЛИ" во вспомогательный стек из основного до первого знака, затем IM1 в VSTEK1 и IM2 в VSTEK2, а затем все остальные идентификаторы до первого "И", а в правом стеке = все идентификаторы до "ИЛИ" с "NO", затем все "NO" и идем на IM2.

POISK: выбираем из STEK1 идентификатор с признаком "NO" и начинаем сравнивать его со всеми идентификаторами STEK2, пока не дойдем до идентификатора, имеющего признак "S".

2) Если $IM1 \neq IM2$, то

S: восстанавливаем STEK2, пока VSTEK2 не станет пуст без сворачивания, раскладываем IM1, выбираем первый идентификатор из STEK1, засылаем его в IM1 и снова сравниваем со всеми идентификаторами STEK2 до идентификатора, имеющего признак "S".

Если IM1 не раскладывается, и не нашлось одинакового ему в STEK2, то смотрим, в каком контексте стоит этот терм:

- если в контексте "И", то

K: сворачиваем эту цепочку по "И", а ближайшему "ИЛИ", т.е. синтерму, свернувшему эту цепочку, даем признак "NS" в "МН" и помещаем рядом с просмотренными "ИЛИ" в STEK1 или в вершину стека и идем на POISK:

- если в контексте "ИЛИ", то

L: даем ему признак "NS" в "МН" и помещаем рядом с просмотренными "ИЛИ" в STEK1 или в вершину стека и на POISK.

Если $IM1 = IM2$, то смотрим, в каком контексте находятся IM1 и IM2:

- Если в STEK1 "&" и в STEK2 "&", то выбираем следующие идентификаторы из STEK1 и STEK2 и сравниваем, пока не исчерпаем всю цепочку "И". Если вся цепочка совпала, то сворачиваем в развернувший ее синтерм, даем ему признак "S" как множество, т.е. в "МН", и помещаем рядом просмотренными "ИЛИ" в STEK1 и на POISK. Если хоть одно "ИЛИ" этой цепочки не совпало, то на K.

- Если в STEK1 "V" и в STEK2 "V", то засылаем в VSTEK1 все "ИЛИ" с "MO", затем все просмотренные, а потом этот IM1 с "C", в правом стеке аналогично и далее - на POISK.

- Если в STEK1 "&", а в STEK2 "V", то на K. - если в STEK1 "V", а в STEK2 "&", то на L.

Процесс прекращаем, когда вся цепочка идентификаторов по "ИЛИ" просмотрена и на P.

Если при поиске первого "NO" в VSTEK1 встречаем синтерм с признаком "R" (при анализе "ИЛИ").

P: ищем в THNA текущее значение счетчика амперсенов и начинает работать алгоритм сворачивания.

После работы алгоритма сворачивания содержимое STEK переписывается в STEK1.

Процесс прекращается, когда в VSTEK1 уже нет идентификаторов с признаком "NO", т.е. VSTEK1 пуст. При этом все идентификаторы с присущими им, или приобретенными при анализе их признаками, попадают в STEK1.

После чего восстанавливаем VSTEK1 и начинаем анализ VSTEK1.

В очередной @четный записываем цепочку идентификаторов, имеющих признак "S" в графе "МН", объединяя их между собой знаком "&". Цепочка должна оканчиваться признаком конца "%" и помещаться в ТОРМ. Когда VSTEK1 опустел в результате поиска следующего "S" в графе "МН", и все идентификаторы оказались в STEK1, переписываем содержимое STEK1 в VSTEK1.

Идем по VSTEK1 в поисках идентификатора, имеющего в графе "МН" признак "NS", переписывая все встретившиеся идентификаторы в STEK1, а все с признаками "S" в графе "МН" в очередной @нечетный. После того, как нашли первый идентификатор с признаком "NS", записываем его в @нечетный, а в STEK1 не переписываем. Далее, в @нечетный дописываем все идентификаторы с признаком "S" в графе "МН", а в STEK1 все идентификаторы со своими признаками.

Когда VSTEK опустел, переписываем STEK1 в VSTEK1 и начинаем поиски следующего "NS", действуя аналогично.

Процесс прекращаем, когда в VSTEK1 нет идентификаторов с признаком "NS". После чего выдается на печать вся таблица ТОРМ. В ней: последний @четный номер - результат пересечения множеств, а последовательность последних @нечетных номеров результат разности множеств. Объединение множеств - это все первое множество плюс разность.

6. Методы обучения, т.е. – построения базы знаний, на базе ТМО, адаптированные к различным условиям их применения

Назовем методы обучения, т.е. – построения базы знаний, на базе ТМО, адаптированные к различным условиям их применения:

- построение атрибутивных грамматик путем автоматического выполнения теоретико-множественного объединения текстов отдельных решений - прототипов;

- прямого ввода контекстно-свободной грамматики и отношений совместности-несовместности для соответствующего типа грамматики;
- прямого ввода множеств операций пересечения, объединения, поиска разницы и дополнения для грамматик с НЕ-факторами – как способа построения пространства поиска решений.

Ниже приведен пример построения атрибутивных грамматик путем автоматического выполнения теоретико-множественного объединения текстов отдельных решений – прототипов.

6.1. Построение атрибутивных грамматик путем автоматического выполнения теоретико-множественного объединения текстов отдельных решений – прототипов

Пусть дано множество прототипов, входящих в тип блоков $A: A'=(P_1 \vee P_2 \vee P_3)$. На рис. 7 приведена обобщенная схема типа A , построенная в рамках предлагаемой концептуальной модели.

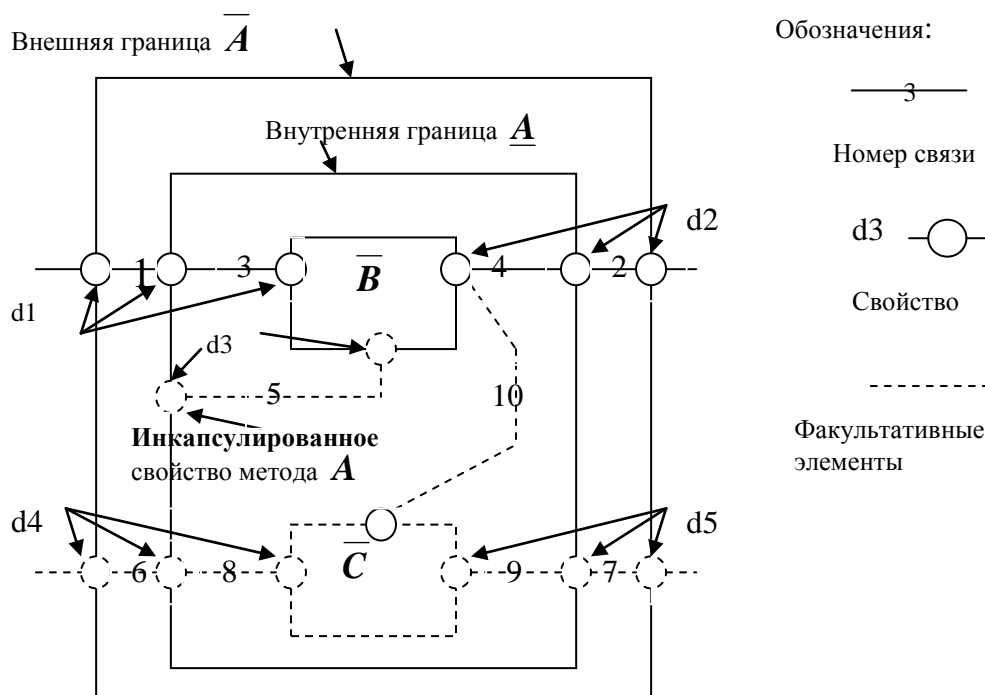


Рисунок 7 - Обобщенная схема типа A

Таблица 2 - Описание множества связей в типе блока.

Номер связи	Описание связи	Системобр./ Факульт-я.	Принадлежность прототипам
1	$\overline{A} : d1 \leftrightarrow \underline{A} : d1$	С	1,2,3
2	$\overline{A} : d2 \leftrightarrow \underline{A} : d2$	С	1,2,3
3	$\underline{A} : d1 \leftrightarrow \overline{B} : d1$	С	1,2,3
4	$\underline{A} : d2 \leftrightarrow \overline{B} : d2$	С	1,2,3
5	$\underline{A} : d3 \leftrightarrow \overline{B} : d3$	Ф	1,2
6	$\overline{A} : d4 \leftrightarrow \underline{A} : d4$	Ф	2,3
7	$\overline{A} : d5 \leftrightarrow \underline{A} : d5$	Ф	2,3
8	$\underline{A} : d4 \leftrightarrow \overline{C} : d4$	Ф	2,3
9	$\underline{A} : d5 \leftrightarrow \overline{C} : d5$	Ф	2,3
10	$\overline{B} : d3 \leftrightarrow \overline{C} : d3$	Ф	2,3

Результат выполнения теоретико-множественных операций над совокупностями связей, образующими данные прототипы, показан на рис. 8.

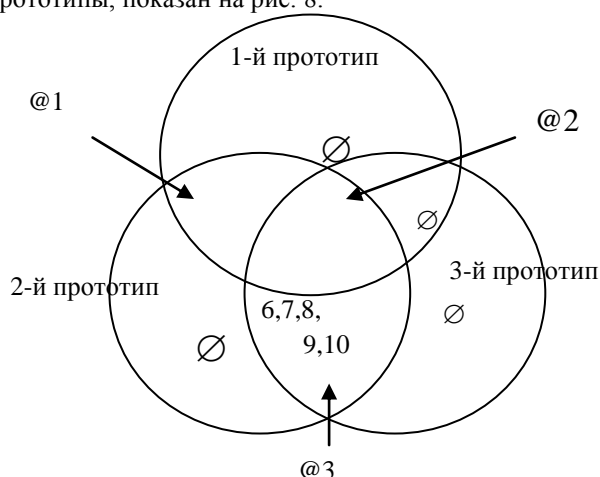


Рисунок 8 - Выделение частей прототипов

Тут: @j - некоторая часть внутренней среды прототипов; 1,2... - номера связей, ∅ - пустое множество связей. При этом:

$$P_1 = @1 \& @2; P_2 = @1 \& @2 \& @3; P_3 = @2 \& @3; \\ @1 = 5; @2 = 1 \& 2 \& 3 \& 4; @3 = 6 \& 7 \& 8 \& 9 \& 10.$$

Преобразуем множество прототипов А к форме И-ИЛИ-дерева:

$$A = (P_1 \vee P_2 \vee P_3) = @2 \& H1; H1 = @1 \vee @3 \vee H2; H2 = @1 \& @3.$$

На рис. 9 изображено полученное И/ИЛИ дерево. В скобках показаны номера прототипов, входящих в данную вершину, числами заданы номера связей, стрелками показан порядок декомпозиции узлов.

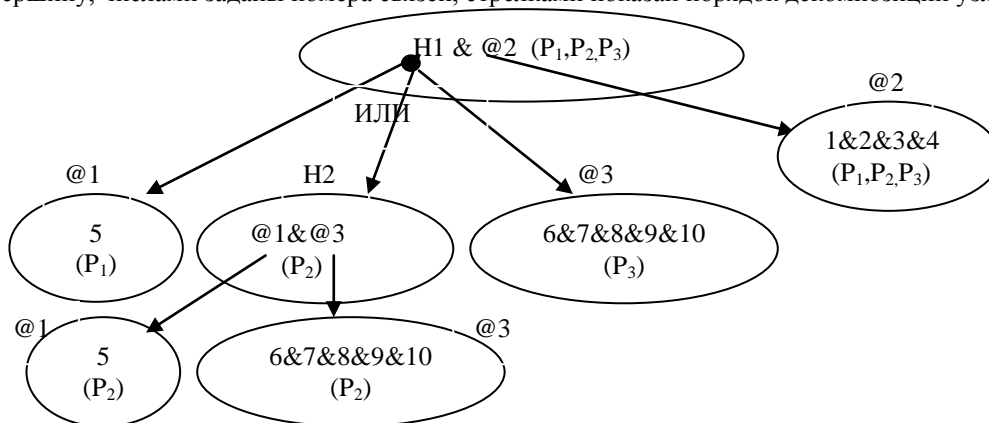


Рисунок 9 - Форма И-ИЛИ-дерева

7. Методы организации логического продукционного вывода на базе ТМО, адаптированные к различным условиям их применения

Структура комплекса методов использования ТМО такова:

- усечения атрибутивных грамматик путем использования комплекса стандартных правил вывода;
- усечения грамматик с отношениями совместности – несовместности путем использования комплекса стандартных правил вывода;
- пересечения, объединения, поиска разницы и дополнения для грамматик с НЕ-факторами, выполняемых в рамках заданного множества продукций.

Фактически, речь идет реализации алгоритма функционирования процедуры П4 управления выводом в СМ. Ниже, в качестве примера, приведен вариант процедуры П4 для реализации пространственно-временной логики в рамках общего метода усечения атрибутивных грамматик путем использования комплекса стандартных правил вывода.

7.1. Вариант П4, обобщенный: часть прототипов известна, а часть - нет.

Пусть дано:

1) ТЗ - часть значений X'_{ij} для ряда T_j ; $X'_{ij} = \{X'_{ij}{}^m\}_{m=1}^{M'ij}$, но собственно искомые прототипы - неизвестны;

2) Куча - совокупность $X_{ij} = \{X_{ij}{}^m\}_{m=1}^{Mij}$, при том, что верно $X_{ij}{}^m \rightarrow \{\Pi_i^1, \dots, \Pi_i^{Ki}, Nil\}$, т.е. имеется частичное соответствие - "границы ПТ <-> состояния пространств".

Найти: все прочие X_{ij} .

Идем по пути 1, т.е. без счета прототипов, но удаляем неподходящие прототипы из числа известных, т.е. выполняем счет прототипов. Если мы сводим модель к одному из известных Π_i^k , то задача решена, т.к. траектория уже известна. Все новые состояния запоминаются как Π_i^k и как часть нового P^k .

Процедура П4:

1) Делаем текущим T_j начальное время T_1 ;

2) Выбираем один из X'_{ij} для данного T_j ;

3) Выполняем пересечение X'_{ij} и X_{ij} по множеству $\{X_{ij}{}^m\}$. Если пересечение пусто - выход по неудаче, если иначе - идем на 4;

1) Сужаем X_{ij} на дополнение к X'_{ij} , т.е. удаляем из X_{ij} неиспользуемые $\{X_{ij}{}^m\}$, а так же формируем список удаляемых состояний пространств Π_i^k (траекторий-прототипов P^k);

4) Удаляем для данного T_j из всех прочих $\{X_{ij}{}^m\}_{m=1}^{Mij}$ удаленные ранее состояния (прототипы), пустые $X_{ij}{}^m$ - не удаляются; затем последовательно проделываем то же самое для всех $X_{ij}{}^m$ из T_{j+1} , T_{j+2} и т.д. до T_N ;

5) Переходим на 7 по просмотру всех $\{X_{ij}\}$ в T_j , иначе идем на 2;

6) Выбираем множество продукций - связей $L_{ij} = \{l_{ij}^k\}_{k=1}^{Kij}$ для данного X_{ij} со всеми $\{X_{ij}{}^m\}_{m=1}^{Mij}$ в данном T_j , а так же множества связей со всеми $\{X_{ij}{}^m\}_{m=1}^{Mij}$ в пространствах T_{j+1} , T_{j+2} и т.д. до T_N ;

7) Выполняем потоковый алгоритм Нариньяни [10] на множестве полученных связей, при том, что множество отношений интерпретации задано множеством границ; т.е.:

- выбираем одну из связей l_{ij}^k и обрабатываем ее, т.е. сужаем множество границ смежной по связи

ПТ $X_{i',j} = \{M_{i',j}^k\}_{k'=1}^{K'ij}$ в данном T_j ;

- далее переопределяем по связям $X_{i',j}$ множество возможных значений свойств у связанных с ней

ПТ и т.д. во всех пространствах вплоть до окончания списка инициированных связей.

Потоковый алгоритм Нариньяни предполагает конечное число шагов [31];

8) Если область определения любого свойства любой ПТ оказалась пуста, то выход по неудаче, иначе - идти на 10;

9) Если обнаружен конец списка $\{X'_{ij}\}$ - то идти на 11, иначе - возврат на 2;

10) При $T_j = T_N$ - переход на 12; иначе - переходим к $X_{ij}{}^m$, часть X_{ij} которого уже определена на предшествующих шагах и далее переходим на 2;

11) Если хотя бы один X_{ij}^m из оставшихся не удаленными оказался пуст - то вводим новую идентификацию для данного нового прототипа и новых состояний в каждом T_j и забыванием их; далее конец алгоритма по успеху.

8. Метод оценки сложности выполнения теоретико-множественных алгоритмов над грамматиками различных типов

Опишем метод оценки базы качества базы знаний, построенной на основе предлагаемого метода представления знаний

Все известные меры, которые возможно применить для оценки качества систем с базами знаний могут быть поделены на две группы:

1) Меры оценки механизмов логического вывода и структуры БЗ включают фактуальные оценки и технологические меры; например, фактуальные оценки включают: сложность БЗ, информативность БЗ, надежность вывода решения, достоверность выведенного решения, устойчивость БЗ, быстрдействие системы представления и обработки знаний и т.д.

2) Меры оценки критериальных свойств для разработчиков - когнитологов и экспертов: релевантность знаний (их подтвержденность и достаточность), полнота умений (необходимость и достаточность процедур), проверенность знаний (их тестированность и целостность), уровень интеллектуальности (обучаемость, гибкость стратегий рассуждения и интерфейса), наличие мета-знаний и т.д.

Следует отметить громоздкость рассмотренной системы оценки качества БД. Мы будем рассматривать только главные выходные показатели эффективности вывода в базе знаний, задающие эффективность D-алгоритма синтеза логической области дедуктивной выводимости (ЛОДВ), соответствующего по структуре нашему алгоритму, а именно:

- L(S) – длину логического описания ОДВ (число конъюнктов в ДНФ ЛОДВ);
- M(S) – число замещений литералов в процессе работы алгоритма.

При этом, L(S) задает пространственную сложностью, M(S) – временную сложностью алгоритма. Временная и пространственная сложность вывода есть функции, зависящие от аргументов, являющимися структурными показателями сложности базы знаний:

- H(S) – высота И-ИЛИ-дерева, т.е. наибольший ярус дерева (ярус целевой вершины полагается нулевым).

- N(S) – число продукций в ЛСП S;
- q(S) – число нетерминальных литералов в ЛСП S, увеличенное на 1;
- x(s) – число терминальных литералов ЛСП S;
- k_v - наибольшая степень ИЛИ-вершин И-ИЛИ-дерева, соответствующего ЛСП S;
- $k_{\&}$ - наибольшая степень И-вершин И-ИЛИ-дерева, соответствующего ЛСП S.

Длина ЛОДВ, построенного D-алгоритмом (ленточная сложность), не превысит

$$L_D(S) \leq k_{\&}^{k_{\&}} * k_v^{(q(S)-H(S)+1)*(H(S)-2)}. \quad (3)$$

Общее число построенных в процессе работы D-алгоритма конъюнктов не превысит (временная сложность):

$$M_D(S) \leq k_v + k_{\&}^{k_{\&}} * k_v^{\frac{k_v^{(q(S)-H(S)+1)*(H(S)-1)} - 1}{k_v - 1}}. \quad (4)$$

9. Методы преобразования грамматического описания объекта, имеющего не допустимую когнитивную сложность представления, к виду, имеющую допустимую форму представления когнитивной сложности

9.1. Краткая характеристика метода построения меры когнитивной сложности (КС)

Работа является последней в серии статей, посвященных когнитивной сложности моделей. Ранее были разработаны: 1) методика тестирования пользователей САПР для определения шкалы абсолютной КС моделей и границ допустимой КС представления моделей [24-26; 2) метод построения формальной меры для оценки когнитивной сложности моделей. Метод построения формальной меры для оценки когнитивной сложности моделей кратко может быть охарактеризован так. Упорядочивания по сложности совокупности отношений в пределах отдельно взятого формального описания прототипа выполняется с помощью модифицированного метода диаграмм Хассе [35].

Кратко изложим суть метода. Пусть дана некоторая замкнутая ограниченная модель внутренней среды прототипа P , представленная множеством экземпляров отношений различных уровней общности. Данные отношения представлены таблично и могут быть упорядочены по взаимному включению в соответствии со

шкалой $\mathbf{Ш}_l$. Вес несравнимых слов, задан системой неизвестных коэффициентов:

$$\mathbf{M} = \{M_i^m\}_{i=1, \overline{I m}}^{m=1, \overline{W}}. \quad (5)$$

Здесь: W - число уровней типов отношений, имеющих место в прототипе; m - номер уровня иерархии отношений, i - номер отношения в пределах уровня; $I m$ - число отношений на уровне. Общее число отношений в диаграмме будем обозначать как I .

Тогда структурная (системная) сложность S_l для объекта-прототипа $P_l, l = \overline{1, Np}$ может быть вычислена по формуле:

$$S_l = S_1^W \text{ для } \forall l : l = \overline{1, Np}. \quad (6)$$

Здесь: $m = W$ - номер верхнего уровня иерархии отношений, $i = I$ - номер единственного отношения верхнего уровня; S_1^W - сложность прототипа, представленного как замкнутая среда.

Сложность прототипа определяется по рекуррентному соотношению:

$$S_i^m = M_i^m * \sum_{j=1}^{K_i^{m-1}} S_j^{m-1}, i = \overline{1, I_i^m}. \quad (7)$$

Тут: i - номер узла m -го уровня; j - номер составляющего отношения в узле; K_i^{m-1} - число отношений в узле. Всякий узел содержит структуру Nil . При этом для базового 1-го уровня (значений свойств) выполняется $S_j^1 = I; \forall j$.

Пример состава отношений приведен на рис. 10.

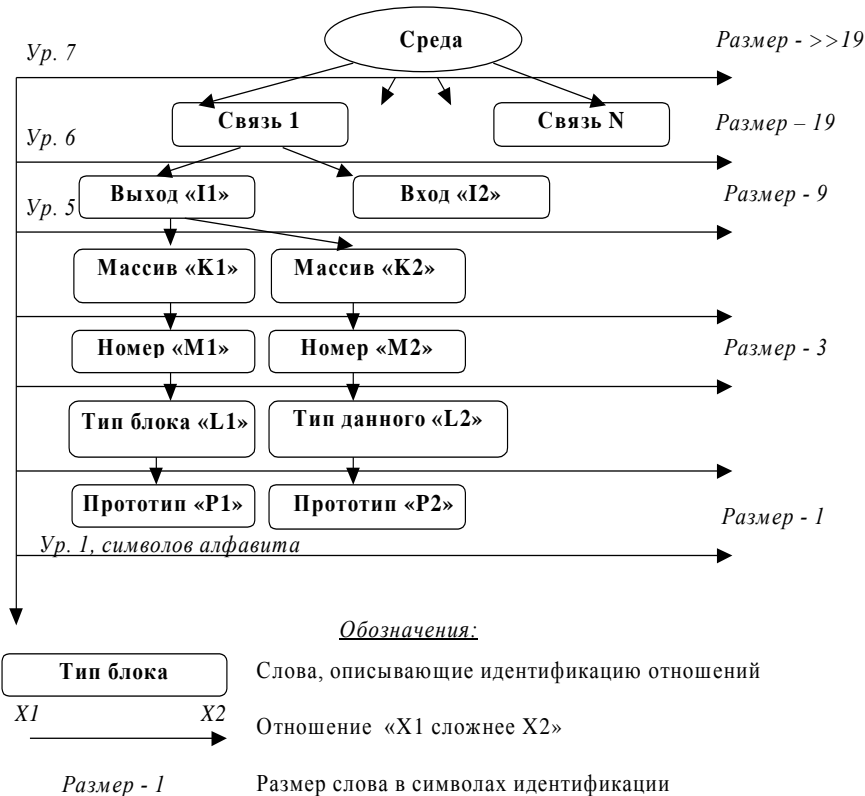


Рисунок 10 - Пример построения меры сложности на упрощенном описании внутренней среды блока-прототипа.

Построение меры КС проводится по следующей методике. Пусть дано:

- 1) $\mathbf{A} = \{A_l\}_{l=1, \overline{Np}}$ - совокупность оценок абсолютной КС по шкале $\mathbf{Ш}_a$ для ряда прототипов

$P_o = \{P_l\}_{l=1, N_p}$, полученные путем тестирования пользователей; данные оценки должны отражать сложность восприятия, проектирования и контроля моделирования моделей объектов;

2) Мера структурной сложности с неизвестными весовыми коэффициентами M_i^m .

Расчет коэффициентов меры КС производится по набору весов КС ряда тестовых примеров. Конкретный метод решения задачи построения абсолютной КС для произвольных прототипов зависит от числа примеров в наборе тестов.

9.2. Общая постановка задачи ограничения КС моделей

Пусть дано:

1) Некоторая замкнутая ограниченная модель внутренней среды прототипа P , представленная множеством экземпляров отношений различных уровней общности. В данное множество отношений входит, в частности:

- Свойство как совокупность значений $D_l = \{\partial_{li}\}_{i=1}^{Nd}$, где: ∂_{li} - идентификаторы значений свойства

D_l ;

- Подблоки данного блока-прототипа $B_t \in B = \{B_t\}_t$, где: B_{jt} - идентификатор подблока B_t , входящего в среду, t - номер в составе множества блоков B , входящих в среду; B_1 - внутренняя граница блока прототипа;

- Принадлежность свойства D_l границе B_t^e блока B_t , заданное как отношение:

$V_{tl} = (B_t^e, D_l)$;

- Граница блока внешняя (внутренняя) как множество отношений принадлежности свойств подблоку B_t среды S : $B_t^e = \{V_{tl}^e\}_l$ ($B_t^y = \{V_{tl}^y\}_l$);

- Связь s_i , т.е. причинно-следственное отношение или отношение эквивалентности значений, определенное над парой отношений принадлежности свойств границам блоков: $s_i = (V_{ik}, V_{ip})$;

- Среда S , т.е. множество причинно-следственных связей, определенных над множеством подблоков, входящих в среду: $S = \{s_i\}_{i=1, N}$.

Примечание. Все перечисленные отношения принадлежат иерархии:

- Уровней их агрегации, что задается совокупностью имен в цепочке идентификации типа блоков и типов свойств;

- Уровней их принадлежности к вышележащим структурным единицам, что задается совокупностью имен в цепочке идентификации вхождений.

2) Мера абсолютной КСП, включающая:

- Собственно меру АКСП, определенную над множеством отношений;

- Интервал допустимой КСП $\mathcal{E}_5 \leq KCP \leq \mathcal{E}_2$, где \mathcal{E}_5 и \mathcal{E}_2 - точная нижняя и точная верхняя грань предельно допустимой КСП для данной предметной области, выбранного языка описания и конкретного пользователя (группы пользователей);

- N - возможная степень превышения КСП U , обеспечивающая безошибочную работу оператора.

Получить: новую форму представления структуры модели P , входящей в интервал допустимой КСП.

Процесс упрощения модели, заданной в такой форме представления, предполагает два последовательно выполняемых действия:

1) Упрощение модели за счет выявления в ее составе фрагментов, составляющих структуру известных решений, хранящихся в базе данных системы и изначально имеющих допустимую КС, с последующей заменой фрагмента соответствующим идентифицированным решением.

2) В случае, если после «действия 1» для оставшихся фрагментов модели нет известного решения, следует выполнить упрощение прототипа автоматически или с участием пользователя, ориентируясь на доступный уровень КС моделей.

Целью нашего изложения будет определения алгоритма выполнения второго из двух вышеописанных действий.

9.3. Алгоритм решения задачи

Предлагается следующий алгоритм обеспечения допустимой КС представления моделей. Суть алгоритма состоит в изменении структуры модели путем перемещения части подблоков прототипа во вновь создаваемые подблоки с целью снижения общей КС модели до уровня, не превышающего заданной предельной верхней границы КС. Для любого «не присоединенного» подблока будем определять дополнительно такие свойства:

- «ценность», исходя из критерия близости, определяющего меру КС его общей части с ранее выбранным набором подблоков;
- «вес», исходя из КС части подблока, отличной от ранее выбранного набора подблоков.

Критерий близости блоков есть функция, производная от числа эквивалентных отношений различных уровней иерархии. Структура критерия определяется составом и весом отношений, определенных на этапе построения меры КС. В частности, состав эквивалентных отношений может характеризоваться количеством: K_s - связей по свойствам; K_t - имен в цепочке идентификации типа блока; K_d - наименования и типов свойств; K_z - значений свойств, заданных в порядке взаимного включения типов свойств (т.е.: время, пространство, все прочие свойства).

В этом случае критерий близости приобретет вид:

$$B = \sum_{i=1}^N K_i * W_i \quad (4)$$

Тут: W_i - известные веса когнитивной сложности различных типов отношений. Исходя из данного критерия, может формироваться множество оценок близости для блоков, имеющих в среде. Блок, имеющий наибольшую по величине меру близости с «пополняемым» блоком, объединяется с ним в единый логический блок.

Т.о. решается оптимизационная задача, близкая по своей постановке к классической «задаче о рюкзаке». Отличие предлагаемой постановки данной задачи заключается в динамически изменяемых ценности и весов предметов, складываемых в рюкзак.

Описание алгоритма.

1) Проверка исходной схемы P на предельно допустимую КС. Если условие $\mathcal{E}_5 \leq KСр \leq \mathcal{E}_2$ не выполняется, то переходим на пункт 2, иначе – на конец алгоритма, т.е. пункт 12.

2) Формирование пустого списка $Sп$, предназначенного для внесения в него извлекаемых из тела прототипа P подблоков.

3) Формирование на базе $Sп$ нового «пустого» блока-аккумулятора Π , имеющего пустой список свойств, составляющих внешнюю и внутреннюю его границу, и внесение его в B .

4) Формирование списка запрещенных блоков Sz и внесение в него внутренней границы прототипа блока B_I и блока-аккумулятора Π ;

5) Выбор из множества B набора блоков $\{Bs\}$, такого, что верно

$$\forall B_s \in B \setminus S_z \quad (8)$$

и выполняется условие максимальной близости Bs со списком $Sн$ по критерию

$$K(B_s \overline{\cap} S_{\Pi}) = \max_{B_I \in B} K(B_I \overline{\cap} S_{\Pi}) \quad (9)$$

где: $\overline{\cap}$ - операция определения общего подмножества отношений.

Если имеет место множество $\{Bs\}$, для которых выполняется (6), например на первом шаге алгоритма, когда список Π пуст, то выполняется переход на пункт 6, иначе на пункт 8.

6) Выбор из множества $\{Bs\}$, для которых выполняется (2), блока Bs' , для которого степень уменьшения КС блока B , получаемая за счет удаления блока Bs' максимальная среди всех B_i , принадлежащих B , т.е.

$$K(B \setminus B_s') = \max_{B_I \in B} K(B \setminus B_I) \quad (10)$$

где: \setminus - операция удаления общего подмножества отношений.

При этом:

- степень увеличения КС блока Π за счет блока Bs' максимальная среди всех B_i , принадлежащих B , т.е.

$$K(B_s \cup \Pi) = \max_{B_I \in B} K(B_I \cup \Pi); \quad (11)$$

- критерий близости B_s с блоком Π имеет максимальный вес, т.е.

$$K(B_s \cap \Pi) = \max_{B_I \in B} K(B_I \cap \Pi); \quad (12)$$

- критерий близости B_s со списком B имеет максимальный вес, т.е.

$$K(B_s \cap (B \setminus B_s)) = \max_{B_I \in B} K(B_I \cap (B \setminus B_s)). \quad (13)$$

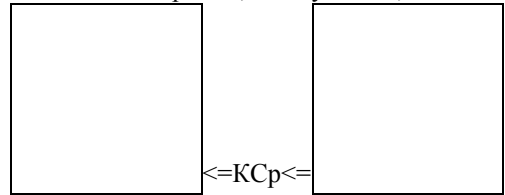
6) Внесение блока B_s в состав списка S_n .

7) Формирование на базе S_n нового блока-аккумулятора Π , имеющего список свойств, составляющих внешнюю и внутреннюю его границу, и внесение его в B ; при этом:

- формируется связи данного блока B_s с подблоками блока Π ;
- внутренняя и внешняя граница блока Π пополняется свойствами, посредством которых блок B_s связан с внешней средой;

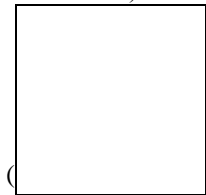
8) Изменение состава блоков и связей исходного прототипа P с учетом удаления из его состава найденного блока B_s и включения связей с новым блоком Π ;

9) Если $S_z = P$ то новый блок Π есть искомый и далее переход на конец алгоритма, т.е. пункт 12;



10) Оценка на КС нового блока Π : если условие выполняется, то идти на выбор следующего вторичного блока, т.е. на пункт 5, иначе на пункт 11;

11) Т.к. выбранный нами новый подблок для Π превысил необходимый уровень



($\leq K_{Cr}$), то делается:

- возврат на предыдущий вариант S_n и соответствующему ему виду P и Π ;
- внесение блока B_s в список запрещенных блоков S_z ;
- переход на выбор следующего вторичного блока, т.е. на пункт 5;

12) Формирование нового «пустого» блока Π , т.е. имеющего пустой список подблоков и свойств, составляющих внешнюю и внутреннюю его границу, и внесение его в B .

10. Изобретение новых решений – как инструмент сужения числа аксиом грамматики, построенной путем обобщения прототипов

Общий подход к синтезу гипотез предполагает, что при склеивании прототипов автоматически формируется декартово произведению всех составляющих всех ИЛИИ-синтермов, входящих в И-ИЛИИ-дерево. Данное И-ИЛИИ-дерево с определенными над ним продукционными зависимостями, составляет динамическую базу данных. Результатом ограничений на каждом этапе является все более суженное "полное" И-ИЛИИ-дерево.

Увеличение числа вводимых прототипов ведет к возникновению все большего числа частей блоков и фрагментов. Можно сделать выводы, что:

1. Количество вариантов сред чрезвычайно большое и пользователь не в состоянии их проверить их на достоверность;

2. Нужны ограничения технологического и семантического характера, способные резко снизить число потенциально возникающих прототипов при присоединении нового прототипа, отсекая решения, не имеющие смысл и решения, проверить адекватность которых для данного пользователя не представляется возможным.

В связи с этим, для решения данной задачи предлагается технология отсечения слишком сложных и не имеющих смысл решений, схема которой показана на рис. 11.

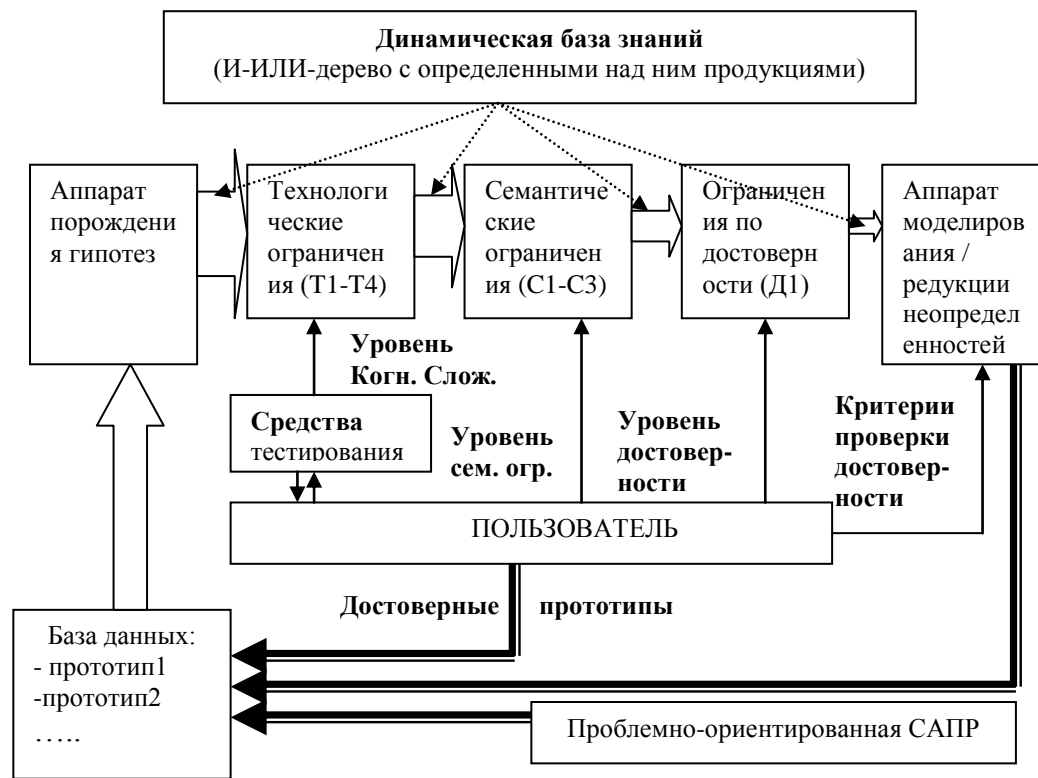


Рисунок 11 - Схема технологии синтеза и отбора гипотез в МЭО

11. Построение системы интерфейсов «Язык предметной области ↔ Язык формальных спецификаций соответствующего уровня абстракции»

Цель построения системы интерфейсов «Язык предметной области ↔ Язык формальных спецификаций соответствующего уровня абстракции»:

- обучения базы знаний методикам проектирования по прототипам, накопленным в проблемно-ориентированном САПР;

- передачи готового решения из инструментальной оболочки в проблемно-ориентированный САПР.

Модель предлагаемого интерфейса может быть представлена следующим образом:

$M=(Go, Fo, Mo, So, Gs, Fs, Ms, Ss, Pos, Pso)$, где:

Go - грамматики языков представления моделей в ИО;

Fo - формат внутренних структур данных представления моделей в ИО;

Mo - описания прототипов в данной проблемной области на языках ИО (знания экспертов, проекты);

So - описания прототипов в данной проблемной области в формате внутренних структур данных МЭО;

Gs - грамматики языков представления моделей в других инструментальных средствах проектирования сложных систем (САПР);

Fs - внутренние структуры данных представления моделей других САПР в данной проблемной области;

Ms - описания прототипов в данной проблемной области на языках прочих САПР (знания экспертов, проекты);

Ss - описания прототипов в данной проблемной области в формате внутренних структур данных прочих САПР;

Pos - процедуры отображения Mo, So в Ms, Ss;

Pso - процедуры отображения Ms, Ss в Mo, So.

Охарактеризуем формы представления моделей в ИО:

1. Грамматика языка представления моделей в ИО (Go) описана в [7].

2. Модель структур во внутреннем формате (So), представляется в виде описаний библиотек, типов, массивов и т.п. в формате DBF.

3. Функциональные модели, задающие соответствия для базовых структурных блоков и связей, представляются в виде динамических недоопределенных вычислительных моделей, описание которых совмещено с описанием структур (So) и представлено в формате DBF.

Заклучение

В работе описаны структура средств и методов работы с формальными грамматиками в рамках С КМ ПрО МЭО. Построение данной КМ выполнено автором в рамках работ [1-50]. Данная КМ является основой построения соответствующего инструментального комплекса по автоматизации построения интеллектуальных САПР некоторого ограниченного класса – мета-эвристической оболочки (МЭО).

Наибольшее влияние на методы построения КМ оказала выбранная форма ее представления – семиотическая модель. А так как главный компонент СМ – это формальные грамматики, то без преувеличения можно сказать, что суть КМ – это набор методов работы с формальными грамматиками, определенными в рамках СМ.

Перспективой работы является дальнейшая работа над расширением предложного комплекса средств и методов.

Литература

1. Григорьев А.В. Специализированная оболочка для синтеза интеллектуальных САПР и АСНИ. // Информатика, кибернетика и вычислительная техника (ИКВТ-97). Сборник трудов ДонГТУ, Выпуск 1. Донецк: ДонГТУ, 1997. С. 225-228.
2. Григорьев А.В. И/ИЛИ-дерево как средство абстрактного представления базы знаний // Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка, випуск 39: – Донецьк, ДонНТУ, 2002. - С. 36–42.
3. Григорьев А.В. Комплекс моделей САПР как система взаимосвязанных уровней о действительности. // Научные труды Донецкого государственного технического университета. Серия "Информатика, кибернетика и вычислительная техника", (ИКВТ-2000) выпуск 10. - Донецк, ДонГТУ, 2000. - С. 155-167.
4. Григорьев А.В. Семиотическая модель базы знаний САПР. // Научные труды Донецкого государственного университета. Серия "Проблемы моделирования и автоматизации проектирования динамических систем". Выпуск 10: - Донецк: ДонГТУ, 1999. - С. 30-37.
5. Представление недоопределенности знаний в инструментальной оболочке для построения САПР. // Искусственный интеллект. N 6, 1999, С. 56-66.
6. Григорьев А.В. Адаптивная система ограничений на сложность при синтезе новых решений в интеллектуальных САПР // Искусственный интеллект. – Донецк, 2001 – N 2 – С. 152 – 167.
7. Григорьев А.В. Унифицированная концептуальная модель предметной области. // Информатика, кибернетика и вычислительная техника (ИКВТ-97). Сборник трудов ДонГТУ, Выпуск 1. Донецк: ДонГТУ, 1997. С.218-224.
8. Григорьев А.В. Концептуальная модель оболочки для построения интеллектуальных САПР вычислительной техники как средство предметной адаптации. В кн. Наукові праці Донецького національного технічного університету. Серія „Інформатика, кібернетика та обчислювальна техніка” (ІКОТ-2009). Випуск 10(153).- Донецьк: ДВНЗ «ДонНТУ». - 2009. – С.255-265.
9. Григорьев А.В. Вербальная модель предметной области для интеллектуальных САПР // Наукові праці Донецького державного технічного університету. Серія "Обчислювальна техніка та автоматизація". Випуск 20. - Донецьк, ДонДТУ, 2000. - С. 171-180.
10. Григорьев А.В. Содержание некоторых категорий абстракций в теории построения интеллектуальных САПР. // Наукові праці Донецького національного технічного університету / Редкол.: Башков Є.О. та інші. Серія: "Обчислювальна техніка та автоматизація": Випуск 64.: Донецьк: Видавництво ДонНТУ, 2003 - С. 166-178.
11. Григорьев А.В. Семантика модели предметной области для интеллектуальных САПР. // Научные труды Донецкого государственного технического университета. Серия "Информатика, кибернетика и вычислительная техника", (ИКВТ-2000) выпуск 10. - Донецк, ДонГТУ, 2000. - С. 148-154.
12. Григорьев А.В. Специфика выполнения теоретико-множественных операций над контекстно-свободными грамматиками в условиях различных форм дополнительных семантических правил в семиотической модели интеллектуальных САПР Научные труды Донецкого национального технического университета. Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП – 2006). Выпуск 5 (116). – Донецк: ДонНТУ, 2006. – С. 91-104.
13. Григорьев А.В. Алгоритм выполнения теоретико-множественных операций над грамматиками в среде специализированной оболочки для создания интеллектуальных САПР // Наукові праці національного технічного університету. Серія «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП -2002). Выпуск 52: Донецк: ДонНТУ, 2002. - С.83-93.
14. Григорьев А.В. Теоретико-множественные операции над грамматиками как механизм работы со знаниями в интеллектуальных САПР // Труды Восточно-украинского технического университета. – Луганск, ВУТУ, 2002. – С. 186–194.

15. Григорьев А.В. Методы построения функций в специализированной оболочке для создания интеллектуальных САПР // Искусственный интеллект. – Донецк, 2001 – №3 – С. 40–53.
16. Григорьев А.В. Обобщение знаний в интеллектуальной системе с семиотической моделью представления знаний // Научные труды Донецкого государственного технического университета. Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 29. – Севастополь: «Вебер», 2001. – С. 114–120.
17. Григорьев А.В. Управление движением объектов в семиотической модели предметной области // Наукові праці національного технічного університету. Серія «Обчислювальна техніка та автоматизація». Випуск 48: Донецк: ДонНТУ, 2002. - С.280-287
18. Григорьев А.В. Организация вывода решений в базе знаний инструментальной оболочки для создания интеллектуальных САПР / Наукові праці Донецького національного технічного університету. Серія "Проблеми моделювання та автоматизації проектування динамічних систем" (МАП-2005). Випуск: 78 - Донецьк: ДонНТУ. - 2005 - с. 171-182.
19. Григорьев А.В. Организация пространственного и временного логического вывода в концептуальной модели интеллектуальных САПР. В кн. Наукові праці Донецького національного технічного університету. Серія „Інформатика, кібернетика та обчислювальна техніка” (КОТ-2008). Випуск 9(132).- Донецьк: ДонНТУ. - 2008. – С.296-311.
20. Григорьев А.В. Обеспечение монотонности вывода и верификация баз знаний в инструментальной оболочке для создания интеллектуальных надстроек над САПР. // В кн. Наукові праці Донецького національного технічного університету. Серія „Інформатика, кібернетика та обчислювальна техніка” (КОТ-2010). Випуск 11(164).- Донецьк: ДВНЗ «ДонНТУ». - 2010. – С.161-164.
21. Григорьев А.В. Методы решения задачи структурного синтеза в интеллектуальных САПР, построенных на основе семиотической модели структур. В кн. Наукові праці Донецького національного технічного університету. Серія „Обчислювальна техніка та автоматизація”. Випуск 171(19) / Редкол.: Башков С.О. (голова) та ін. - Донецьк: ДонНТУ, 2010. – С.128-140.
22. Григорьев А.В. Система автоматизированного решения вычислительных задач в САПР, основанная на методе программирования в ограничениях. В кн. Наукові праці національного технічного університету. Серія «Проблеми моделювання та автоматизації проектування динамічних систем. (МАП-2011). Випуск 9 (129): Донецьк: ДонНТУ. - 2011. – С. 43-51.
23. Григорьев А.В. Логические системы представления знаний и вывода в искусственном интеллекте. // Научные труды Донецкого государственного технического университета. Серия "Информатика, кибернетика и вычислительная техника", (ИКВТ-2000) выпуск 10. - Донецк, ДонГТУ, 2000. - С. 225-230.
24. Григорьев А.В. Ограничение когнитивной сложности моделей. // Прогрессивные технологии и системы машиностроения.: Международный сб. научных трудов. - Донецк: ДонГТУ, 2000. Випуск. 10 - С. 49-58.
25. Григорьев А.В. Методика тестирования для определения когнитивной сложности моделей различных предметных областей. // Научные труды Донецкого государственного технического университета. Серия: Информатика, кибернетика и вычислительная техника, выпуск 6 (ИКВТ-99): - Донецк: ДонГТУ, 1999. - С. 246-251.
26. Григорьев А.В. Оценка когнитивной сложности моделей. // Научные труды Донецкого государственного технического университета. Серия: Информатика, кибернетика и вычислительная техника, выпуск 6 (ИКВТ-99): - Донецк: ДонГТУ, 1999. - С. 252-259.
27. Григорьев А.В. Изобретение как метод автоматизации процесса обучения методикам проектирования в семиотической модели САПР. // Информатика и кибернетика. – Д.: ДонНТУ, – 2015. – № 1. - С. 51- 66.
28. Григорьев А.В. Анализ эффективности и перспектив развития методов построения двусторонних трансляторов в задаче создания интеллектуальных надстроек над проблемно-ориентированными САПР. В кн. Наукові праці національного технічного університету. Серія «Обчислювальна техніка та автоматизація». Випуск 20 (182). –: Донецьк: ДонНТУ, 2011. – С. 116-129.
29. Григорьев А.В. Анализ существующих способов создания интерфейса «языки формальных спецификаций — проблемно-ориентированные языки». В кн. Серія „Інформатика, кібернетика та обчислювальна техніка” (КОТ-2011). Випуск 14(185).- Донецьк: ДВНЗ «ДонНТУ», 2011.270-275.
30. Григорьев А.В. Интерфейс табличного процессора EXCEL и специализированной оболочки для синтеза интеллектуальных САПР и АСНИ. // Информатика, кибернетика и вычислительная техника (ИКВТ-97). Сборник трудов ДонГТУ, Випуск 1. Донецк: ДонГТУ, 1997. С. 229-238.
31. Нариньяни А.С. Недоопределенность в системах представления и обработки знаний. // Известия АН СРСР. Техническая кибернетика. - 1986.- № 5. – С. 3-28.
32. Д.В. Банасюкевич, И.Д. Гофман, Д.А. Инищев, А.С. Нариньяни. Интеллектуальная система планирования и управления проектами на базе недоопределенной математики. В кн.: “КИИ '2000 седьмая

национальная конференция по искусственному интеллекту с международным участием”: Том 1. Москва, 2000, С.. 617 — 624.

33. Нариньяни А.С. Не-фактори: неточность и недоопределенность – расхождение и взаимосвязь. Известия академии наук. Теория и системы управления, 2000, №5, С. 44-56.

34. Норенков И.П. Разработка систем автоматизации проектирования. М.: МГТУ им. Э.Н.Баумана, 1994. – 207 с.

35. Солодовников В.В., Тумаркин В.И. Теория сложности и проектирование систем управления. - М.: Наука. 1990. - 186 с.

36. . Поспелов Д.А. Ситуационное управление: теория и практика. М.: Наука, 1986. - 288 с.

Григорьев А.В. Формальные грамматики в семиотической концептуальной модели предметной области. Рассмотрен комплекс методов работы с формальными грамматиками, определенных в рамках концептуальной модели предметной области инструментальной оболочки для автоматизации построения интеллектуальных САПР ограниченного класса. Работа носит итоговый характер.

Ключевые слова: формальные грамматики, концептуальная модель предметной области, семиотическая модель, САПР.

Grigoriev A.V. Formal grammar semiotic conceptual domain model. The complex of methods of work with formal grammars defined within the conceptual model of the instrumental domain shell for building intelligent automation CAD limited class. The work is the final character.

Keywords: formal grammar, conceptual domain model, the semiotic model of CAD.

Механизм срабатывания модулей знаний интеллектуальной оболочки

О.В. Морозова
Донецкий национальный технический университет
olmalyavka@gmail.com

Морозова О.В. Механизм срабатывания модулей знаний интеллектуальной оболочки. Рассмотрена проблема проектирования локальной сети со стороны эксперта предметной области (системного администратора). Рассмотрена семантика предметной области. Рассмотрены достоинства и недостатки современных программных продуктов для проектирования локальных сетей. Представлена иерархия модулей знаний. Разработана машина логического вывода срабатывания модулей знаний. Описаны особенности продукционной базы знаний. Выявлены достоинства модульной продукционной базы знаний.

Ключевые слова: локальная сеть, модуль знаний, продукционная база знаний, машина логического вывода.

Введение

Современные информационные потоки, которые представляют собой передачу информации от одного объекта среды к другому, в различных предметных областях многократно возрастают с каждым годом. Поэтому необходимо четкое взаимодействие всех потоков во всех областях.

Для регулирования и упорядочивания информационных потоков служат вычислительные сети различного назначения, от локальных частных сетей, корпоративных до глобальных мировых сетей. Предметной областью является локальная вычислительная сеть, в которой происходит взаимодействие компонентов различных предметных областей.

Локальная вычислительная сеть позволяет объединить персональные компьютеры для получения максимальной эффективности:

- со стороны использованных ресурсов (оборудования, данных, программного обеспечения, процессов);
- для объединения одинаковых функций для выполнения разными узлами сети;
- для соединения всех компьютеров в единое целое без промежуточных устройств.

Главное достоинство сети — это легкость в модификации и расширении готовой локальной вычислительной сети. Для этого служат готовые схемы проектирования.

Современные локальные сети, как промышленного предприятия, так и частного пользования проектируются системным администратором. В данной предметной области (локальные сети) системный администратор выступает экспертом предметной области и только его опыт и знания гарантируют качественное и эффективное функционирование сегмента сети. Поэтому для облегчения построения и эксплуатации локальной сети необходимо добавление в современные промышленные системы автоматизированного проектирования (САПР) соответствующей интеллектуальной надстройки [1].

Интеллектуальная надстройка объединяет в себе взаимодействие трех видов моделей предметной области, такие как функциональную, процессную и информационную. Логика представления основных структур базы данных – это информационная модель.

Процесс обработки причинно-следственных отношений информации об объекте предметной области является процессной моделью. Функциональная модель организует взаимосвязь объектов предметной области с объектами реального мира.

Цель статьи:

- рассмотреть проблемы проектирования локальных вычислительных сетей со стороны эксперта предметной области и предложить пути их решения с помощью интеллектуальной надстройки;
- рассмотреть алгоритм и метод реализаций организации уровней модулей знаний и блоков представления знаний;
- выявить достоинства и недостатки современных программных продуктов;
- привести структуру взаимодействия блоков;

– рассмотреть семантику представления предметной области и семантику решения задачи проектирования локальной сети.

Анализ современных программных продуктов для создания локальных сетей

Рассмотрим существующие программные продукты Network Notepad, EDraw Network Diagram, LanFlow, 10-Strike LANState. Проанализируем и выявим достоинства и недостатки из которых составим список требований и критериев, которым должна отвечать интеллектуальная оболочка.

Network Notepad программа для создания интерактивных сетевых схем [2].

Достоинства системы Network Notepad:

- быстрое составление физической схемы сети;
- возможность применять значки сетевого оборудования (маршрутизаторы, коммутаторы, и др.);
- поддержка метода Drag&Drop.

Недостатками можно выделить:

- ориентация на сетевых специалистов;
- нет автоматизации проектирования;
- инженер самостоятельно принимает решения при создании проекта локальной сети.

EDraw Network Diagram - это программа для построения диаграмм представление физической схемы локальной сети.

Достоинства системы EDraw Network Diagram:

- основные виды представления локальных сетей;
- топологические схемы сети в виде диаграмм;
- проектирование сетей определенной фирмы производителя Cisco;
- имеет разнообразные примеры для создания диаграмм, символов и шаблонов.

Из недостатков можно выделить:

- ориентация на сетевых специалистов;
- нет автоматизации проектирования;
- инженер самостоятельно принимает решения при создании проекта локальной сети.

10-Strike LANState - программа для администрирования и мониторинга локальных вычислительных сетей.

Достоинства системы 10-Strike LANState:

- наблюдение текущего состояния сети в графическом редакторе;
- мониторинг удаленных станций сети с помощью периодического опроса компьютеров;
- работа с сетью в реальном времени.

Из недостатков можно выделить:

- мало подходит для построения проекта локальной сети;
- большее ориентировано на работу уже с построенной сетью и реальным оборудованием;
- в основном применяется для администрирования существующей сети.

LanFlow - программа для проектирования и документирования сетей.

Достоинства системы LanFlow:

- создание схем локальной сети;
- представление сетевого оборудования в виде 2D и 3D символов;
- добавление клипарта для специфики сети.

Из недостатков можно выделить:

- для работы с программой необходимы специализированные знания;
- закрытая база компонентов.

Проектирование локальных вычислительных сетей

Задачи, которые необходимо решить эксперту предметной области (системному администратору) [3] для создания оптимальной схемы взаимодействия устройств:

- выбор релевантного оборудования;
- установка связей между всеми устройствами сети;
- назначение прав доступа для каждого устройства сети;
- регулирование передачи пакетов в сети между устройствами;
- назначение адресов узлов в сети, проверка на уникальность адресов в данном сегменте сети;
- отслеживание ошибок настройки сети и работоспособности всех компонентов без сбоев;
- рассчитать возможности перехода на другое телекоммуникационное оборудование;
- обеспечение защиты данных в сети.

Для проектирования физической модели локальной сети (см. рис. 1) системному администратору придется учитывать много параметров:

- количество этажей;
- количество комнат на этаже с параметрами комнат (длина, ширина и высота этажа);
- количество рабочих мест;
- сетевое оборудование для соединения устройств сети;
- серверное оборудование;
- и др.

Такой объем выполняемой работы проектирования (или модернизации) одной сети будет выполняться системным администратором в течении нескольких дней и в несколько этапов. На первом этапе будет определена топология сети и разработан порядок расстановки рабочих станций. На втором этапе осуществлён просчет параметров комнат и этажей для оборудования, и выбор оптимального сетевого оборудования (коммутаторов, маршрутизаторов, серверного оборудования и т.д.). На третьем этапе будет рассчитана экономическая составляющая проекта. Заключительный этап - это монтаж сети.

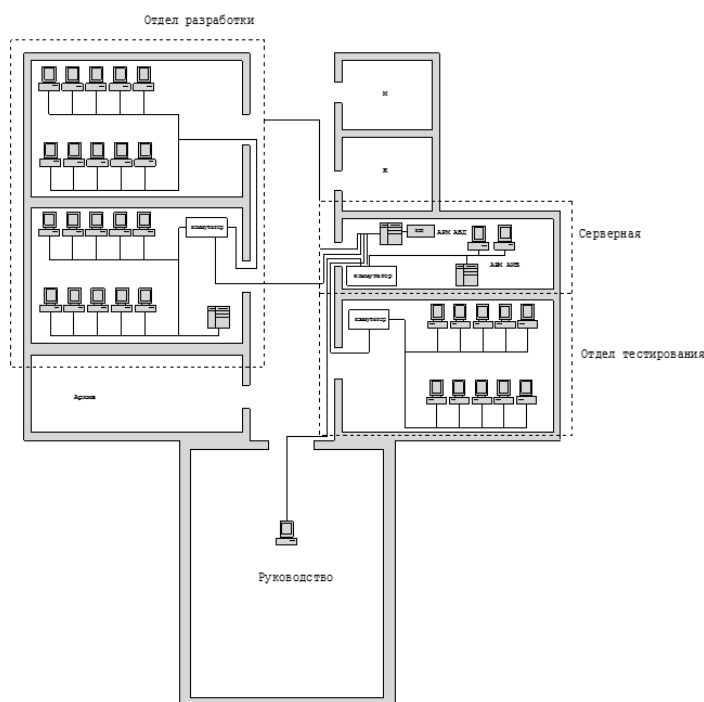


Рисунок 1 – Пример физической модели локальной сети

Для ускорения и эффективного проектирования сети можно применить САПР с интеллектуальной надстройкой именно на первом и втором этапах. Так как существует множество САПР различного назначения от графического представления до просчета экономичности [1], рассмотрим формирование и срабатывание базы знаний из готовых прототипов сети и модернизации их до конкретных сетей общего и частного пользования.

Структура продукционной базы знаний

Оптимизация построения локальной сети в интеллектуальной оболочке происходит так же в два этапа. Первый этап - организация диалога в виде набора правил и предъявления требований к структуре сети (самоинтервью) [4]. На втором этапе осуществляется автоматическая параметризация и выбор оптимального оборудования.

Такой метод проектирования сетей позволяет абстрагироваться от конкретного оборудования и позволяет построить структурную модель сети.

Для выявления всей полноты информации о предметной области и описания ее в иерархическом структурированном виде будем использовать модульный принцип организации базы знаний. При таком подходе происходит покрытие всех требований, накладываемых на предметную область.

Дескриптивная модель позволяет описать топологию взаимодействия сети и информацию, передаваемую в данной локальной сети. Она позволяет выбирать оптимальный метод решения поставленной задачи и построить формальное представление задачи.

Формирование базы знаний будем производить на основании самоинтервью. Множество фактов хранимых в базе знаний имеют тип и прототип [5]. Прототип – это отдельно взятое решение, которое имеет вид факта. Факт может быть единичный, базовый, или – сложный. Сложный факт задает множество фактов, если

решение формируется как набор из базового множества фактов или модификация этого множества. Множество решений, составляющих тип - это И-ИЛИ-дерево. В типе есть неизменная часть, т.е. - системообразующая, которая принадлежит любому прототипу, и – факультативная часть, как набор разнообразных частей.

Тип по отношению к типу блока – это множество:

- структур блока, если блок не базовый;
- функций, если блок базовый.

Используется модульная база знаний [6].

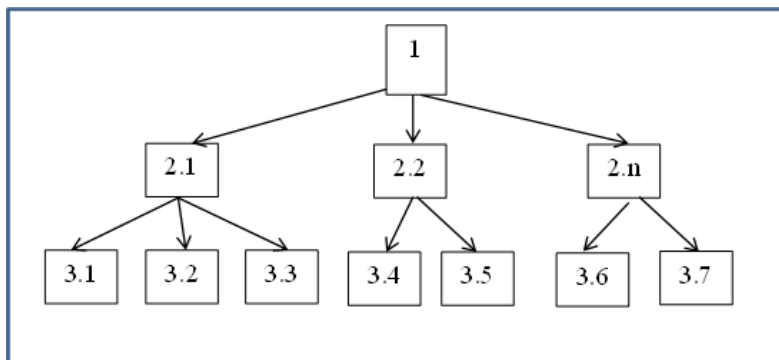


Рисунок 2 – Иерархия модулей знаний

Тип – это модуль знаний, прототип – отдельно взятое решение.

Если модуль знаний делится на подмодули (внешняя граница, внутренняя и т.д.), то так же имеется и набор фактов - база данных этого же уровня. Совокупность прототипов, составляющих тип, может явно не задаваться перечислением и не входить в базе данных в состав модуля знаний, например, типа блока. Тогда в глобальной базе знаний имеется только тип блока как модель знаний.

Список прототипов или фактов всех уровней сложности, включая готовые блоки в сборе, варианты внешних границ, варианты внутренних наборов подблоков, варианты групп связей разных форм (внутренние и т.д.) формируется в процессе вывода.

Каждый модуль знаний представляет собой независимую структурную единицу. При этом дробление каждого модуля возможно на подмодули. Связь устанавливается в зависимости от границ входов и выходов. Если в модуле отсутствуют выходы, он является конечным. Для хранения всех границ используется таблица границ входов и выходов (см. рис. 3). Для обозначения связей между модулями используется таблица связей. В ней указываются порядковые номера входов и выходов.

Система поддерживает два типа связей:

- звезда;
- последовательное соединение.

Механизм логической машины вывода

При срабатывании интеллектуальной надстройки, генерируется базовый блок первого уровня. В зависимости от условий, он имеет определённое количество входов и выходов [7].

В течении работы логической машины вывода, внутри базового блока синтезируются подблоки. Так же происходит соединение входов и выходов внутренней границы базового подблока первого уровня, с внешними границами синтезированных подблоков второго уровня.

На следующем этапе работы логической машины происходит переход по связи из блока первого уровня в блок второго уровня. Теперь мы можем произвести синтез подблоков третьего уровня внутри блока второго уровня.

Далее происходит соединение входов и выходов внутренней границы блока второго уровня с внешними границами синтезированных блоков третьего уровня. Данная операция синтеза может повторяться, до тех пор, пока не будет достигнут конечный блок, не имеющий внутренних границ входов и выходов (см. рис. 3).

При создании базы знаний, пользователь сам указывает тип связи. После подачи команды, задаётся узловой блок, с которым устанавливаются все последующие связи. При поступлении нового блока, указатель смещается на него, и пока не будет дана команда отмены, данный блок является узловым (см. рис 4).

В таблице связей указываются переходы от одного блока к другому. Для учёта берутся порядковые номера записей из таблицы входов и выходов, и заносятся в поля. Первое поле указывает на источник связи, а второе на приёмник.

Использование модульной базы знаний и минимального набора операторов позволяют редактировать непосредственно каждый блок. При вводе новых продукций необходимо придерживаться точного порядка следования правил. При работе машины вывода, происходит построчное чтение базы. Если допустить ошибку при вводе правил, может не сработать условие и либо произойдёт аварийное завершение работы машины вывода, либо в построенной топологии будут присутствовать ошибки.

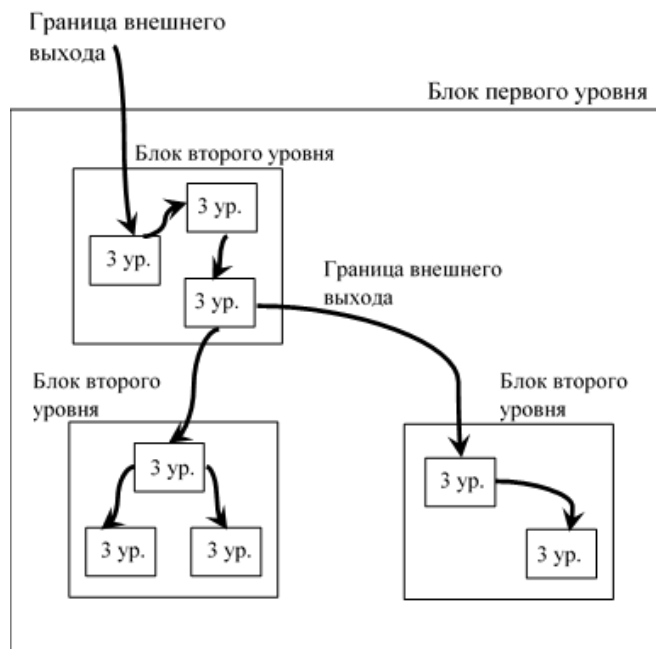


Рисунок 3 – Организация уровней связи блоков

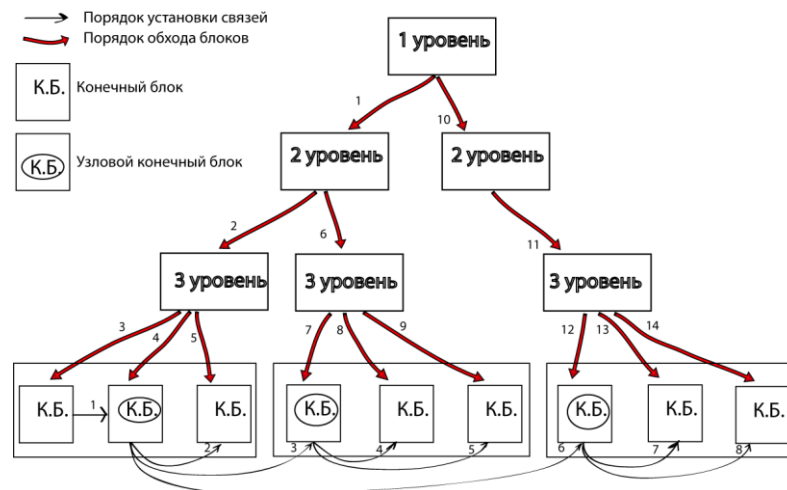


Рисунок 4 – Иерархия связи блоков

При работе машины вывода, происходит построчное чтение продукций в текущем модуле знаний. В зависимости от типа первого символа, выполняются определённые продукции. При переходе в новый подблок, работа с текущей базой останавливается. Возобновление работы происходит после возврата в блок, когда обработаны все нижестоящие подблоки. Работа с блоком возобновляется со следующей строки после команды перехода на блок нижнего уровня.

Во время работы с блоками, возможен доступ к структурам машины вывода. В данных структурах хранится информация о доступных блоках, количестве загрузок каждого блока, в каком блоке находимся в текущий момент.

При возникновении ситуации, когда модуль полностью прочитан, но остались необработанные продукции, работа машины вывода останавливается, и пользователю выдаётся ошибка о нарушении целостности базы.

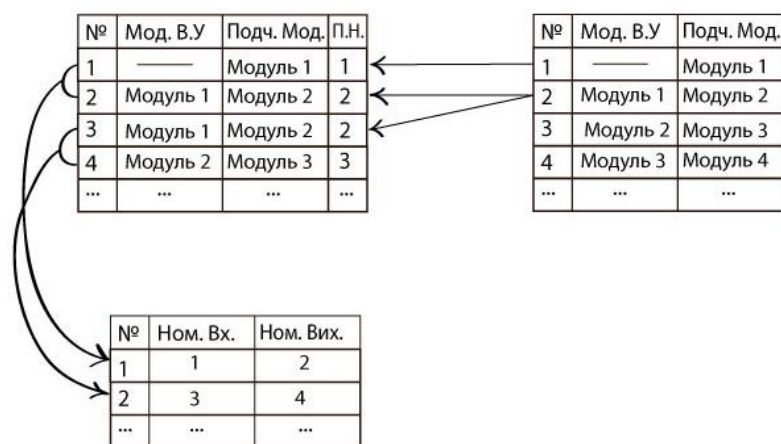


Рисунок 5 – Схема взаимодействия модулей знаний в базе данных

Выводы

В статье был проведен анализ проблемы проектирования локальных вычислительных сетей на основе эксперта предметной области. Проанализированы современные программные системы для проектирования сети и выявлены в них достоинства и недостатки. Предложен процесс упрощения и эффективного проектирования локальных сетей при использовании интеллектуальной надстройки для систем автоматизированного проектирования. Была описана структура продукционной базы знаний в виде иерархии модулей знаний. Представлен механизм логической машины вывода и организация взаимосвязей блоков данных в интеллектуальной надстройке. Выявлены достоинства модульной продукционной базы знаний:

- отдельный модуль знаний, позволяет организовать быстрый поиск в иерархии модулей знаний и по дереву решений;
- возможность многократного использования модуля знаний на разных уровнях логической машины вывода.

Перспективой дальнейшего развития работы является создание и описание нового метода проектирования локальных сетей на основе топология «звезда» и «иерархическая звезда». Так же необходимо будет учесть функциональные зависимости объекта сети с ее структурными элементами.

Литература

1. Крылов К.С. Синтез проектов локальных вычислительных сетей с использованием экспертных методик/ К.С. Крылов, А.В. Григорьев // Труды конференции ИУС и КМ -2011 том третий. – Донецк, 2011. – 200 с.
2. Автоматизированное проектирование СКС и ЛВС NetWizard [Электронный ресурс]. - Электрон. дан. – Режим доступа <http://netwizard.ru/>
3. Олифер, Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е издание / В. Г. Олифер // СПб.: Питер, 2006 – С. 45-79.
4. Григорьев, А.В. Создание интеллектуальной оболочки по методикам проектирования локальных сетей при использовании проектировщика как эксперта по знаниям / А.В. Григорьев, О.В. Морозова // Наукові праці ДонНТУ, Серія: «Проблеми моделювання та автоматизації проектування». Вип. №1(10)-2(11). – Донецьк, 2012. – С. 144-155.
5. Солодовников В.В. Теория сложности и проектирование систем управления/ В.В. Солодовников, В.И. Тумаркин// М.: Наука, 1990. – 168 с.
6. Ильянсов Б. Г. Модульная структура базы знаний экспертной системы «Выбор специальности при поступлении в вуз»/ Б. Г. Ильянсов, Е. Б. Старцева, Н. Р. Янгуразова/ Электронный журнал Нефтегазовое дело, 2006 –С. 4-5.

7. Морозова О.В. Анализ методов построения экспертных систем в продукционных инструментальных оболочках / О.В. Морозова, А.В. Григорьев // Наукові праці ДонНТУ. – Донецьк, 2012. – Вип. 16(204). – С. 180-191.

8. Григорьев А.В. Анализ существующих способов создания интерфейса «языки формальных спецификаций — проблемно-ориентированные языки» / Григорьев А.В., Морозова О.В. // Сборник научных трудов донецкого национального технического университета. – Донецк, 2011. – № 14. – С. 270 – 275.

9. Григорьев А.В. Построение двухсторонних трансляторов в задаче создания интеллектуальных надстроек над проблемно-ориентированными САПР / Григорьев А.В., Морозова О.В. // Сборник трудов XI международной научной конференции им. Т.А. Таран. - Киев: Просвита, 2011. - С. 68-75.

10. Люгер, Д. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. 4-е издание / Пер. с англ. – М.: «Вильямс», 2003 -С. 219-262.

11. Гаврилова, Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский СПб.: Питер, 2000 –С. 71-73.

12. Голдовский, П. Knowledgeware: как задействовать самые ценные корпоративные активы [Электронный ресурс] / Павел Голдовский, Андрей Бубнов // САПР и графика. - Электрон. журн. - №4'2001. - Режим доступа к журн.: <http://www.sapr.ru/article.aspx?id=7125&iid=291>

Морозова О.В. Механизм срабатывания модулей знаний интеллектуальной оболочки. Рассмотрена проблема проектирования локальной сети со стороны эксперта предметной области (системного администратора). Рассмотрена семантика предметной области. Рассмотрены достоинства и недостатки современных программных продуктов для проектирования локальных сетей. Представлена иерархия модулей знаний. Разработана машина логического вывода срабатывания модулей знаний. Описаны особенности продукционной базы знаний. Выявлены достоинства модульной продукционной базы знаний.

Ключевые слова: локальная сеть, модуль знаний, продукционная база знаний, машина логического вывода.

Morozova OV Response Mechanism of intellectual shell of knowledge modules. The problem of designing a LAN by subject area (system administrator) expert. We consider the semantics of the subject area. The advantages and disadvantages of modern software products for the design of local networks. It presents a hierarchy of knowledge modules. A car trip inference of knowledge modules. The features production base of knowledge. Revealed the dignity of modular production base of knowledge.

Keywords: LAN, knowledge module produktionnaja knowledge base, inference engine.

СЕКЦИЯ «ПРИКЛАДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИНЯТИЕ РЕШЕНИЙ»

УДК 004.89

Прототипирование продукционных баз знаний на основе диаграмм классов

Н.О. Дородных, А.Ю. Юрин

Институт динамики и теории управления им. В.М. Матросова СО РАН (ИДСТУ СО РАН)

Иркутский национальный исследовательский технический университет (ИрНТУ)

iskander@icc.ru

Дородных Н.О., Юрин А.Ю. Прототипирование продукционных баз знаний на основе диаграмм классов. В работе описана программная система для прототипирования продукционных баз знаний – Personal Knowledge Base Designer. Система ориентирована на непрограммирующего специалиста и позволяет автоматизировать процесс создания баз знаний благодаря: интеграции с CASE-средствами IBM Rational Rose и StarUML, в части импорта UML-моделей, генерации CLIPS-кода и тестирования разработанных баз знаний путем запуска машины вывода. Приведены результаты апробации программной системы при решении учебных (тестовых) задач.

Ключевые слова: программная система, базы знаний, правила, концептуальная модель, генерация кода, CLIPS, UML, диаграммы классов.

Введение

Сложность и трудоемкость процесса разработки экспертных систем (ЭС) обусловлена, главным образом, сложностью и трудоемкостью этапа разработки баз знаний (БЗ), который включает задачи по формализации предметных знаний и их описанию на определенном языке представления знаний (ЯПЗ) [1]. Повышение эффективности решения данных задач, путем их автоматизации, обуславливает необходимость разработки специализированных программных средств. Подобные программные средства в виде специализированных редакторов БЗ (например, Visual Expert System Designer, Expert System Designer, ES-Builder и др.), позволяют повысить эффективность процесса разработки за счет использования визуального моделирования, шаблонов представления знаний, автоматизации процесса верификации и генерации кода БЗ.

Одним из распространенных источников знаний являются концептуальные модели, создаваемые в процессе проектирования программного обеспечения, анализа и моделирования предметной области. Данные модели, представленные в форме диаграмм (DFD, IDEF0, IDEF5, UML и др.) (Рис. 1), являясь результатом этапов идентификации и концептуализации, в большинстве случаев не преобразовываются (или преобразование сильно ограничено) в программные коды БЗ.

При наличии большого количества CASE-средств, обеспечивающих построение диаграмм (в частности, UML: IBM Rational Rose, StarUML, Enterprise Architect и др.) и синтез каркасных программных кодов на языках программирования общего назначения, не выявлено ни одного средства, генерирующего программные коды для БЗ ЭС. В тоже время в наиболее распространенных системах программирования продукционных БЗ (например, Exsys Corvid, ClipsWin и др.) отсутствует возможность интеграции с CASE-средствами в части импорта концептуальных моделей. Что, в свою очередь, обуславливает постановку задачи создания, специализированного алгоритмического и программного обеспечения, обеспечивающего анализ концептуальных моделей и синтез программных кодов БЗ продукционного типа.

Таким образом, требуется разработать программную систему, обеспечивающую:

- поддержку процесса проектирования БЗ продукционного типа;
- интеграцию с CASE-средствами в части импорта концептуальных моделей в форме диаграмм классов UML;
- синтез и проверку программных кодов на CLIPS (Language Integrated Production System) [2];

– поддержку RVML (Rule Visual Modelling Language) [3] – графической нотации для моделирования логических правил.

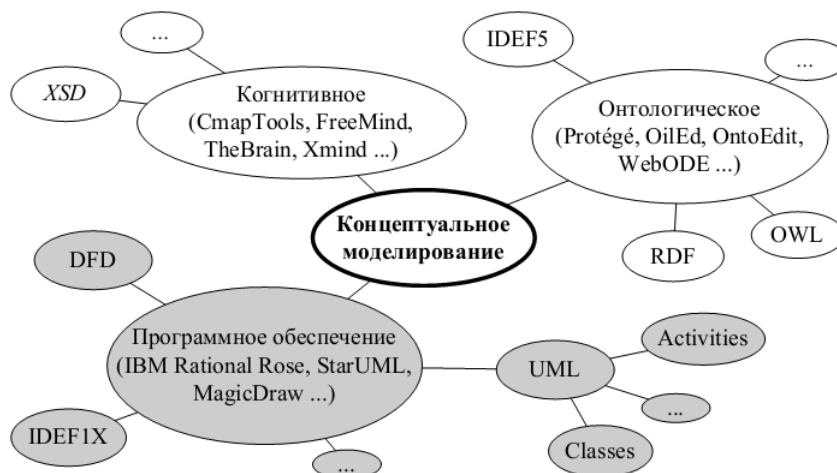


Рисунок 1 – Наиболее популярные стандарты и нотации создания концептуальных моделей

В качестве CASE-средства для интеграции предлагается использовать IBM Rational Rose [4].

Применение подобной программной системы позволит исключить ошибки, возникающие при ручном программировании, и обеспечит возможность быстрого прототипирования БЗ. Подробному описанию алгоритмического обеспечения, особенностям преобразования концептуальных моделей и краткому описанию RVML посвящены статьи [5, 6], поэтому в данной работе уделим основное внимание описанию программного средства и оценке эффективности его применения путем апробации на учебных (тестовых) примерах.

Personal Knowledge Base Designer

В результате решения поставленной задачи была разработана программная система – Personal Knowledge Base Designer, которая представляет собой специализированный редактор для разработки и тестирования производственных БЗ ЭС, в частности, для CLIPS [7].

Основными функциями редактора являются:

- создание элементов производственных БЗ (шаблонов фактов и правил, а также фактов и правил) непрограммирующим пользователем, благодаря использованию либо набора подпрограмм-мастеров, либо предварительно подготовленных шаблонов фактов и правил;
- поддержка авторской графической нотации – RVML [3] для визуального представления продукций. Что позволяет более наглядно отобразить причинно-следственные отношения и учесть некоторые особенности CLIPS (например, важность правил и коэффициенты уверенности);
- интеграция с CASE-средствами IBM Rational Rose [4] и StarUML, в части импорта концептуальных моделей (диаграмм классов UML), которые могут быть использованы для генерации элементов БЗ;
- интеграция с CLIPS [2], в части синтеза отжуждаемого кода БЗ и его тестирования, благодаря включению в состав модулей программной системы машины вывода CLIPS;
- формирование специализированных отчетов.

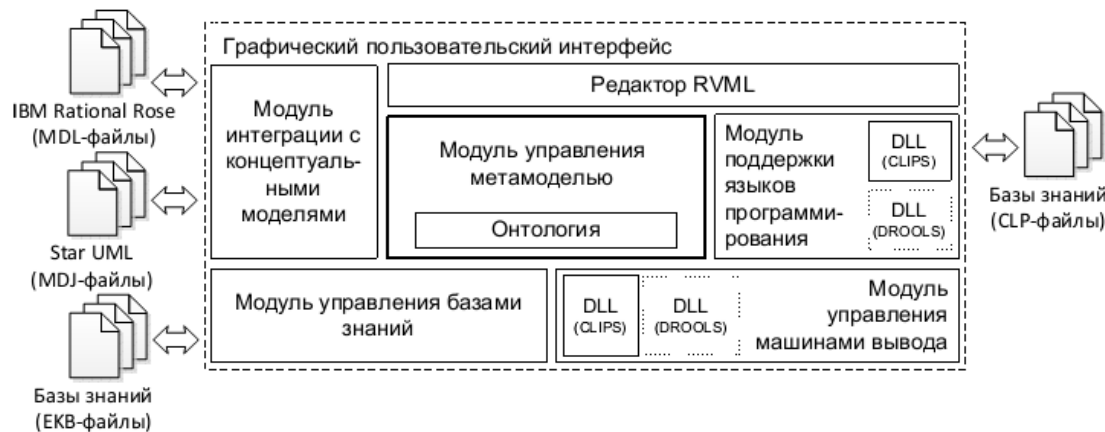


Рисунок 2 – Архитектура Personal Knowledge Base Designer

С целью реализации требований и функций, разработана архитектура [7], включающая следующие основные модули (Рис. 2):

- управления базами знаний – обеспечивает загрузку и сохранение БЗ в формате ЕКВ (XML-подобный формат программной системы для хранения знаний);

- управления метамоделью – обеспечивает внутреннее представление продукционной модели знаний, которое не зависит от определенного языка программирования БЗ, а также манипулирование (создание, редактирование, удаление) элементами этой модели;

- управления модулями поддержки языков программирования – обеспечивает подключение и отключение модулей ЯПЗ, а также доступ к их функциям;

- интеграции с концептуальными моделями – обеспечивает загрузку элементов из концептуальных моделей (диаграмм классов UML), построенных в CASE-средствах IBM Rational Rose и StarUML;

- управления машинами вывода – обеспечивает использование машин вывода (в виде динамических библиотек) для тестирования БЗ, включая объяснение полученных результатов;

- редактор RVML – обеспечивает представление элементов БЗ в виде графических примитивов, расширяющих UML [8],

- графический пользовательский интерфейс – обеспечивает доступ к перечисленным функциям.

Разработанный редактор обладает простым и интуитивно понятным графическим интерфейсом, который представлен основным рабочим пространством и набором программ-мастеров, которые подставляют собой последовательность экранных форм, сегментирующих и упорядочивающих процессы ввода и редактирования элементов БЗ.

В частности, при создании шаблона факта пользователю последовательно предлагается задать: имя шаблона (используется для отображения в редакторе), описание и свойства (слоты). Подобные программы-мастера также применяются для создания и редактирования шаблонов правил.

На основе данных шаблонов, пользователь может вводить конкретные факты и правила.

Быстрое прототипирование баз знаний

Редактор позволяет быстро создавать и тестировать продукционные БЗ, в том числе, на основе анализа и преобразование концептуальных моделей, в частности, диаграмм классов UML.

Рассмотрим простейший пример (использовался в учебном процессе) разработки ЭС для поддержки принятия решений при решении задач прогнозирования лесных пожаров. Основные понятия и отношения предметной области могут быть описаны в виде концептуальной модели (Рис. 3), анализ которой при импортировании в редактор позволяет сформировать описание шаблонов фактов и правил, используемых в дальнейшем для создания конкретных фактов и правил. При этом каждый элемент БЗ имеет графическое представление в виде RVML-схемы (Рис.4) на основе которого синтезируются программный код CLIPS.

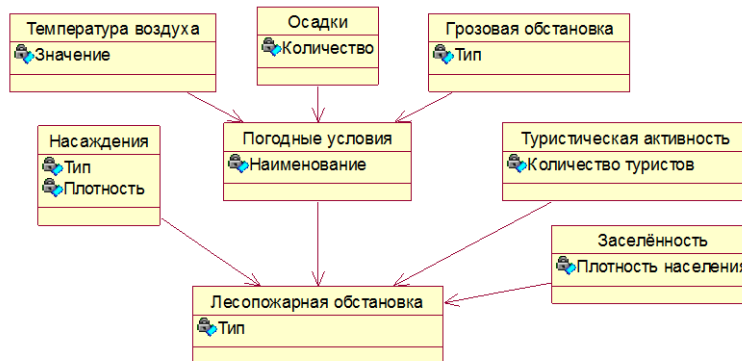


Рисунок 3 – Пример диаграммы классов UML

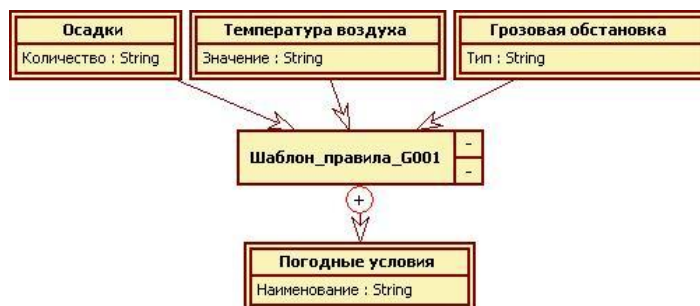


Рисунок 4 – Пример RVML-схемы для шаблона правила

Для проведения тестирования разработанной БЗ необходимо ввести начальные факты и активировать машину вывода (Рис. 5). В результате подобной проверки возможно ознакомиться с активированными правилами и изменениями в рабочей памяти, например, добавленными фактами.

Проверенная экспертом БЗ может быть сохранена в формате CLIPS и использована в сторонних приложениях.

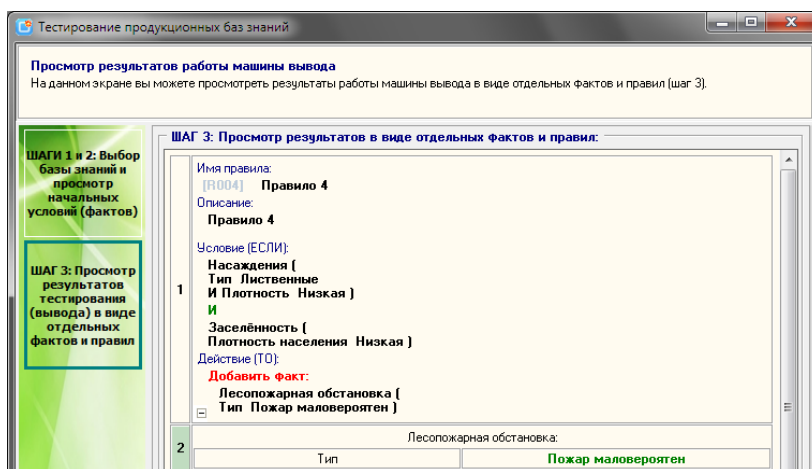


Рисунок 6 – Результаты тестирования базы знаний

Апробация

Разработанной алгоритмическое и программное обеспечение проходило апробацию на базе Иркутского национального исследовательского технического университета (ИрННТУ). В апробации приняло участие 60 студентов Института кибернетики им. Е.И. Попова изучающих курсы «CASE-средства», «Инструментальные средства информационных технологий» и «Технологии программирования», группы АСУз-10, АСУбз-11, АСУб-12, ЭВМбзс-12. Студенты обладали знаниями в области проектирования программного обеспечения, UML и искусственного интеллекта.

Основная задача апробации состояла в том, чтобы на примере решения учебных задач оценить трудоемкость разработки БЗ в формате CLIPS с использованием разработанного редактора и без него (путем применения других средств).

Таким образом, студентам было предложено разработать статические ЭС для решения задач диагностирования или прогнозирования в определенной предметной области (в зависимости от варианта задания, Табл. 1) тремя различными способами:

C1 – путем построения диаграмм классов UML в IBM Rational Rose, с последующим их импортом в Personal Knowledge Base Designer (PKBD) и доработкой полученных структур;

C2 – путем построения диаграмм классов UML в IBM Rational Rose, с последующим их использованием в других средствах разработки БЗ CLIPS;

C3 – путем использования других средств разработки БЗ CLIPS без построения каких либо диаграмм. В качестве «другого средства» для разработки БЗ CLIPS выбрана среда разработки – ClipsWin [9].

При этом для обеспечения возможности неоднократного повторения процесса решения задач и их временной компактности выполнения были введены ограничения на варианты заданий:

число предметных сущностей: 5-10;

число свойств предметных сущностей: до 3;

число связей между сущностями: 5-10;

число причинно-следственных связей: 3-4;

число экземпляров причинно-следственных связей (возможных правил): 10-15.

Следует отметить, что первые 4 пункта представляют собой ограничения, налагаемые на этап концептуального моделирования (ограничения на элементы диаграммы классов UML). Последний пункт относится к этапу программирования (реализации) кода БЗ в среде разработки (ограничение на количество возможных правил в БЗ).

Для оценки трудоемкости использовался временной критерий (затраты времени на выполнение отдельных этапов разработки ЭС). Оценка осуществлялась на следующих этапах [1, 10]:

Концептуализация:

формулировка базовых концепции и отношений между ними, включая:

- характеристику различных видов, используемых данных,
- анализ информационных потоков и лежащих в их основе структур в предметной области в терминах,

- причинно-следственных связей,
 - отношений частное/целое,
 - постоянное/временное и т.п.
- построение концептуальной модели.

Таблица 1. Описание решаемых задач.

Вариант	Задача	Предметная область	Предметные сущности (ед.)	Связи (ед.)	Причинно-следственные связи (ед.)	Правила (ед.)
1	Диагностирование	Автомобиль	6	5	3	10
2		Грибы	5	6	3	10
3		Компьютер	8	5	3	10
4		Грипп	5	8	4	11
5		Ангина	8	7	3	12
6		Электрочайник	9	5	3	10
7		Сотовый телефон	5	6	3	14
8		Тостер	8	7	4	14
9		Утюг	6	5	3	15
10	Прогнозирование	Погода	7	10	3	12
11		Курс валют	5	6	3	11
12		Цена бензина	5	6	3	12
13		Урожай	7	7	3	14
14		Общественное мнение	8	5	3	11
15		Настроение	7	6	3	18
16		Преступность	6	8	3	14
17		Заболееваемость	6	5	3	11
18		Лесные пожары	8	7	3	12
19		Наводнения (паводки)	7	8	3	10
20		Рождаемость	5	7	3	12

Таблица 2. Результаты оценки временных затрат.

Вариант	Моделирование, мин.	Работа в РКВД, мин.	C1, мин.	C2, мин.	C3, мин.	C3: Ошибки програм., шт.	Относительная разница, % C1 и C3	Относительная разница, % C1 и C2
1	10,89	7,2	18,09	41,29	30,4	2	40,49	56,19
2	8,36	7,1	15,46	32,86	24,5	0	36,89	52,95
3	8,58	8,3	16,88	36,46	27,88	1	39,45	53,70
4	9,36	5,83	15,19	26,82	17,46	0	13,00	43,36
5	11,25	5,52	16,77	64,41	53,16	3	68,45	73,96
6	10,78	4,6	15,38	43,8	33,02	1	53,42	64,89
7	6,6	15,82	22,42	68	61,4	5	63,48	67,03
8	10,95	7,56	18,51	57,23	46,28	3	60,00	67,66
9	7,37	7,2	14,57	54,71	47,34	3	69,22	73,37
10	12,58	6,6	19,18	42,7	30,12	0	36,32	55,08
11	8,69	5,5	14,19	38,01	29,32	0	51,60	62,67
12	8,36	6	14,36	45,22	36,86	2	61,04	68,24
13	10,64	7,42	18,06	44,31	33,67	2	46,36	59,24
14	10,66	10,23	20,89	50,57	39,91	1	47,66	58,69
15	10,01	7,56	17,57	55,5	45,49	4	61,38	68,34
16	10,92	8,96	19,88	49,42	38,5	1	48,36	59,77
17	8,14	8,36	16,5	45,59	37,45	1	55,94	63,81
18	11,55	18	29,55	47,43	35,88	3	17,64	37,70
19	11,85	5,2	17,05	43,28	31,43	1	45,75	60,61
20	9,12	7,44	16,56	40,95	31,83	2	47,97	59,56
Итоговое среднее значение относительной разницы:							48,2	60,3

Формализация:

- перевод ключевых понятий и отношений на некоторый формальный ЯПЗ;
- оценка полноты и степени достоверности (неопределенности) информации и других ограничений, накладываемых на логическую интерпретацию данных, таких как зависимость от времени, надежность и полнота различных источников информации.

Реализация:

преобразование формализованных знаний в работающую программу (код БЗ).

Основным результатом этапов концептуализации и формализации являлась концептуальная модель предметной области, представленная в форме диаграмм классов UML. Основным результатом этапа реализации – синтаксически корректный программный код БЗ, проверенный на адекватность и непротиворечивость.

Результаты оценки временных затрат представлены в таблице 2 и на рисунке 7. При этом выделены минимальные и максимальные процентные значения относительной разницы между С1 и С3, С1 и С2.

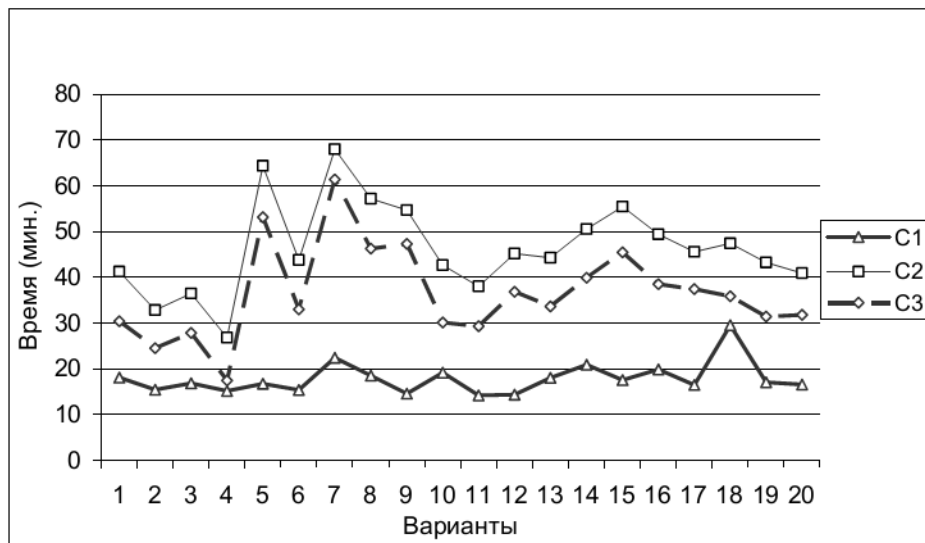


Рисунок 7 – Результаты оценки временных затрат

Отметим ряд особенностей способов:

C2: данный способ показал самые большие временные показатели, обусловленные тем, что построенные модели вручную переносились в среду разработки ClipsWin, так как у данного средства отсутствует поддержка возможности автоматической кодогенерации БЗ на основе созданных концептуальных моделей. При этом следует отметить, что другие программные средства, позволяющие синтезировать код БЗ в формате CLIPS (например, [11]), не удалось применить для этой задачи;

C3: Функциональные ограничения ClipsWin в части редактирования водимого программного кода обусловили применение дополнительного текстового редактора (Programmer's Notepad) при выполнении этапа кодирования. В частности, сначала осуществлялось описание кода БЗ во внешнем текстовом редакторе (используя возможности копирования и вставки отдельных блоков кода), а затем полученный код импортировался в ClipsWin, где осуществлялась проверка синтаксиса. На практике данная схема позволила снизить в 1,5 раза время на создание БЗ;

Анализ эффективности предлагаемого метода по временному критерию показал, что эффективность разработки баз знаний методом С1 может быть повышена, в среднем на 60.3% по сравнению с С2 и на 48.2% по сравнению с С3 за счет автоматической кодогенерации на основе визуальных моделей, что в свою очередь позволяет:

эффективно использовать результаты этапов концептуализации и формализации в форме диаграмм классов UML, рассматривая последние не как статические графические артефакты, а как основу для формирования программного кода в соответствии с идеологией модельно-управляемого подхода (Model Driven Architecture) [12];

снизить риск ошибок проектирования, за счет возможности быстрого прототипирования БЗ и получения их программного кода;

исключить ошибки программирования, за счет автоматического отображения элементов концептуальной модели в языковые конструкции CLIPS.

Заключение

Концептуальные модели остаются важным источником знаний при разработке БЗ ЭС, который в настоящее время достаточно ограниченно используется. Данные ограничения обусловлены недостатками существующего алгоритмического и программного обеспечения, что, в свою очередь, обуславливает необходимость создания специализированных программных систем в форме редакторов БЗ, способных интегрироваться с различными CASE-средствами в части импорта концептуальных моделей.

Примером подобного редактора является Personal Knowledge Base Designer [13], особенностями которого являются:

- ориентация на непрограммирующего специалиста (свойство реализовано с помощью набора подпрограмм-мастеров, обеспечивающих описание знаний в виде продукций).

- поддержка авторской графической нотации – Rule Visual Modeling Language (RVML) [3];

- интеграция с CASE-средствами IBM Rational Rose [4] и StarUML, в части импорта и экспорта концептуальных моделей (диаграмм классов UML).

Программная система используется в учебном процессе в Иркутском национальном исследовательском техническом университете (ИрННТУ) при выполнении лабораторных работ по курсам «CASE-средства», «Инструментальные средства информационных систем» и «Технологии программирования».

Апробация редактора показала значительное снижение времени разработки прототипов БЗ производственных ЭС. Кроме того, использование концептуальных моделей и Personal Knowledge Base Designer позволяет:

- уменьшить риск ошибок проектирования (быстрое прототипирование позволяет проверять модели);
- восстанавливать элементы БЗ на основе анализа концептуальных моделей;
- исключить ошибки программирования, благодаря автоматической генерации кодов.

Необходимо отметить, что концептуальные модели, используемые в качестве основы для формирования БЗ производственных ЭС, должны содержать описание причинно-следственных отношений. В противном случае результаты анализа моделей (осуществляемого при импорте) будут лишены содержательного смысла. Кроме того, текущая версия редактора поддерживает ограниченный набор основных конструкций языка, достаточный для автоматического синтеза исполнимых и синтаксически корректных программ. В частности, не поддерживаются: глобальные переменные, мультислоты, функции и сложные условия.

В дальнейшем планируется расширить поддержку форматов концептуальных моделей и языков программирования БЗ.

Работа выполнена при частичной финансовой поддержке РФФИ (проекты № 15-07-03088, 15-37-20655).

Литература

1. Гаврилова Т. А., Кудрявцев Д. В., Муромцев Д. И. Инженерия знаний. Модели и методы. – СПб.: Лань, 2016. – 324 с.
2. Частиков А.П., Гаврилова Т.А., Белов Д.Л. Разработка экспертных систем среда CLIPS. СПб: БХВ-Петербург, 2003. – 393 с.
3. Rule Visual Modeling Language (RVML). Режим доступа: <http://www.knowledge-core.ru/index.php?p=rvml> (дата обращения 27.10.2016).
4. IBM Rational Rose Enterprise. Режим доступа: <http://www-03.ibm.com/software/products/ru/enterprise> (дата обращения 27.10.2016).
5. Дородных Н.О., Юрин А.Ю. Использование диаграмм классов UML для формирования производственных баз знаний // Программная инженерия. – 2015. – №4. – С.3-9.
6. Грищенко М.А., Дородных Н.О., Николайчук О.А., Юрин А.Ю. Применение модельно-управляемого подхода для создания производственных экспертных систем и баз знаний // Искусственный интеллект и принятие решений. – 2016. – №2. – С. 16-29.
7. Юрин А.Ю., Грищенко М.А. Редактор баз знаний в формате CLIPS // Программные продукты и системы. – 2012. – №4. – С. 83-87.
8. Юрин А.Ю. Нотация для проектирования баз знаний производственных экспертных систем // Объектные системы – 2016: материалы XII Международной научно-практической конференции (Ростов-на-Дону, 10-12 мая 2016 г.) / Под общ. ред. П.П. Олейника. – Ростов-на-Дону: ШИ (ф) ЮРГПУ (НПИ) им. М.И. Платова, 2016. – С. 48-54.
9. ClipsWin: CLIPS Rule Based Programming Language. Режим доступа: <https://sourceforge.net/p/clipsrules/news/2008/01/clipswin-6241/> (дата обращения 27.10.2016).
10. Джексон П. Введение в экспертные системы. Пер. с англ. М: Вильямс, 2001. – 624 с.
11. Meditskos G., Bassiliades N. CLIPS-OWL: A framework for providing object-oriented extensional ontology queries in a production rule engine // Data & Knowledge Engineering, Vol. 70, No. 7, 2011. – p. 661-681.
12. MDA Guide rev. 2.0 // OMG Document ormsc/2014-06-01. Режим доступа:

<http://www.omg.org/mda/specs.htm> (дата обращения 27.10.2016).

13. Personal Knowledge Base Designer (PKBD). Режим доступа: <http://www.knowledge-core.ru/index.php?p=pkbd> (дата обращения 27.10.2016).

Дородных Н.О., Юрин А.Ю. Прототипирование продукционных баз знаний на основе диаграмм классов. В работе описана программная система для прототипирования продукционных баз знаний – Personal Knowledge Base Designer. Система ориентирована на непрограммирующего специалиста и позволяет автоматизировать процесс создания баз знаний благодаря: интеграции с CASE-средствами IBM Rational Rose и StarUML, в части импорта UML-моделей, генерации CLIPS-кода и тестирования разработанных баз знаний путем запуска машины вывода. Приведены результаты апробации программной системы при решении учебных (тестовых) задач.

Ключевые слова: программная система, базы знаний, правила, концептуальная модель, генерация кода, CLIPS, UML, диаграммы классов.

Dorodnykh N.O., Yurin A.Yu. Prototyping the rule knowledge bases on the basis of class diagrams The paper describes the software for designing and prototyping of rule knowledge bases – Personal Knowledge Base Designer. The software is focused on the non-programming specialists and automates the process of knowledge bases creation by means of: integraton with IBM Rational Rose (in terms of UML-models import and export), generation of CLIPS programming codes and also testing designed knowledge bases. The results of approbation of software are presented.

Keywords: software, knowledge base, rules, conceptual model, code generation, CLIPS, UML, class diagrams

Моделирование принятия субъективно рациональных решений

Г.П. Виноградов, Н.Г. Виноградова
Тверской государственной технической университет
wgp272ng@mail.ru

Виноградов Г.П., Виноградова Н.Г. Моделирование принятия субъективно рациональных решений. Актуальность работы: эффективность рефлексивного управления поведением людей в организационно-технологических системах особенно в условиях неопределенности и риска зависит от представлений у управляющего органа о том, как люди выявляют и воспринимают объективные признаки складывающихся ситуаций, строят оценки, вырабатывают и согласуют формы поведения. Это предполагает разработку моделей принятия решений субъектом, у которого есть внутренние образы себя и воздействующей на него стороны, и которые учитывают субъективное понимание ситуации выбора. Наличие таких моделей позволяет делать оценки эффективности информационного воздействия до того, как оно будет произведено.

Цель работы: разработка концептуальной схемы принятия решений, учитывающей процессы восприятия, осознаний и понимания свойств ситуации выбора для получения возможности: объяснения принятия решений субъектом в конкретных ситуациях; предсказания принимающим решение возможных реакций другого субъекта в различных ситуациях; решения задачи создания у управляемой стороны нужного образа ситуации выбора.

Результаты: представлена концептуальная схема моделирования принятия субъективно рациональных решений, предназначенная для понимания специфики управления эволюцией инновационных систем, в которых особую значимость приобретает гармоничность взаимодействия «субъект-проект-среда». В этом контексте важна проблема сборки субъектов, идентифицирующих себя с системой, в которой происходит их деятельность, и регулирующих свою активность с учетом ее влияния на результаты этой системы. Для ее решения предложено в модели выбора рассматривать субъект интересов, объект интересов, средства реализации интересов и порождаемые ими отношения. Определено, что мотивация активности субъекта связана с интересами и со стремлением к их реализации. Для формального описания интересов субъекта введены два показателя: удельная ценность ситуации целеустремленного состояния по результату и удельная ценность ситуации целеустремленного состояния по эффективности. Предложено разделить процесс принятия решений на этапы: 1) формирование представлений о свойствах ситуации выбора; 2) собственно принятие решения о способе действия; 3) построение алгоритма (плана) реализации решения; 4) реализация алгоритма; 5) оценка результатов. С формальной точки зрения рассмотрены два первых этапа. Качество принятого решения предложено оценивать по критерию удельной ценности ситуации целеустремленного состояния по результату, качество формирования представлений о ситуации выбора – критерием потерь ценности. Показано, что при принятии решения о способе действия субъект для удовлетворения своих интересов использует три множества альтернатив: множество способов действия, множество вариантов представлений, множество стратегий идентификации. Выбор альтернативы из соответствующего множества производится по своему индивидуальному критерию качества. Предложено его связать с некоторой виртуальной оперирующей стороной, имеющей в пределах своей компетенции свободу выбора. Интересы субъекта являются доминирующими для выделенных сторон, и он выступает для них в виде центра управления. Показано, что в этом случае принятие решений следует рассматривать как корпоративную игру с противоположными интересами. Поскольку интересы всех игроков взаимозависимы, то ее решение возможно на множестве согласованных компромиссных вариантов. Сформулированы требования к структуре представлений и наборам носителей информации для построения требуемых критериев. Это позволяет гарантировать, что для каждой выделенной стороны задача изолированного выбора ее «наилучшей» индивидуальной альтернативы по соответствующему критерию будет разрешима.

Ключевые слова: принятие решений, моделирование выбора, теория игр, согласование представлений, активные системы, идентификация.

Введение

Развитие идеи субъективно рационального выбора открыло возможность: 1) объяснить принятие решений субъектом в конкретных ситуациях; 2) предсказания принимающим решение возможных реакций другого субъекта в различных ситуациях; 3) решать задачу активного прогноза, когда управляющая сторона создает у управляемой стороны нужный образ будущего [8]. Субъективно рациональный выбор предполагает, что его мотивация определяется как внешними, так и внутренними факторами. Внутренние факторы отражают интересы субъекта, индуцируемые его потребностями и этической системой, которой он придерживается. Оценки удовлетворенности текущей ситуацией целеустремленного состояния субъектом, как показано в [1], могут приводить к изменению структуры интересов субъекта, и он ее может выбирать. Поскольку предпочтения субъекта в процессе выбора отражают его интересы, то можно определить множество G альтернативных вариантов структуры предпочтений, которые будем называть структурными альтернативами [2].

Исходные предположения

1. Выбор субъектом осуществляется на основе представлений о ситуации целеустремленного состояния.

2. Компоненты представления отражают различные аспекты понимания субъектом ситуации целеустремленного состояния и образуют информационную структуру представлений. Множество возможных вариантов представлений обозначим через X .

3. Для множества состояний окружения Ω множество наблюдаемых состояний окружения удовлетворяют условию $\Omega \cap X \neq \emptyset$, то есть представления субъекта могут содержать как объективную составляющую, так и фантомную.

4. Структурные альтернативы субъект выбирает в зависимости от оценок удовлетворенности значениями свойств ситуации целеустремленного состояния.

5. Формирование представлений осуществляется на основе процедур восприятия, осознания и анализа согласно с когнитивными возможностями субъекта.

Активные интеллектуальные системы содержит в себе субъект интересов, объект интересов, средства реализации интересов и порождаемые ими отношения. В работах [1, 3] доказано, что формально интересы субъекта (в дальнейшем агента) могут быть представлены двумя показателями: удельная ценность ситуации целеустремленного состояния по результату и удельная ценность ситуации целеустремленного состояния по эффективности. Они позволяют оценить степень желательности для агента результатов и величину прикладываемых для этого усилий. С усилиями связано определение возможных способов действия, которые с формальной стороны следует рассматривать как управление. Оценку качества выбора следует связывать со стремлением к реализации интересов с некоторым «наилучшим результатом». Формально понятие «наилучший результат» следует характеризовать, например, отношением удельной ценности ситуации целеустремленного состояния по результату к удельной ценности ситуации целеустремленного состояния по эффективности [3]. Значения этого показателя являются значениями некоторой шкалы, характеризующей субъективное отношение агента к эволюции своего состояния в зависимости от выбранного управления, а также к эволюции системы в целом. Полярные значения этой шкалы: гибель – доминирование. Понятно, что оценки по этой шкале будут определять конкретное содержание интересов, их эмоциональное переживание, степень стремления к их реализации. На их значения будут оказывать влияние система ценностей и норм агента, а также этическая система. Если учитывать тот факт, что реализация интересов агента возможна только в рамках некоторой организационной системы, то можно считать, что интересы системы в целом и каждого агента не противоположными, но, в общем случае, не совпадающими.

Как установлено в [4] система внутренних ценностей может рассматриваться априорно заданной и инвариантной лишь до тех пор, пока не возникнет ситуация возможной гибели агента или получение таких оценок свойств ситуации целеустремленного состояния, которые он не может считать удовлетворительными. С другой стороны, систему ценностей следует рассматривать зависимой от принятой агентом этической системы, которая определяет для человека такие фундаментальные понятия, как «Добро» и «Зло» [6]. Этические системы, построенные на импликации этих понятий в терминах намерение (цель), действие, результат, определяются какими-то глубинными свойствами этноса, группы. Следовательно, понятие «с наилучшими результатами» зависит от этической системы через систему ценностей и норм, которая в свою очередь определяет структуру предпочтений на действиях агента. Отметим, что в нормативной теории принятия решений структура предпочтений агента считается априорно заданной [5].

В данной схеме формализации интересов агента можно определить его субъективные оценки желательности и возможности достижения различных значений удельной ценности ситуации целеустремленного состояния, и через них конкретные значения показателей качества «жизни», к которому стремится агент. Поскольку их значения формируются на основе субъективных оценок ценности и возможности реализации интересов, то интересы агента и соответствующие цели являются эндогенными, то

есть формируемыми внутри системы. Структура и конкретные значения целей определяются степенью выражения в них интересов агента через субъективные оценки свойств компонент ситуации целеустремленного состояния (подробно об этом см. в [3]).

Стремление выжить, не допустить снижение удельной ценности ситуации целеустремленного состояния и обеспечить ее рост, данное в инстинктах, определяют второе свойство любой системы, проявляющей поведение. Оно связано с необходимостью непрерывного изучения своего окружения, а также совершенствования методов и способов действия. Его цель для агента состоит в поддержке прогрессивной эволюции себя как целостной системы.

Поведение агента с эндогенным принципом формирования целей

Введем ряд утверждений:

1. Мотивацией поведения субъекта являются интересы и потребности. Характеризующие их показатели образуют иерархическую структуру, например, в смысле Маслоу.
2. Агент, как носитель интересов, выступает для объекта интересов в качестве управляющей системы.
3. Объект интересов агента может быть, как пассивным, так и активным, обладающим свойством суверенности, активности, целеустремленности и интеллектуальности.
4. Субъект интересов может выбирать структуру предпочтений на множестве состояний объекта интересов, то есть его структура предпочтений не фиксирована.
5. Цель управления системой не задана экзогенно, а формируется внутри системы на основе согласования интересов и понимания обстановки.
6. Мотивация деятельности субъектов системы определяется их заинтересованностью в прогрессивной эволюции системы в целом и отдельных ее частей.

Согласно введенной аксиоматике мотивация на управление (действие) определяется заинтересованностью субъекта эволюцией объекта управления. Следовательно, существует набор переменных, которые агент наблюдает и воспринимает, и которыми он характеризует состояние объекта и свое окружение. Их восприятие и осознание является необходимым внешним условием, побуждающим субъекта к выбору управляющих воздействий, направленных на реализацию его интересов в ситуации выбора.

Заинтересованность агента в эволюции объекта интересов предполагает его отношение к состоянию окружения, которое порождается его представлениями о соответствии наблюдаемого состояния успешности реализации его интересов, намерений, интенций. Эти представления определяются многоуровневой системой обработки информации мозгом человека, структура которой с точки зрения постнеклассического принципа рациональности рассмотрена в [8]. Само отношение выражается на качественном уровне посредством вербальных оценок и касается всех компонент состояния, в том числе и самого субъекта выбора. Таким образом, представление порождает как отношение к состоянию, так и некоторое отношение к возможности и эффективности с помощью управляющих воздействий реализовать интересы с «наилучшим результатом». Существование такого отношения позволяет субъекту ориентироваться при выборе управления (способа действия), а также определять содержательный смысл (кроме того и прагматический смысл) его стремления к реализации интересов с «наилучшим результатом».

Определение 1. Качественные характеристики, определяющие отношение субъекта к 1) состоянию окружения, 2) оценке направления его эволюции, 3) ценности, эффективности возможных результатов, 4) вероятности реализации интересов с точки зрения представлений о состоянии окружения, определяемое его внутренней системой интересов, ценностей и норм, а также его этической системой будем называть оценками ситуации целеустремленного состояния.

Поскольку представление о ситуации целеустремленного состояния строится на основе субъективных оценок состояния окружения и о себе самом, то можно утверждать, что выбор управляющих воздействий, направленных на реализацию интересов с наилучшим результатом» является **субъективно рациональным выбором**, осуществляемый на основе представлений о ситуации целеустремленного состояния. Состояние окружения в этом случае следует рассматривать как экзогенный фактор.

Ситуация целеустремленного состояния существует в сознании субъекта, отражает его индивидуальные особенности по моделированию состояния окружения, в частности представление о состоянии среды может быть фантомным. Поэтому следует ожидать отношения субъекта к своим представлениям или рефлексии своих представлений. Это факт обсуждался в работах [4, 7, 8]. В частности, предложено оценивать представления по критерию их полезности при реализации интересов. В любом случае следует постулировать факт использования человеком определенных процедур, повышающих его убежденность в полезности представлений. Следовательно, человек осуществляет выбор из множества возможных вариантов представлений, то есть он выбирает модель ситуации целеустремленного состояния в зависимости от степени убежденности в ее адекватности и полезности для перехода в желаемое состояние. Для этой цели он разрабатывает и реализует специальные процедуры, которые будем называть процедурами идентификации ситуации целеустремленного состояния. Цель их применения – осуществление выбора модели ситуации

целеустремленного состояния для получения оценок возможных вариантов результатов. Следовательно, ситуация целеустремленного состояния является эндогенным фактором в модели выбора субъекта и определяет его отношение к наблюдаемому состоянию, характеризуемое им с помощью набора качественных характеристик.

Эти рассуждения позволяют предложить следующую схему выбора управляющих воздействий:

1. Для каждого состояния окружения $\omega \in \Omega$ агент ставит в соответствие ситуацию целеустремленного состояния $x_\omega \in X$, $g \in G$, где множество G характеризует предпочтения на множестве X . Для ситуации целеустремленного состояния определяется ее удельная ценность $E\phi$, как оценка степени реализации интересов (ожиданий). Проводится оценка степени удовлетворенности ситуацией, а также оценка степени соответствия представлений о ситуации целеустремленного состояния состоянию Ω . В случае, если убежденность в адекватном отражении состояния Ω ниже некоторого порога, выполняется процедура идентификации ситуации.

2. Если удовлетворенность ситуацией целеустремленного состояния ниже заданного порога, то из множества G структурных альтернатив (оно соответствует позитивному опыту $G' \subseteq G$, когда с их помощью субъект наблюдал реализацию своих интересов) выбирается управляющее воздействие $c \in C$, позволяющее достичь желаемой ситуации целеустремленного состояния, ценность которой либо превышает пороговое значение, либо является оптимальной величиной при имеющихся возможностях. Определяется значение ценности желаемой ситуации целеустремленного состояния, строится дерево целей и способов действия, позволяющих ее достичь и конкретизирующее интересы субъекта. Переход к п.6.

3. В противном случае определяется возможность достичь приемлемого состояния с помощью других вариантов структурных альтернатив $g \in G$. Если это невозможно, то фиксируется возникновение проблемы, составляется проект ее преодоления путем соответствующих исследований и разработок. Их цель: а) расширение множества способов действия $C \uparrow$; б) расшивка ограничений и т.п. Для этой цели определяется направление эволюции в пространстве показателей и соответствующая возможностям величина шага.

4. Формируется план формирования нового намеченного способа действия $c \in C \uparrow$ и план расширения множества способов действия $C' \equiv C \uparrow$.

5. Определяется величина стимулирования для создания определенного уровня мотивации субъектов, задействованных в реализации интересов субъекта интересов.

6. Для ситуации целеустремленного состояния реализуются структурные управляющие воздействия в форме, определенной в п. 4-5.

Такую схему, вытекающую из аксиом, постулирующих мотивацию управления (целеустремленного поведения) будем называть схемой управления целенаправленной эволюции, определяемой стремлением субъекта к выживанию, сохранению достигнутого уровня или доминирования.

Согласно этой схеме субъект принимает управляющие решения, учитывая два типа условий: 1) экзогенных (объективных), порождаемых динамикой окружения и объекта интересов; 2) эндогенных (субъективных), порождаемых интересами субъекта.

Это возможно, если субъект обладает *стремлением к реализации интересов, к повышению вероятности их достижения за счет роста своего уровня знаний* закономерностях динамических процессов в окружении и динамики перехода объекта управления в различные состояния и обладает соответствующими *креативными способностями*. Варианты интересов могут быть представлены следующими терминами на оценках удельной ценности ситуации целеустремленного состояния: *выживание, сохранение достигнутого уровня жизни, возможность целенаправленного изменения среды, доминирование*.

Согласно аксиомам выбора, управляющая альтернатива выбирается в зависимости от представлений о ситуации, которые формируются с помощью процедуры идентификации сообразно с состоянием. Тогда функция полезности должна быть определена на множествах способов действия, оценок ситуации $x \in X$ и состояний $\omega \in \Omega$. Пусть субъект имеет возможность выбирать структуру предпочтений из заданного множества структурных альтернатив G для получения приемлемых оценок удовлетворенности ситуации целеустремленного состояния. Тогда его функция полезности будет зависеть от структурной альтернативы $g \in G$ как от параметра вида $E\phi^g : (C \times \Omega \times X) \rightarrow R^1$, представляющая априорные предпочтения на управляющих альтернативах $c \in C$ в соответствии с условием: $c' \succ c \Leftrightarrow E\phi^g(c', \omega, x) > E\phi^g(c, \omega, x)$. Подробно содержательный смысл и способы задания функции полезности для агентов с субъективно рациональной формой поведения рассмотрены в [3].

Как показано в [3] функция полезности $E\phi^g(c', \omega, x)$ для агента имеет смысл оценки удельной ценности ситуации целеустремленного состояния и представляет его априорные внутренние предпочтения на управляющих альтернативах в зависимости от состояний и ситуаций, а также его системы ценностей и норм, задаваемых этической системой. Так как выбор осуществляется в условиях неопределенности, то это предполагает в ведение в модель выбора предположений и правил устранения неопределенности, которые еще

называют гипотезами детерминизма. Различные варианты элиминации неопределенности рассмотрены в работах Д.А. Новикова [9]. Здесь же ограничимся замечанием, что правило управления естественно выбирать из условия достижения максимума ожидаемой удельной ценности ситуации целеустремленного состояния.

Поскольку понятие ситуации целеустремленного состояния существует только в сознании агента, то суть ее формирования состоит в выборе модели ситуации целеустремленного состояния в зависимости от состояния окружения. Очевидно, что если возможные модели рассматривать в роли альтернатив идентификации состояния окружения, то требуется некоторый критерий оценки их полезности и качества. Поскольку модель ситуации целеустремленного состояния является основой для выбора, то функция полезности может служить основанием для определения смысла и структуры требуемого критерия. Действительно, пусть для каждого состояния $\omega \in \Omega$ определена пара переменных $(x^*, c^*)_{\omega} \in X \times C$, на которой функция полезности достигает своего максимального значения. Пусть теперь из некоторых соображений в состоянии $\omega \in \Omega$ выбрана ситуация $x \in X$ в роли результата идентификации, и с учетом этого выбрано управляющее воздействие $c \in C$. В таком случае будут иметь место определенные «потери полезности» в сравнении с возможным наилучшим выбором.

Поскольку, в общем случае, окружение субъекта интересов является неопределенным и слабо формализуемым и наблюдаемая траектория состояний зависит от состояния окружения и от правила выбора управляющих, действий, то качество представлений будет зависеть от используемых процедур идентификации состояния, которые можно рассматривать как средство детерминации представлений. Однако получить оценки потерь в этом случае возможно только по результатам реализации выбора. Поэтому человек на значениях возможных значений функции полезности в ситуации неопределенности строит два типа оценок: первый связан с оценками полезностью представлений для получения желаемых состояний, а второй – с оценками возможности их достижения в виде оценок степени сомнения в правильности (адекватности) представлений.

Определим степень сомнения в правильности представлений о ситуации выбора в состоянии окружения $\omega \in \Omega$ относительно возможности достижения желаемого состояния $o \in O$ располагаемыми способами действия $c \in C$ как степень стремления у агента получить дополнительную информацию о количестве и степени влияния факторов окружения для повышения своей информированности $X \uparrow$.

Поскольку степень стремления к получению дополнительной информации изменяется от нуля до некоторого максимума, то эти оценки можно рассматривать как значения шкалы лингвистической переменной степень сомнения.

Оценки полезности представлений связаны с ценностью появления желаемых состояний $o \in O$ в окружении $\omega \in \Omega$ при выборе способа действия $c \in C$ на основе представления $x \in X$, которые также являются значениями лингвистической переменной с тем же названием.

Очевидно, что интегральная оценка потерь полезности может быть получена по известным правилам теории нечетких множеств. Соответствующую оценку, как и в теории статистических решений, будем называть «риском»

Таким образом, в схеме целеустремленного управления для описания качества управления естественно использовать критерий, имеющий смысл ожидаемой удельной ценности ситуации целеустремленного состояния, а для описания качества идентификации состояния окружения – критерий, имеющий смысл лингвистической переменной риск. Ясно, что выбираемые таким образом правила будут определенным способом взаимозависимы. В этих условиях проблема выбора имеет игровое содержание, а ее «наилучшее» решение состоит в построении некоторого устойчивого компромисса между достижением максимальной ожидаемой полезности и минимального риска. Такой компромисс принято называть «равновесием» [9]. Определение подобного равновесия естественно рассматривать в качестве внутренней цели управления, обеспечивающей требуемое значение по шкале удовлетворенности.

С учетом выполненных рассуждений окончательно концепцию управления в условиях принятой аксиоматики определим следующими основными положениями.

1. Наблюдение и обнаружение состояния является необходимым, но не достаточным условием осуществимости выбора управления.
2. Достаточное условие осуществимости выбора управления определяется заданием отношения субъекта к состоянию, определяемого некоторой качественной характеристикой, называемой ситуацией целеустремленного состояния.
3. Вследствие того, что причинно-следственные связи, определяющие поведение окружения, недоступны непосредственному наблюдению, возникает необходимость выполнения процедур идентификации состояния окружения, их цель – выбор модели ситуации наблюдаемого состояния.
4. Выбор правила управления выполняется по критерию ожидаемой полезности.
5. Выбор правила идентификации выполняется по критерию риска.
6. Проблема выбора правил управления и идентификации имеет игровое содержание, «наилучшее» решение которой состоит в построении устойчивого компромисса, называемого «равновесием».

7. Построение и использование равновесных правил управления и идентификации является внутренней целью управления.

Сформулированные положения определяют концепцию «целестремленного управления активными системами с эндогенно формируемыми целями».

Согласно введенной аксиоматике объект интересов может быть активным, динамическим при этом природа его динамики не связана с механическим движением. Если его эволюция описывается правилом $f : X \times C \rightarrow X$ и f – нечеткая функция вследствие использования лингвистических оценок, то соответствующая система является нечеткой системой, состояние которой в момент $t+1$ есть условное по x_t и c_t нечеткое множество, характеризуемое функцией принадлежности вида $\mu(x_{t+1} | x_t, c_t)$. В условиях естественно полагать, что эволюция объекта описывается нечетким марковским процессом.

Без ограничения общности можно полагать, что множество состояний окружения Ω некоторым способом упорядочено и их значения случайны, то естественно полагать, что задано априорное распределение вероятностей $\beta(\Omega)$ начальных состояний.

Согласно концепции целестремленного управления, выбор управляющих альтернатив осуществляется в ситуации целестремленного состояния, которая является качественной характеристикой, определяющей отношение субъекта к состоянию. Возможное множество ситуаций X может быть лишь конечным и при этом должно выполняться условие $|X| \leq |\Omega|$. Следовательно, к определенным выше двум аспектам интересов концепция целестремленного управления вводит третий аспект, который связан с необходимостью идентификации ситуации в зависимости от наблюдаемого состояния.

Естественно полагать, что для каждого аспекта интересов существуют альтернативные варианты их реализации, и заданы соответствующие множества допустимых альтернатив. Тогда, следуя концепции целестремленного управления, положим, что для аспекта интересов, связанного с управлением эволюцией состояний, задано множество C способов действия – управляющих альтернатив. Поскольку управляющие альтернативы выбираются в зависимости от ситуаций $x \in X$, то естественно полагать, что существуют ограничения на допустимость управляющих альтернатив в зависимости от ситуации $x \in X$. Такие ограничения естественно задавать включениями вида $C_x \subset C, x \in X$, либо $C_x \subseteq C, x \in X$.

Как показано выше, чтобы ситуация целестремленного состояния была бы доступна сознанию агента, необходимо выполнение ее идентификации, содержание которой сводится к выбору ситуации из заданного множества в зависимости от состояния, то есть $x \in X$. Очевидно, альтернативы идентификации зависят от состояния $s \in S$, то есть $X_s \subset X, s \in S$.

Субъект заинтересован в реализации своих интересов для этого он выбирает структуру своих предпочтений $g \in G$ из заданного множества G структурных альтернатив.

Структура предпочтений отражает только интересы агента. Поэтому их выбор не может зависеть от состояний и ситуаций. То есть структурные альтернативы должны рассматриваться в качестве общего для состояний и ситуаций параметра, определяющего закономерность динамики состояний, как объекта интересов, так и окружения.

Естественно полагать, что в составе активной системы субъект может наблюдать состояния и принимать управляющие решения лишь в дискретные моменты времени. С учетом этого процесс управления будем понимать, как процесс последовательного принятия решений с дискретным временем.

Если субъект не меняет на интервале $n > 1$ стремления достичь определенного состояния и не меняет структуру своих предпочтений, то такая структурная альтернатива является стратегической, в противном случае – тактической.

Согласно введенному выше предположению эволюция объекта интересов описывается нечетким марковским процессом в пространстве состояний S . Применение управляющих воздействий в дискретные моменты времени порождает управляемый марковский процесс с дискретным временем с тем же множеством состояний S . Динамика такого процесса определяется переходной функцией, задающей вероятности одношаговых переходов на множестве состояний в зависимости от выбора управляющих альтернатив $u \in U$. При этом она будет зависеть также и от структурной альтернативы $g \in G$, но как от параметра. Обозначим переходную функцию управляемого процесса символом $q^g(S | S \times C)$ [2].

Согласно концепции целестремленного управления должна быть задана функция полезности $E\varphi^g : (C \times S \times X) \rightarrow R^1$, представляющая априорные предпочтения на управляющих альтернативах $c \in C$ в соответствии с условием:

$$c' \succ c \Leftrightarrow E\varphi^g(c', s, x) > E\varphi^g(c, s, x).$$

Введенные предположения определяют условия принятия управляющих решений, где функция полезности зависит от структурной альтернативы $g \in G$ как от параметра, а управление выполняется в соответствии со схемой ситуационного управления, мотивированного интересами.

Информационная структура представлений

В соответствие с введенными положениями, используя результаты работы [2], определим соответствующие объектам интересов носители информации как набор следующих формальных объектов:

$$I = \{S, \beta_0(S), X, [X_s \subset X, s \in S], C, [C_x \subseteq C, x \in X], G, q^g(S | S \times C), \text{Eф}^g(C \times S \times X), g \in G\},$$

где S – множество состояний; $\beta(S)$ – априорное распределение на множестве состояний; X – множество ситуаций; $X_s \subset X$ – ограничения на допустимость ситуаций в роли альтернатив идентификации в зависимости от состояний $s \in S$; C – множество управляющих альтернатив; $C_x \subseteq C$ – ограничения на допустимость управляющих альтернатив в зависимости от ситуаций $x \in X$; G – множество структурных альтернатив; $q^g(S | S \times C)$ – переходная функция из $S \times C$ в S ; $\text{Eф}^g(C \times S \times X)$ – функция полезности, представляющая априорные предпочтения на альтернативах $c \in C$ в зависимости от состояний $s \in S$, ситуаций $x \in X$ и структурных альтернатив $g \in G$.

Особенность информационной структуры I заключается в том, что выбор управляющих воздействий зависит от представлений о ситуации выбора, формируемых на основе процедуры идентификации. В этих условиях закономерность динамики ситуаций априори не может быть задана. Поэтому агент использует субъективные оценки закономерности динамики состояний, определяемой переходной функцией $q^g(S | S \times C)$ из $S \times C$ в S .

Если субъект сомневается в своих представлениях об элементах информационной структуры, то он использует дополнительную информацию для формулирования некоторого правдоподобного приближения. Например, если явный вид переходной функции априори не известен, то всегда можно сформулировать некоторое множество гипотез Γ о ней.

Пусть, например, в базовой информационной структуре I переходная функция $q^g(S | S \times C)$ не задана, но задано множество гипотез Γ о ней. Тогда формально можно полагать, что переходная функция $q(g, \gamma)(S | S \times C)$ зависит от некоторого параметра γ принимающего значения из заданного множества Γ , но истинное значение такого параметра остается неизвестным. В этих условиях информационная структура будет определяться набором объектов вида

$$I\Gamma = \{S, \beta(S), X, [X_s \subset X, s \in S], Y, [Y_x \subseteq Y, x \in X], G, \Gamma, q(g, \gamma)(S | S \times X), \text{Eф}^g(Y \times (S \times X)), g \in G, \gamma \in \Gamma\}$$

Очевидно, что это приведет к расширению содержания проблемы, которая теперь вместе с выбором ранее введенных альтернатив потребует также выбора в определенном смысле «наилучшей» гипотезы о переходной функции.

Элементы модели субъективно рационального выбора

Введенные предположения фиксируют существование двух аспектов интересов субъекта, один из которых определяется заинтересованностью в эволюции объекта интересов, а другой – в выборе структуры интересов. Концепция целеустремленного управления порождает третий аспект интересов, связанный с необходимостью идентификации ситуации в зависимости от наблюдаемого состояния. В соответствии с этими тремя аспектами информационная структура I предполагает задание множества C управляющих альтернатив, множества G структурных альтернатив и множества X альтернатив идентификации. Предполагается задание также функции полезности $\text{Eф}^g(C \times S \times X)$ и переходной функции $q^g(S | S \times C)$ из $S \times C$ в S . Это позволяет задать критерий качества выбора управления, имеющего смысл ожидаемой полезности, и критерий качества выбора правила идентификации, имеющего смысл риска. Для выбора структурных альтернатив может быть введен критерий качества, например, отражающий степень удовлетворения интересов.

Таким образом, при принятии решения о способе действия субъект для удовлетворения своих интересов использует три множества альтернатив. Выбор альтернативы из соответствующего множества производится по своему индивидуальному критерию качества и его естественно связать с некоторой виртуальной оперирующей стороной, имеющей в пределах своей компетенции свободу выбора. Интересы субъекта являются доминирующими для выделенных сторон, и он выступает для них в виде центра управления. Поскольку интересы всех игроков взаимозависимы, то такая игра относится к классу игр с корпоративными интересами. Ее решение возможно на множестве согласованных компромиссных вариантов. Оно будет

устойчивым, если его невозможно улучшить без ухудшения хотя бы одного из критериев. Подобное моделирование выбора будет возможно, если для структуры представлений определить наборы носителей информации для построения требуемых критериев. Это позволяет гарантировать, что для каждой выделенной стороны задача изолированного выбора ее «наилучшей» индивидуальной альтернативы по соответствующему критерию будет разрешима. Тогда можно ожидать, что и задача совместного отыскания компромисса, удовлетворяющего требованиям корпоративной устойчивости, также будет разрешима.

Заключение

В статье представлена концептуальная схема моделирования принятия субъективно рациональных решений, предназначенная для понимания специфики управления эволюцией инновационных систем, в которых особую значимость приобретает гармоничность взаимодействия «субъект-проект-среда». В этом контексте важна проблема сборки субъектов, идентифицирующих себя с системой, в которой происходит их деятельность, и регулирующих свою активность с учетом ее влияния на результаты этой системы. Принято разделить процесс принятия решений на этапы: 1) формирования представлений о свойствах ситуации выбора; 2) собственно принятие решения о способе действия; 3) построение алгоритма (плана) реализации решения; 4) реализация алгоритма; 5) оценка результатов. С формальной точки зрения рассмотрены два первых этапа. Качество принятого решения предложено оценивать по критерию удельной ценности ситуации целеустремленного состояния по результату, качество формирования представлений о ситуации выбора – критерием потерь ценности. Показано, что в этом случае принятие решений следует рассматривать как корпоративную игру с не противоположными интересами.

Литература

1. Виноградов, Г.П. Моделирование поведения агента с учетом субъективных представлений о ситуации выбора / Г.П. Виноградов, В.Н. Кузнецов // Искусственный интеллект и принятие решений. 2011. № 3. с. 58-72.
2. Баранов В.В. Динамические равновесия в задачах стохастического управления и принятия решений при неопределенностях // Изв. РАН. Теория и системы управления. 2002. №3. С. 77-93.
3. Виноградов, Г.П. Методы и алгоритмы принятия решений в автоматизированных системах управления производствами с непрерывной технологией на основе субъективных представлений: монография. / Г.П. Виноградов. Тверь: ТГТУ, 2013. 256 с.
4. Виноградов, Г.П. Формирование представлений агента о предметной области в ситуации выбора/ Г.П. Виноградов, Г.П. Шматов, Д.А. Борзов // Программные продукты и системы. №2 (110), 2015. – С. 83–94.
5. Ларичев О.И., Мовшович Е.М. Качественные методы принятия решений. Вербальный анализ решений. – М.: Наука, 1996.
6. Lefebvre V. Algebra of Conscience. Dordrecht/Boston/London.: Kluwer Academic Publ. – 2001.
7. Виноградов Г.П., Кузнецов В.Н. Целеуказание на основе представлений агента о целеустремленной ситуации в слабоструктурированной среде. XII Всероссийское совещание по проблемам управления ВСПУ–2014. Москва, 16-19 июня 2014г.: Труды [Электронный ресурс]. М.: Институт проблем управления им. В.А. Трапезникова РАН, 2014. Электронные текстовые данные. ISBN 978-5-91450-151-5. Номер государственной регистрации 0321401153. – с. 7907-7918.
8. Виноградов Г.П., Кузнецов В.Н. Эволюция представлений о моделях принятия решений в контексте научной рациональности. Рефлексивные процессы и управление. Сборник материалов X Международного симпозиума 15-16 октября 2015 г. Москва / Отв. ред. В.Е. Лепский – М.: «Когито-Центр, 2015. – с. 27-30.
9. Новиков Д.А., Ивашенко А.А. Модели и методы организационного управления инновационным развитием фирмы. – М.: ЛЕНАНД, 2006. – 336 с.

***The relevance of the work:** due to the fact that the effectiveness of reflexive control the behavior of people in organizational and technological systems especially in the conditions of uncertainty and risk depends on the views of the governing body about how people identify and perceive objective evidence of factual situations, build assessments, produce and agree on forms of behavior. This includes the development of models of decision-making entity, which has internal images of yourself and acting on it side, and which take into account the subjective understanding of the situation of choice. The availability of such models allows to estimate the effectiveness of informational influence before it is produced.*

***Goal:** to develop a conceptual framework of decision-making that takes into account the processes of perception, awareness and understanding of the properties of choice situations to be able to: explain the decision-making entity in specific situations; prediction decision possible reactions of another subject in different situations; the creation of a managed side the desirable image of the situation of choice.*

Research methods: General methodology and methods of system analysis, analytical modeling, the work also uses methods theories: sets of active systems, decision making, artificial intelligence, fuzzy systems and fuzzy logic, game theory.

Results: the conceptual diagram of the simulation of the adoption of the subjectively rational decisions designed to understand the specificity of management evolution of innovation systems in which of particular importance is the harmonious interaction of the "subject-project environment. In this context, an important problem of the Assembly of constituent entities identifying themselves with a system where their activities and regulating their activity, given its impact on the results of this system. For its solution is proposed in the selection model to consider the subject of interest, object of interest, means of realization of interests and resulting relations. It is determined that motivation of activity of the subject connected with the interests and with the desire to implement them. For a formal description of the interests of the subject introduced two indicators: the specific value of a purposeful state situation, the result and the specific value of a purposeful state situation. It is proposed to divide the decision process into stages: 1) formation of ideas about the properties of choice situations; 2) the actual decision-making about mode of action; 3) build an algorithm (plan) implement the solution; 4) implementation; 5) evaluation of results. From a formal point of view, are considered the first two stages. The quality of the decision proposed to be assessed against the specific value of a purposeful state situation according to the result, the quality of formation of representations about a situation of choice – the criterion of loss of value. It is shown that when deciding on the mode of action of the subject to suit their interests uses three sets of alternatives: multiple ways of action, lots of view options a lot of strategies of identification. The choice of an alternative from the set is your personal criterion. It is proposed to associate with a virtual operating party that has within its competence the freedom of choice. The interests of the subject are dominant for the selected parties, and he speaks to them in the form of control center. It is shown that in this case decisions should be considered as a corporate game with not opposite interests. Endogenous aim of the subject of the decision in this case is this distribution of effort between stages, to achieve the maximum satisfaction from the desired state. Since the interests of all players are interdependent, then its solution is possible on many of the agreed compromise. It will be sustainable if it cannot be improved without worsening at least one of the criteria. Requirements for the structure of the representations and sets of media to build the required criteria. This ensures that each selected part, the task of the isolated choice of its best individual alternatives on the appropriate criteria to be solvable.

Key words: decision making, choice modeling, game theory, coordination of ideas, active systems, identification.

Разработка структуры системы автоматического управления роботом-манипулятором для малоинвазивной хирургии

Е. Ю. Косенко, Е.Д. Синявская, О.В. Косенко
Южный федеральный университет
eksosenko@sfedu.ru

Косенко Е. Ю., Синявская Е.Д., Косенко О.В. Разработка структуры системы автоматического управления роботом-манипулятором для малоинвазивной хирургии. В статье рассмотрены задачи, связанные с разработкой структуры системы автоматического управления роботохирургическими устройствами для малоинвазивной хирургии. С этой целью в статье рассматривается функционирование роботизированного держателя лапароскопа с позиции системного подхода, определяется тип конструкции, идентифицируется кинематическая модель и определяется метод оценивания эффективности его работы.

Ключевые слова: роботохирургия, лапароскоп, робот-манипулятор, система управления, кинематическая модель, целевая функция

Введение

В настоящее время, как правило, при проведении малоинвазивной хирургии, способной минимизировать послеоперационную травму для пациента визуализация оперируемой области обеспечивается за счет действий ассистента под руководством хирурга. Подобный подход помимо достоинств этого метода хирургического вмешательства имеет ряд недостатков, которые заключаются в снижении хирургом восприятия глубины манипуляций, (что осложняет зрительно-моторную координацию), увеличение тремора рук, ограничении диапазона движения и степеней свободы, повышенной утомляемости и тактильной ограниченности. Также существует ряд трудностей, связанных с осуществлением контроля над положением лапароскопической камеры, которой во время операции управляет ассистент хирурга, пытающийся совместить обзор оперируемой области с рекомендациями врача, что не всегда обеспечивает безопасное перемещение лапароскопа в требуемое положение.

Более прогрессивное использование автоматизированных робототехнических комплексов для контроля и управления лапароскопом, таких как Aesop, EndoAssist, Viky, SoloAssist или Naviot [1 – 4], преодолевают некоторые из указанных недостатков, позволяя хирургу контролировать камеру напрямую через интерфейс управления посредством педали или джойстика, или через более сложные средства: голосовое управление или движения головы хирурга [5 – 7]. Однако такие системы не исключают влияния человеческого фактора, что может приводить к ошибкам хирурга как оператора [8].

Разработка и внедрение методов и систем автоматического управления, позиционирования и слежения лапароскопической камеры за хирургическими инструментами в процессе выполнения операций позволит снять с ассистента или хирурга задачу управления лапароскопом, снизить влияние человеческого фактора, повысить безопасность и качество проводимых операций, уменьшить количество медицинского персонала и расширить функциональные возможности хирурга при проведении операции.

Целью настоящей статьи является разработка структуры и общих принципов функционирования системы автоматического управления роботом-манипулятором для перемещения лапароскопа, идентификация кинематической модели манипулятора и определение методов оценивания работы предлагаемой системы.

Описание системы автоматического управления роботом-манипулятором для удержания и перемещения лапароскопа во время операции

Большинство существующих роботохирургических систем и устройств (da Vinci, Soloassist, EndoAssist, Robolens, Freehand System [1 – 4]) обеспечивают хирурга только визуальной обратной связью и не обеспечивают систему в целом числовыми данными о положении и ориентации кончика хирургического инструмента. Такая обратная связь необходима для введения режима автоматического слежения за хирургическими инструментами.

Проведение лапароскопической операции при помощи роботизированных держателей можно

представить в виде системы, состоящей из совокупности взаимосвязанных блоков. На рисунке 1 показана структурная схема автоматической системы управления лапароскопом при проведении малоинвазивной операции [9]. На первоначальном этапе рассматривается позиционирование одного хирургического инструмента, поскольку положение второго (вспомогательного) инструмента будет определяться аналогично первому, тогда как изменение положения лапароскопа внутри оперируемой области выполняется относительно ведущего инструмента (как правило, хирургического инструмента, находящегося в правой руке хирурга) [3 – 7, 9, 10].

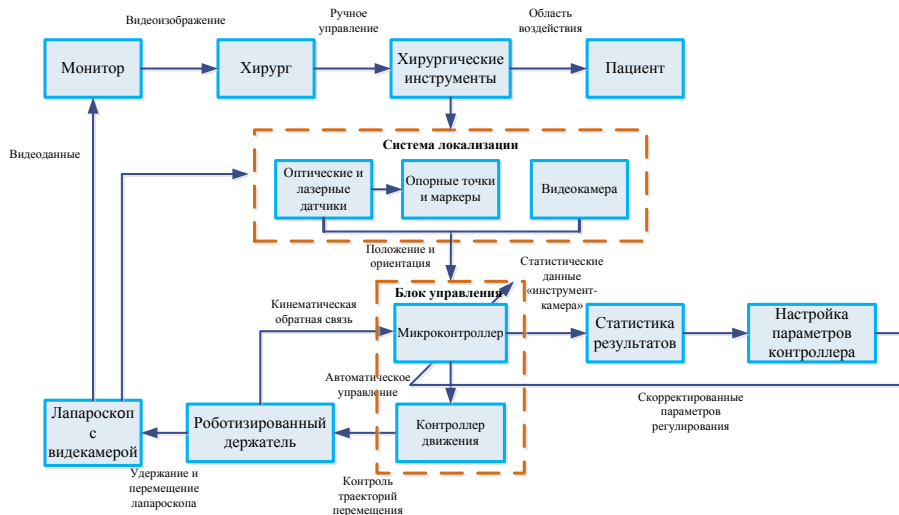


Рисунок 1 – Структурная схема системы автоматического управления роботом-манипулятором для удержания и перемещения лапароскопа во время операции

Хирург на основании видеоизображения оперируемой области выполняет операцию по средствам хирургических инструментов, введенных при помощи троакаров. С помощью оптической системы локализации и внешней видеокамеры определяются текущие координаты положения лапароскопа и лапароскопических инструментов. Системы локализации такого типа предполагают использование внешних опорных точек и маркеров, находящихся на видимых частях троакаров. По углу отражённых лучей от опорных точек определяется местоположение и угол наклона инструментов и лапароскопа. Полученные координаты поступают в блок управления, в котором микроконтроллер формирует управляющие команды для изменения положения и ориентации роботизированного держателя или фокусировки видеокамеры лапароскопа с целью обеспечения оптимальной визуализации оперируемой области. Блок управления затем посылает сгенерированные сигналы управления на контроллер движения. Контроллер движения подает команды на исполнительные механизмы робота, что приводит к изменению положения звеньев робота в соответствии с желаемыми траекториями перемещения звеньев и целевыми позициями для лапароскопа.

Все принятые решения, соответствующие определенным входным условиям, то есть положениям хирургических инструментов, и выходным заключениям, характеризующим перемещения лапароскопа, сохраняются в блоке статистических данных. Данные по всем результатам управления используются для сравнительного анализа работы и последующей адаптации параметров блока управления.

Роботизированный держатель перемещает лапароскоп в соответствии с областью хирургического вмешательства и текущего положения инструментов. Для определения границ или безопасной области функционирования, необходимо использовать кинематическую связь между звеньями робота в процессе его перемещения. Далее по каналам связи от видеокамеры лапароскопа изображение оперируемой области передается на монитор, что позволяет хирургу в режиме реального времени отслеживать свои действия и продолжать выполнение хирургической операции.

Рассмотрим установку для выполнения операции с применением лапароскопа, как лапароскопическую систему с типовыми элементами и представим ее в терминах теории систем [11 – 13].

$$S = \{ \langle A \rangle, \langle QA \rangle, \langle R \rangle, \langle Q_R \rangle \}, \quad (1)$$

где A – множество элементов $\{ \langle a_1 \rangle, \langle a_2 \rangle, \dots, \langle a_n \rangle \}$ – составляющее основной комплекс оборудования, необходимый для выполнения лапароскопической операции: a_1 – эндхирургическая система освещения; a_2 – эндхирургическая видеосистема; a_3 – лапароскоп; a_4 – манипулятор; a_5 – монитор; a_6 – блок хранения и обработки информации; a_6 – хирургические инструменты.

Q_A – множество параметров, характеризующих свойства элементов системы, определяется набором:

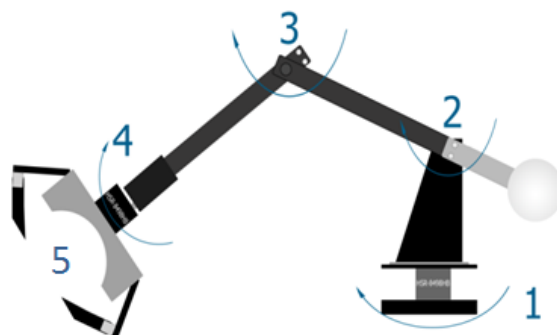


Рисунок 3 – Схематическое представление робота-манипулятора

Кинематическая структура робота-манипулятора представлена на рисунке 4. Робот-манипулятор состоит из звеньев вращательного типа. При этом вращение звеньев манипулятора выполняется вокруг собственной оси на 190^0 . Движение робота-манипулятора осуществляется в сферической системе координат, что обеспечивает обслуживание наибольшей рабочей зоны. Движение рабочего органа (схвата) происходит за счет относительных угловых поворотов звеньев робота.

Для нахождения кинематической модели робота-манипулятора решается классическая прямая задача кинематики, которая состоит в расчете положения и ориентации звеньев манипулятора [18].

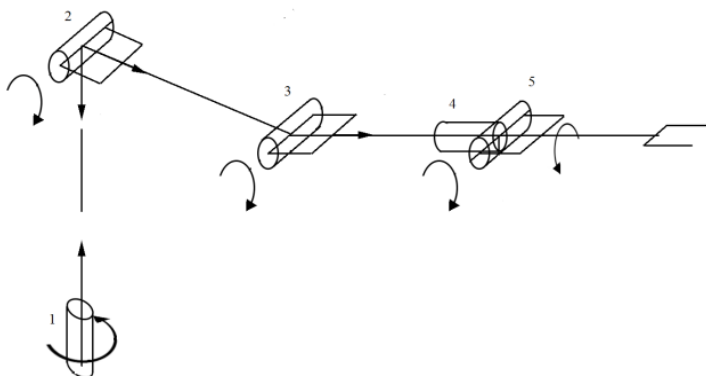


Рисунок 4 – Кинематическая структура манипулятора

Для решения прямой задачи и описания вращательных и поступательных связей между соседними звеньями предложено использовать матричный метод последовательного построения систем координат, связанных с каждым звеном кинематической цепи, определенного как метод преобразования Денавита–Хартенберга [18]. Для этого необходимо сформировать однородную матрицу преобразования, описывающую положение системы координат каждого звена относительно системы координат предыдущего звена, что дает возможность последовательного преобразования координат схвата манипулятора из системы отсчета, связанной с последним звеном, в базовую систему отсчета, которая является инерциальной системой координат для рассматриваемой динамической системы [18] (рисунок 5).

Каждая система координат формируется на основе следующих трех правил [18]:

- 1) ось z_{i-1} направлена вдоль оси i -го сочленения, $i = 1;5$;
- 2) ось x_i перпендикулярна оси z_{i-1} и направлена от нее;
- 3) ось y_i дополняет оси x_i, z_i до правой декартовой системы координат.

Представление Денавита–Хартенберга зависит от геометрических параметров, соответствующих каждому звену, которые полностью описывают любое вращательное или поступательное движение.

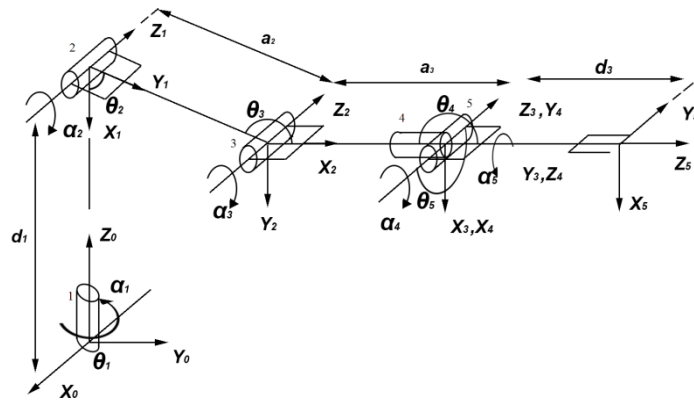


Рисунок 5 – Разложение звеньев робота-манипулятора на основе метода преобразования Денавита–Хартенберга

Основываясь на схеме, которая представлена на рисунке 5, составляется таблица значений параметров звеньев робота-манипулятора. В таблице 1 представлены обобщенные параметры манипулятора.

Таблица 1. Таблица значений параметров звеньев робота-манипулятора

Номер сочленения	θ	d	a	α	Ω
1	q_1	12.8	0	190^0	R
2	q_2	0	15	190^0	R
3	q_3	0	12.2	190^0	R
4	q_4	0	0	190^0	R
5	q_5	0	0	190^0	R

При этом: Ω – тип звеньев; R – звено вращательного типа; α – угол поворота звена манипулятора (постоянная величина); $a(i)$ – длина звена; d – смещение звена; θ – угол между сочленениями манипулятора.

Согласно таблице 1 и рисунку 5 представление Денавита – Хартенберга [18] примет следующий вид:

$$A_{i-1}^i = \begin{pmatrix} C\theta_i & C\alpha_i S\theta_i & S\alpha_i S\theta_i & \alpha_i C\theta_i \\ S\theta_i & C\alpha_i S\theta_i & -S\alpha_i C\theta_i & \alpha_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

где $C_i = \cos(\theta_i + \theta_j + \theta_k)$, $S_i = \sin(\theta_i + \theta_j + \theta_k)$.

Данный способ позволяет определить положение любого i -го звена манипулятора относительно системы координат любой j -й системы координат. После формирования системы Денавита – Хартенберга для всех звеньев составляется матрица преобразования, которая связывает i -ю и $(i-1)$ -ю систему координат [18].

$$T_0^5 = A_0^1 A_1^2 A_2^3 A_3^4 A_4^5. \quad (4)$$

Таким образом, выражение (4) является решением прямой задачи кинематики для шестизвенного манипулятора, который работает в сферической системе координат. Уравнения, полученные в результате решения прямой задачи кинематики, будут использоваться при решении обратной задачи кинематики.

Важной задачей при выполнении проекта является имитационное моделирование, которое позволяет верифицировать кинематическую модель манипулятора и реализовать разрабатываемые методы управления. Все это возможно при использовании современных пакетов прикладных программ. При выполнении проекта для визуализации результатов и задании параметров работы робота-манипулятора предлагается использовать программную среду Matlab с пакетом Robotic Toolbox.

Для построения модели робота-манипулятора используются стандартные функции пакета Robotic Toolbox [19]. Функция Link () создает вектор объектов, данная функция создает звено по входным параметрам. При помощи команды SerialLink соединяются все звенья, которые описаны функцией Link. В результате модель робота-манипулятора примет вид, представленный на рисунке 6.

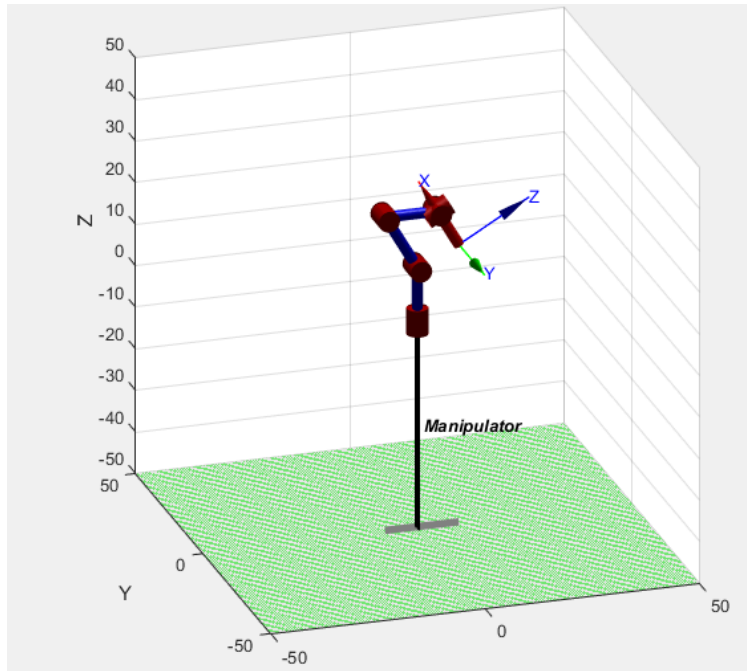


Рисунок 6 – Имитационная модель робота-манипулятора в программной среде Matlab с использованием пакета Robotic Toolbox

В результате проведенной работы была решена прямая задача кинематики пятизвенного робота-манипулятора, которая в дальнейшем позволит решить обратную задачу кинематики и перейти к моделированию робота, апробации алгоритмов управления и позиционирования.

В дальнейшем полученная модель в Matlab будет визуализирована и взаимосвязана с реальным объектом. Результирующая кинематическая модель предлагается для дальнейшего использования при построении алгоритмов управления роботом-манипулятором, а также при отладке и калибровке робота.

Оценка работы робота-манипулятора автоматическом режиме для малоинвазивной хирургии

В общем виде оценку работы робота-манипулятора для автоматического удержания и перемещения лапароскопа с позиции системного подхода можно представить в виде совокупности оценочных параметров:

$$S(O) = \langle t, L, D, P, M \rangle \quad (5)$$

Также работу роботаманипулятора можно оптимизировать при помощи целевой функции [20]:

$$F(O) = w_1 b_t + w_2 b_L + w_3 b_D + w_4 b_P + w_5 b_M \rightarrow \min, \quad (6)$$

- t – время поиска решения, $t = \bar{1}, i$.
- b_t – параметр, характеризующий время поиска решения, $b_t = \frac{\bar{t}_i}{t}$, где \bar{t} – среднее время по результатам выполнения заданного количества экспериментов,
- L – ошибка позиционирования (2 мм). Разность между требуемым положением и текущим, $L = \bar{1}, i$:

$$L: \sqrt{(x_{mp} - x_{mek})^2 + (y_{mp} - y_{mek})^2 + (z_{mp} - z_{mek})^2} \leq 2 \text{ мм},$$

где x_{mp} , y_{mp} , z_{mp} – координаты требуемого положения лапароскопа; x_{mek} , y_{mek} , z_{mek} – координаты текущего положения лапароскопа;

b_L – параметр, характеризующий ошибку позиционирования, $b_L = \frac{\bar{L}_i}{L}$, где \bar{L} – среднее значение ошибки позиционирования по результатам выполнения заданного количества экспериментов, $N = \bar{1}, j$,

D – расстояние от начальной позиции до цели, $D = \bar{1}, d$,

b_D – параметр, характеризующий расстояние или перемещение от начальной позиции до конечной, $b_D = \frac{D_d}{D}$,

где \bar{D} – среднее расстояние от начала пути до цели по результатам выполнения заданного количества экспериментов.

P – показатель безопасности– это минимальное расстояние между любым датчиком робота и любым препятствием вдоль всей траектории, $P \geq 2$ см;

$b_p = \frac{P_p}{\bar{P}}$, где \bar{P} – среднее значение показателя безопасности по результатам выполнения заданного количества экспериментов, $N = \overline{1, j}$,

M – экспертная оценка оптимальности обзора оперируемой области, $M = \overline{1, m}$,

b_M – параметр, характеризующий оптимальность обзора оперируемой области, $b_M = \frac{M_m}{\bar{M}}$, где \bar{M} – среднее

число корректировок или настроек камеры по результатам выполнения заданного количества экспериментов
 $w_z, z = \overline{1, 5}$ – весовые коэффициенты,

Приоритетность оценочных параметров считаем следующей: $P > L > M > t > D$, что отражается в значениях весовых коэффициентов. Расчет целевой функции можно выполнить по результатам заполнения таблицы 2.

Таблица 2. Оценка работы робота-манипулятора в автоматическом режиме

№ теста	t	L	D	P	M	
1	t_1	L_1	D_1	P_1	M_1	
2	t_2	L_2	D_2	P_2	M_2	
...						
N	t_i	L_l	D_d	P_p	M_m	
Среднее значение	\bar{t}	\bar{L}	\bar{D}	\bar{P}	\bar{M}	
Значение весов	$w_1(0,1)$	$w_2(0,25)$	$w_3(0,1)$	$w_4(0,35)$	$w_5(0,2)$	$\sum_{j=1}^N w_z = 1$

Таим образом, рассчитывается значение целевой функции для каждого эксперимента и определяется минимальное значение, что позволяет оценить эффективность работы робота и в последствии при необходимости оптимизировать его показатели.

Заключение

Предложенная структура системы автоматического управления роботом-манипулятором для удержания и перемещения лапароскопа позволяет с точки зрения системного подхода учесть все параметры и свойства системы в целом, а также факторы, влияющие на эффективность рассматриваемого процесса проведения малоинвазивной операции. Использование классических подходов к идентификации кинематических моделей позволяет описывать рассматриваемые конструкции роботов-манипуляторов с требуемой точностью. Также в статье предложена общая оценка функционирования робота-манипулятора для малоинвазивной хирургии, основывающаяся на задании весовых коэффициентов, обеспечивающих минимум целевой функции оценки. Дальнейшие исследования направлены на экспериментальную проверку разработанной модели и апробации методов управления и слежения на действующем оборудовании.

Литература

1. http://robotics.com.ua/shows/series_robots_and_humans/3345-your_health_health_robotics_today Robotics. 2014. Режим доступа: http://robotics.com.ua/shows/series_robots_and_humans/3345-your_health_health_robotics_today
2. <http://medrobot.ru/744-pechalnye-oshibki-medicinskih-robotov.html> Медицинские роботы. Режим доступа: <http://medrobot.ru/744-pechalnye-oshibki-medicinskih-robotov.html>
3. Sim H. G., H. Yip S. K., and Cheng C. W. S. Equipment and technology in surgical robotics // World Journal of Urology, Vol. 24, N. 2, 2006. – p. 128–135.
4. Rane, Kommu S., Eddy B., Rimington P. and Anderson C. Initial experience with the endoassist (r) camera holding robot in laparoscopic urological surgery // European Urology Supplements, Vol. 6, N. 2, 2007. – p. 186–186.
5. Robot Surgery, edited by Seung Hyuk Baik, INTECH, ISBN 978-953-7619-77-0, January 2010. – 172 p.
6. Russell H. Taylor, Kazanzides P. Medical Robotics and Computer-Integrated Interventional Medicine // Elsevier, 2008. – p. 24.
7. Dumpert J. Towards Supervised Autonomous Task Completion Using an in vivo Surgical Robot: dissertation. – University of Nebraska, Lincoln, 2009. – 212 p.
8. Потенциал российских инноваций на рынке систем автоматизации и робототехники: Экспертно-аналитический отчет. – М: ОАО «РВК», 2014. – 128 с.

9. J.E.N. Jaspers. Simple tools for surgeons/ Design and evaluation of mechanical alternatives for “robotic” for minimally invasive surgery // Dissertation, Delft University of Technology, Nederland, 2006. –141 p.
10. Sinyavskaya E.D., Shestova E.A., Medvedev M.Yu., Kosenko E.Yu. Positioning method basing on external reference points for surgical robots // Proceedings of the 1st International Conference on Interactive Collaborative Robotics, Budapest, 2016. – p. 153 – 162.
11. Финаев В.И. Модели систем принятия решений: учебное пособие. – Таганрог: Изд-во ТРТУ, 2005. – 118с.
12. Теория систем и системный анализ: учеб. пособие / В.Н. Чернышов, А.В. Чернышов. – Тамбов: Изд-во Тамб. гос. техн. ун-та, 2008. – 96 с.
13. Bliznyuk Y.Y., Finaev V.I., Kosenko O.V., Shestova E.A., Sinyavskaya E.D. Method of choice of the robot-manipulator of laparoscope controlling in the minimally invasive surgery // International Journal of Applied Engineering Research ISSN 0973-4562, 2016. Vol. 11, Number 9, 2016. – p. 6230-6235
14. Farbod F. Autonomous Robots Modeling, Path Planning, and Control. – Springer Science+Business Media, 2009. – 348 p.
15. Зенкевич С., Ющенко А. Основы управления манипуляционными роботами. – МГТУ им. Н. Э. Баумана, 2004. –.480 с.
16. Москвичев А., Кварталов А., Устинов Б. Захватные устройства промышленных роботов и манипуляторов: учебное пособие. – Форум, 2015.– 176 с.
17. Проектирование роботов и робототехнических систем. Учебное пособие. – Таганрог: Изд-во ЮФУ. Гайдук А.Р. Пшихопов В.Х. Медведев М.Ю. Костюков В.А. Медведева Т.Н., 2013. – 120 с.
18. Шахинпур М.. Курс робототехники. Пер. с англ. – М.: Мир, 1990. – 527 с.
19. Corke P. Robotics, Version and Control: Fundamental Algorithms in MATLAB/ P.Corke. – Springer, 2011. – 570p.
20. Таха Х. А.. Введение в исследование операций. - М.: Вильямс, 2007. – 912 с.

Косенко Е. Ю., Синявская Е.Д., Косенко О.В. Разработка структуры системы автоматического управления роботом-манипулятором для малоинвазивной хирургии. В статье рассмотрены задачи, связанные с разработкой структуры системы автоматического управления роботохирургическими устройствами для малоинвазивной хирургии. С этой целью в статье рассматривается функционирование роботизированного держателя лапароскопа с позиции системного подхода, определяется тип конструкции, идентифицируется кинематическая модель и определяется метод оценивания эффективности его работы.

Ключевые слова: Роботохирургия, лапароскоп, робот-манипулятор, система управления, кинематическая модель, целевая функция

Kosenko E.Y., Sinyavskaya E.D., Kosenko O.V. Development of automatic control system structure of a robot manipulator for minimally invasive surgery. The paper describes the problems associated with development of automatic control system structure robotic surgery systems for minimally invasive surgery. Work of robotic holder of laparoscope is developed in the view of system approach, type of robot is determined, kinematic model is identified and the estimation method of its work effectiveness is offered.

Keywords: Robotic surgery, laparoscope, robot manipulator, control system, kinematic model, objective function.

Согласованная оптимизация целевой программы развития промышленного предприятия

Г.П. Виноградов, В.Н. Кузнецов
Тверской государственной технической университет
wgp272ng@mail.ru

Виноградов Г.П., Кузнецов В.Н. Согласованная оптимизация целевой программы развития промышленного предприятия. В статье рассматривается проблема согласования интересов в организационно-экономических, эффективность функционирования которых в значительной степени определяется поведением производственного персонала. Рассматриваемая в данной статье задача научных исследований отличается от известных решений тем, что в ней рассматривается согласованная оптимизация, как на множестве компромиссных решений, так и на множестве кооперативных решений. Такой подход позволил разработать методы и алгоритмы согласованного управления, заключающиеся в разработке, принятии и реализации проектов ЦПР промышленного предприятия, обеспечивающих на рассматриваемом периоде времени максимальный гарантированный социально – экономический результат при условии удовлетворения интересов собственников, инвесторов, руководства и персонала предприятия. Кооперация является необходимым элементом совместной деятельности. Алгоритмическое предписание решения дискретной задачи согласованной оптимизации строится на основе алгоритма решения многокритериальной дискретной задачи оптимального управления для функции полезности, ядром которого являются градиентные методы математического программирования.

Ключевые слова: согласованная оптимизация, целевая программа развития, активные субъекты, информационное управление.

Введение

В статье рассматриваются формализация и постановки задач согласованной оптимизации целевой программы развития промышленного предприятия (ЦПР). При этом применяется методология согласования интересов в экономических информационных системах [10], разработанная в Тверском государственном техническом университете совместно с Институтом проблем управления им. В.А. Трапезникова РАН. Предполагается, что отраслевые условия определяют формат и структуру ЦПР, а также традиции, субъекты и правила принятия решений.

1 Применяемый подход согласования интересов

1. Методология согласования интересов в экономических информационных системах отличается от других подходов согласования экономических интересов, например, от [2, 4, 6, 7, 8, 21, 22], следующим:

1. Ставится и решается проблема повышения эффективности управленческих решений на основе разработки и применения формальных методов и средств обработки и отображения информации, и согласованной оптимизации.

2. Стратегия исследования. Модели теории активных систем дополняются необходимыми для описания человеко-машинных процессов согласования и оптимизации составляющими. Это

- Модели человеко-машинных процессов согласования и оптимизации на языке теории активных систем.

- Модели задач многокритериальной оптимизации с неизвестными функциями полезности, описывающими интересы пользователей активной системы, на языке эвристических процедур и теории активных систем.

- Эвристические алгоритмические модели человеко-машинных процессов согласования и оптимизации в виде алгоритмов "сводимости", т. е. в виде алгоритмических предписаний.

- Формальные алгоритмические модели человеко-машинных процессов решения экстремальных задач, описываемые с помощью алгоритмических методов оптимизации.

- Модель устойчивости коллективной стратегии пользователей активной системы на языке теории игр с непротивоположными интересами.
- Математические модели экстремальных задач, решаемых на каждой итерации человеко-машинных процессов, на языке исследования операций и активных систем.
- Структурные модели информационных процессов, описывающие информационную технологию согласования и оптимизации на языке структурного программирования.
- Экспериментальные модели человеко-машинных процессов согласования и оптимизации в виде компьютерных моделей и имитационных игр в активных системах.
- Модели интеллектуальной поддержки согласованной оптимизации [12, 17].

Управляющей и интегрирующей средой этих моделей друг с другом и с моделями теории активных систем является концептуальная модель человеко-машинных процессов согласования и оптимизации, дающая содержательное представление о существенных свойствах этих процессов и главных связях между ними.

3. Применяется, разработанный на основе принципа согласованного управления [5], принцип согласованной оптимизации на множестве компромиссных решений [11]. Принцип согласованной оптимизации это композиция: {коалиция согласования решений + принцип оптимального согласованного планирования состояний коалиции + человеко-машинная процедура согласования и согласованной оптимизации на множестве компромиссных решений + описание задачи согласованной оптимизации на множестве компромиссных решений + схема функционирования активной системы с коалицией согласования решений + коллективная стратегия согласованной оптимизации, обеспечивающая устойчивость коллективных решений}. За счет применения этого принципа получают решения по Нейману - Моргенштерну. Эти решения принадлежат множеству компромиссных решений, обладающему свойствами внутренней устойчивости, состоящей в том, что полученные оптимальные согласованные решения нельзя противопоставлять друг другу, и внешней устойчивости, состоящей в возможности каждому не принадлежащему решению по Нейману - Моргенштерну и претендующему на роль более эффективного противопоставить оптимальное согласованное решение, принадлежащее этому же множеству.

Алгоритмические предписания человеко-машинных процессов согласованной оптимизации строятся на основе алгоритмов выпуклого программирования. Для этого применяются метод Франка – Вольфа [9] и градиентные методы. Они обладают хорошей сходимостью на первых шагах поиска и устойчивы к выбору направления и шага спуска. Это обеспечивает сходимость человеко-машинных процессов и устойчивость скорости сходимости к точности определения градиентов на каждом шаге поиска. В этом случае на каждом шаге поиска можно применять или субъективные оценки направления движения в пространстве критериев или субъективные оценки в виде коэффициентов их значимости.

4. Можно осуществлять разработку на стадии проектирования автоматизированной экономической информационной системы концептуальных, информационных ее моделей, математических, алгоритмических и структурных моделей информационных процессов анализа, согласования и оптимизации для конкретной экономической системы, моделей представления и отображения информации на языке пользовательского интерфейса и языке деловой графики.

5. Методология позволяет формализовать некоторые концептуальные модели экономических и гуманитарных наук, например, [14].

С помощью этой методологии были поставлены и решены задачи научных исследований согласованного управления проектами вооружения и военной техники ВМФ России [3], согласованной оптимизации бюджетных целевых программ региона [15,18], развития на множестве компромиссных решений информационного управления финансово - хозяйственной деятельности предприятий [13], согласованной оптимизации имущественно – земельного комплекса [23], согласования экономических интересов в анализе деятельности проектных дорожных организаций с целеустремленными агентами [19] и др.

2 Критерии и математическое описание согласованной оптимизации целевой программы промышленного предприятия

Задача согласованного управления заключается в разработке, принятии и реализации проектов ЦПР промышленного предприятия, обеспечивающих на рассматриваемом периоде времени максимальный гарантированный социально – экономический результат при условии удовлетворения интересов собственников, инвесторов, руководства и персонала предприятия.

Проект ЦПР включает в себя сведения о составе и наименовании стратегических целей, тактических задач и о их содержании; о сущности, выражении и значениях целевых показателей; о распределении финансирования по целям и задачам; о перечне и наименовании мероприятий по достижению задач, о их содержании и о перечне, наименовании и значениях факторных показателей мероприятий.

Математическое описание цели согласованного управления в социальных и экономических системах на множестве компромиссных и кооперативных решений для разработки ЦПР будет иметь следующий вид.

$$p \times \max_{y \in P(x,f)} f_0^z(x,y) + (1-p) \times \min_{y \in P(x,f)} f_0^z(x,y) \rightarrow \max_{x \in X(a)},$$

$$P(x, f_0^z) = \prod_{i \in I} P_i(x_i, y_i), \quad \mathbb{F}(x_i, f_i^z) = \text{Arg max}_{y_i \in Y_i} f_i^z(x_i, y_i),$$

$$x \in X(a), \quad y_i \in Y_i(b_i), \quad X(a) = \prod_{i \in I} X_i(a_i), \quad Y(b) = \prod_{i \in I} Y_i(b_i),$$

$$f_0^z(x,y) = \min_{a \in A} f_0(x,y), \quad f^z(x_i, y_i) = \min_{b_i \in B_i} f_i(x_i, y_i),$$

$$x \in X(a), \quad y_i \in Y_i(b_i)$$

где: $f_0(\cdot)$ - целевая функция, математически описывающая с помощью функции удельной ценности предпочтения собственников предприятия социально – экономических его состояний; $f_i(\cdot)$ - целевая функция i -го целеустремленного агента, участвующего в процессе реализации ЦПР, математически описывающая его интересы; p – коэффициент сотрудничества целеустремленных агентов, участвующих в процессе реализации ЦПР; $x = (x_1, \dots, x_i, \dots, x_h)$ – составляющие проекта ЦПР предприятия; $x_i = (x_{i1}, \dots, x_{ij}, \dots, x_{im})$ – составляющие проекта ЦПР по i -ой цели; $y = (y_1, \dots, y_i, \dots, y_h)$ – реализация проекта ЦПР предприятия; $y_i = (y_{i1}, \dots, y_{ij}, \dots, y_{im})$ – реализация проекта ЦПР по i – ой цели; $X(a)$ - множество допустимых проектов ЦПР; $a = (a_1, \dots, a_i, \dots, a_h)$ – вектор факторных параметров, определяющих ограниченное множество допустимых проектов ЦПР; $X_i(a_i)$ - множество допустимых проектов ЦПР; $a_i = (a_{i1}, \dots, a_{ij}, \dots, a_{ik})$ - вектор факторных параметров, определяющих ограниченное множество допустимых проектов ЦПР по i – ой цели; $P(x, f)$ - множество решений игры целеустремленных агентов или множество всех их локально-оптимальных состояний; $P_i(x_i, f_i)$ - множество локально-оптимальных состояний i -го целеустремленного агента; $Y_i(b_i)$ - множество реализаций проектов ЦПР по целям; $b_i = (b_{i1}, \dots, b_{ij}, \dots, b_{ik})$ - вектор факторных параметров, определяющих ограниченное множество допустимых значений факторных показателей i -го целеустремленного агента.

Представления у целеустремленных агентов, участвующих в разработке ЦПР, о согласовании интересов можно формализовать в виде множества согласованных значений факторных показателей.

$$X_c[n] = \{ x \mid f_{oi}^z(x[n], y[n]) \geq f_{oi}^z(x[n=0], y[n=0]) + \varphi_{oi}(x[n], y[n]),$$

$$\varphi_{oi}(x[n], y[n]) \geq \sum_{n \in N} \Delta_{oi}[n], \quad x[n] \in X, \quad (y=x) \cup (y \in Y(x)),$$

$$\Delta_{oi}[n] > 0, \quad n \in N, \quad N = \{ n \mid n = 1, 2, \dots, n^* \}$$

с использованием функций выигрыша $\varphi_{o,i}(x,y)$ элементов активной системы относительно исходного состояния, гарантируемых им центром при вступлении в коалицию. Эта функция представляет также функцию нечувствительности относительно выбора элементами активной системы (x,y) или $(x[n=0], y[n=0])$ в исходной точке поиска для $n=0$ с учетом входных барьеров, затрат на поиск и ограниченных возможностей человеко-машинных ресурсов, или относительно решений задач оптимального или оптимального согласованного планирования, определяющих гарантированные значения $\nabla_{o,i} > 0$ функции нечувствительности.

Эти выражения означает, что согласованное управление ЦПР определяет планируемые способы действия исполнителей, которые являются лучшими для собственников, руководства и персонала предприятия по их функциям предпочтения, отражающим их интересы.

Для учета применяемых целеустремленными агентами стратегий сотрудничества и компромисса запишем их функции выигрыша следующим образом.

$$\varphi_i(\cdot) = d_{io} \times f_0^z(\cdot) + \sum_j d_{ji} \times f_i^z(\cdot), \quad d_{io} + \sum_j d_{ji} = 1.$$

Коэффициенты d_{io} представляют коэффициенты сотрудничества активных элементов с хозяйствующим субъектом, а d_{ij} - коэффициенты компромисса. Если все $d_{ij} = 0$ и $d_{io} = 1$, то это стратегия сотрудничества. Если $d_{ij} > 0$ и $d_{io} < 1$, то это стратегия компромисса.

Если выполняется предположение, что в условиях возможности конфликта $(x, y) \notin X_c$ обязательно множество решений игры элементов $R(x, f) \in X_c(x)$, то тогда

$$\max \{ f_0^z(x,y) \mid x \in X, y \in Y, (x, y) \in X_c \} \geq \max \{ \min f_0^z(x,y) \mid y \in R(x, f) \subseteq X_c(x) \mid x \in X \},$$

где множество решений игры элементов $R(x, f)$ образуют множества всех их локально-оптимальных состояний $R_i(x, f_i)$.

$$R(x, f^c) = \prod_{i \in I} R_i(x_i, f_i^c), \quad R_i(x, f) = \text{Arg} \max_{y \in X_c} f_i^c(x, y),$$

$$X_c(x) \in \{y \mid x \in X, (x, y) \in X_c\}.$$

Из этого соотношения следует, что применение согласованного управления ЦПР не менее эффективно по сравнению со всеми описанными и примененными на практике процедурами согласованной оптимизации.

Процесс согласованного управления ЦПР имеет следующую структуру: согласованная оптимизация проектов ЦПР, разработка и утверждение целевой программы, формирование на конкурсной основе контрактов, мониторинг реализации целевой программы. В настоящей работе рассматривается только согласованная оптимизация информационного содержания проектов по ЦПР, которое включает состав и наименования стратегических целей и тактических задач, их содержание, распределение финансирования по целям и задачам, перечень и наименование мероприятий по достижению задач, их содержание, перечень, наименование и значения факторных показателей мероприятий.

Реализация данной цели осуществляется в процессе продвижения, который состоит из следующих этапов.

1. Анализ социально – экономического состояния промышленного предприятия за прошлый и текущий периоды.
2. Определение направления движения от исходного состояния в пространстве стратегических целей, тактических задач и мероприятий по выполнению ЦПР.
3. Определение направления движения от исходного состояния в пространстве финансирования ЦПР по тактическим задачам.
4. Определение направления движения от исходного состояния в пространстве факторных показателей мероприятий по достижению тактических задач при заданном их финансировании.
5. Оценка удовлетворительности ЦПР и анализ слабых и сильных ее сторон последовательно группой стратегического анализа, командной группой предприятия, советом по развитию, в который входят представители собственников, руководства и персонала предприятия.
6. Остановка процесса. Принятие, доработка или непринятие ЦПР.

Для оценки продвижения и ЦПР используются показатели продвижения к цели согласованного управления ЦПР, определяемые как приращения функций удельной ценности социально - экономических состояний за анализируемый, прогнозируемый или плановый периоды. Понятие удельной ценности результатов было введено в системном подходе при описании поведения целеустремленных систем [1]:

$w = w_o(w_1, \dots, w_i, \dots, w_n)$ - удельная ценность социально – экономического состояния предприятия,

$$i \in I = I_n \cap I_z \cap I_a \cap I_{om};$$

$w_i = w_i(k_{i1}, \dots, k_{ij}, \dots, k_{im_i})$ - удельная ценность социально – экономического состояния предприятия по

целям, $j \in J = J_n \cap J_z \cap J_a \cap J_{om};$

$k_{ij} = k_{ij}(x_{11k}, \dots, x_{ijk}, \dots, x_{nm_i})$ - удельная ценность социально – экономического состояния предприятия

по задачам, $m \in M = M_n \cap M_z \cap M_a \cap M_{om};$

x_{ijk} - факторный показатель k -го мероприятия;

$s_o, s_i, s_{ij}, s_{ijk}$ - общие финансовые затраты на достижение удельных ценностей предприятия, по целям, по задачам и по мероприятиям;

$$k_{ij} = k_{ij}^s(s_{ij}), \quad x_{ijk} = x_{ijk}^s(s_{ijk}), \quad s_{ijk} = s_{ijk}^x(x_{ijk});$$

$\Delta w, \Delta w_i, \Delta k_{ij}$ - целевые показатели (темпы прироста) развития предприятия, стратегических целей и тактических задач;

Δx_{ijk} - темп прироста факторного показателя мероприятия;

$\Delta s_o, \Delta s_i, \Delta s_{ij}, \Delta s_{ijk}$ - дополнительное финансирование темпов прироста;

$$\Delta w \cong \sum_{i \in I} \sum_{j \in J} (\partial w_o / \partial w_i) \times (\partial w_i / \partial k_{ij}) \times \Delta k_{ij},$$

$$\Delta w \equiv \sum_{i \in I} \sum_{j \in J} (\partial w_o / \partial w_i) \times \partial w_i / \partial k_{ij} \times (\partial k_{ij} / \partial s_{ij}) \times \Delta s_{ij},$$

$$\Delta w_i \equiv \sum_{j \in J} (\partial w_i / \partial k_{ij}) \times \Delta k_{ij}, \quad \Delta w_i \equiv \sum_{j \in J} (\partial w_i / \partial k_{ij}) \times (\partial k_{ij} / \partial s_{ij}) \times \Delta s_{ij},$$

$$\Delta k_{ij} = (\partial k_{ij} / \partial s_{ij}) \times \Delta s_{ij}, \quad \Delta k_{ij} \equiv \sum_{k \in K} (\partial k_{ij} / \partial x_{ijk}) \times \Delta x_{ijk},$$

3 Метод и модель согласованной оптимизации состава стратегических целей, тактических задач и мероприятий

Задача согласованной оптимизации состава стратегических целей и тактических задач по выполнению ЦПР предприятия описывается моделью.

$$CO(\Delta w(r_w, r_k, k^*)) \rightarrow \max, r_w = \{r_i | i = 1_w, 2_w, \dots, n_w\}, r_k = \{r_{ij} | j = 1_{ik}, 2_{ik}, \dots, m_{ik}\},$$

$$\Delta k^* = \{\Delta k_{ij}^*\} \in \Delta K^*, \sum_{i \in I} r_i \leq n_w < n^*, \sum_{j \in J_i} r_{ij} \leq m_{ik} < m_i^*$$

$$CO(\Delta w(r_w, r_k, k^*)) \equiv \sum_{i \in I} \sum_{j \in J_i} CO(\partial w_o / \partial w_i) \times r_i \times CO(\partial w_i / \partial k_{ij}) \times r_{ij} \times CO(\Delta k_{ij}^*),$$

$$CO(\Delta w_i(r, k^*)) \equiv \sum_{j \in J} CO(\partial w_i / \partial k_{ij}) \times r_{ij} \times CO(\Delta k_{ij}^*) \geq CO(\Delta_i^y).$$

$$i \in CO(I) = CO(I_c) \cap CO(I_{un}) \cap CO(I_p) \cap CO(I_n) = CO\{i | i = 1, 2, \dots, n^*\},$$

$$j \in CO(J_i) = CO(J_{ci}) \cap CO(J_{uni}) \cap CO(J_{pi}) \cap CO(J_{ni}) = CO\{j | j = 1_i, 2_i, \dots, m_i^*\},$$

$$r_i = \begin{cases} 1, & \text{если выбирается цель } i; \\ 0, & \text{в противном случае.} \end{cases} \quad r_{ij} = \begin{cases} 1, & \text{если выбирается задача } ij; \\ 0, & \text{в противном случае.} \end{cases}$$

где: $CO(\dots)$ – субъективная оценка, $\Delta w(r_w, r_k, k^*)$ – темпы прироста социально – экономического результата предприятия за рассматриваемый период времени, r_w – множество выбранных целей; r_k – множество выбранных задач, $\Delta k^* = \{\Delta k_{ij}^*\} \in \Delta K^*$ – вектор непрерывных числовых переменных, компонента которого определяет значение желаемого темпа прироста по задаче j цели i , $\partial w_o / \partial w_i$ – весовой коэффициент (значимости) цели i по темпу прироста социально – экономического результата предприятия, равный частной производной функции социально – экономического результата предприятия по функции результата по цели i , $\partial w_i / \partial k_{ij}$ – весовой коэффициент (значимости) задачи j по темпу прироста цели i , равный частной производной функции результата предприятия для цели i по функции результата по задаче j , I_c, I_{un}, I_p, I_n, I – множество возможных целей, желаемых собственниками, инвесторами, руководством и персоналом предприятия и их пересечение, $J_c, J_{un}, J_p, J_{on}, J$ – множество возможных задач, желаемых собственниками, инвесторами, руководством и персоналом предприятия и их пересечение.

Задача согласованной оптимизации состава стратегических целей, тактических задач по выполнению ЦПР предприятия заключается в определении таких целей и задач из заранее построенных множеств, и таких темпов прироста результатов социально – экономического развития по задачам, которые определяют оптимальное значение темпа прироста социально – экономического состояния предприятия. Множество строится заранее по результатам научных и социологических исследований, по результатам диагностики собственников, инвесторов, руководства и персонала с применением методов практической социальной психологии, по результатам обследования предприятия и по результатам проводимых конкурсов предложений на лучшие цели и задачи.

Создается центр ответственности (ЦО) за разработку ЦПР, стратегических целей и тактических задач. При центре ответственности создается фокус – группы. Проводятся заседания по сессиям в виде совещаний и определяются в интерактивном процессе субъективные оценки множеств возможных целей $CO(I) = CO(I_c) \cap CO(I_{un}) \cap CO(I_p) \cap CO(I_n)$ собственников, инвесторов, руководства, персонала

предприятия $CO(J_i) = CO(J_{ci}) \cap CO(J_{ini}) \cap CO(J_{pi}) \cap CO(J_{ni})$; субъективные оценки весовых коэффициентов $CO(\partial w_o / \partial w_i)$ целевого показателя развития предприятия (темпа прироста социально – экономического состояния предприятия) Δw_o по целевым показателям стратегических целей (темпы прироста) Δw_i ; весовых коэффициентов $CO(\partial w_i / \partial k_{ij})$ целевых показателей стратегических целей w_i по целевым показателям тактических задач (темпы прироста) Δk_{ij} , целевых значений темпов прироста социально – экономического состояния отрасли по тактическим задачам Δk_{ij}^* ; условий согласования по целевым показателям целей $CO(\Delta_i^w)$ и задач $CO(\Delta_{ij}^w)$. При этом применяется метод фокус – групп.

Фокус - группы предполагают меньшие затраты времени и средств. Предполагают анализ слов и образов, а не цифр. Применяется наблюдение и смысловая характеристика. Основным методом фокус – групп является групповая дискуссия, которая имеет следующие организационные этапы.

- определение целей и темы дискуссия;
- сбор информации (сведений, знаний, суждений, мнений, новых идей, предложений всех участников дискуссии) по обсуждаемой проблеме;
- упорядочивание, интерпретация и оценка полученной в ходе дискуссии информации;
- подведение итогов: сопоставление целей дискуссии и ее результатов.

Количество участников 4 – 5 человек. Выделяется модератор, который является лидером группы. Должна быть максимальная однородность членов группы по интересам. Это позволяет обеспечить сотрудничество и лояльность друг к другу. Целью фокус – группы является получение «субъективной информации», а не достижение единства мнений.

Если пересечение окажется пустым, то делается вывод о наличии конфликта. Деструктивный конфликт не допустим [2]. В этом случае применяются жесткие меры по реструктуризации руководства предприятия, по совершенствованию системы их стимулирования, по пересмотру кадровой политике предприятия и др. Продуктивный конфликт играет положительную роль [2] ускорения процесса принятия решений и повышения его эффективности. Но он допустим только в процессе согласованной оптимизации. При формировании окончательного решения он должен быть решен. Для этого предусмотрен внешний цикл алгоритмического предписания согласованной оптимизации целей и задач, цикл по сближению интересов. Он включает поиск ЦО способов согласования интересов: поиск общих потребностей и опасений, изменение приоритетов, состыковка интересов, оказание взаимных услуг, осуществление неспецифических компенсаций и применение санкций. После этого оказывается влияние ЦО в процессе общения на собственников, руководство, инвесторов и персонал. Проводятся беседы, совещания, консультации, дискуссии, переговоры, тренинги, коучинг и компьютерные презентации.

После этого на каждом шаге интерактивного процесса осуществляется решение задач оптимизации субъективной оценки целевого показателя $CO(\Delta w)$ и согласование субъективных оценок целевых показателей $CO(\Delta w_i)$ или методом линейного целочисленного программирования [20] или методом компьютерного моделирования, и определения информационного содержания целей и задач, значений Δk_{ij}^* .

Если существуют сильные ограничения по времени, специалистам, трудовым и финансовым затратам, по общению с собственниками, инвесторами, руководством и персоналом, то задача согласованной оптимизации упрощается. В этом случае применяются последовательный анализ вариантов [24], используемый при проектировании сложных систем, компьютерное моделирование и следующая схема согласованной оптимизации. Центром ответственности формируется первый вариант целей и задач. Осуществляется его компьютерное моделирование и презентация (оценка собственниками, инвесторами, руководством и персоналом). Определяются слабые и сильные его стороны. Формируется второй вариант целей и задач и т.д. Когда накапливается достаточное количество вариантов, то осуществляется их оценка и групповой выбор на совещаниях, советах и конференциях. Может быть создана комиссия.

После определения стратегических целей формируются центры ответственности по целям (ЦО_i).

4 Метод и модель согласованной оптимизации ресурсов целевой программы

Модель согласованной оптимизации ресурсов ЦПР на множестве компромиссных и кооперативных решений собственников, инвесторов, руководства, исполнителей ЦПР, органа оценки качества и принятия ЦПР по тактическим задачам с использованием субъективных оценок центра ответственности по программе, по целям будет иметь следующий вид.

$$CO(\Delta w(\Delta s_{ij})) \cong \sum_{i \in I} \sum_{j \in J_i} CO(\partial w_o / \partial w_i) \times CO(\partial w_i / \partial k_{ij}) \times$$

$$\times CO(\partial k_{ij} / \partial s_{ij}) \times \Delta s_{ij} \xrightarrow{\Delta s_{ij}} \max, \quad c_o \geq \Delta s_o = \sum_{i \in I} \sum_{j \in J_i} \Delta s_{ij} \geq CO(\Delta_o^s),$$

$$CO(\Delta w_i(\Delta s_{ij})) \cong \sum_{j \in J_i} CO(\partial w_i / \partial k_{ij}) \times CO(\partial k_{ij} / \partial s_{ij}) \times \Delta s_{ij} \geq CO(\Delta_i^w),$$

$$CO(\Delta k_{ij}) \geq \Delta k_{ij}^*, \quad \Delta s_i = \sum_{j \in J_i} \Delta s_{ij} \geq CO(\Delta_i^s),$$

$$I_w = \{i | i = 1_w, 2_w, \dots, n_w\}, \quad J_i = \{j | j = 1_{ik}, 2_{ik}, \dots, m_{ik}\}.$$

где: $CO(\cdot)$ – субъективная оценка, Δs_{ij} – переменная, определяющая размер финансирования задачи j цели i , $\Delta w(\Delta s_{ij})$ – функция, значения которой определяют темпы прироста социально – экономического результата предприятия от размеров финансирования по задачам, $\partial k_{ij} / \partial s_{ij}$ – темп прироста результата по задаче j цели i , приходящийся на единицу финансирования этой задачи, $\Delta w_i(\Delta s_{ij})$ – функция, значения которой определяют темпы прироста социально – экономического результата цели i от размеров финансирования по задачам, Δs_{ij} – размер финансирования по задаче j цели i , c_o – ограничения на финансовые ресурсы, $\Delta s_o = \sum_{i \in I} \sum_{j \in J_i} \Delta s_{ij}$ – финансирование по ЦПР, Δ_o^s – желаемый размер финансирование по ЦПР, Δk_{ij} – темп прироста результата по задаче j , Δk_{ij}^* – желаемый темп прироста результата по задаче j , $\Delta s_i = \sum_j \Delta s_{ij}$ – размер финансирования по

цели i , Δ_i^s – желаемый размер финансирование по цели i .

Задача согласованной оптимизации ресурсов ЦПР на множестве компромиссных и кооперативных решений заключается в определении таких размеров финансирования по задачам, которые определяют оптимальное значение темпа прироста социально – экономического состояния предприятия. Ограничения на финансовые ресурсы определяет финансовая служба предприятия с учетом фонда развития, поступающей выручки, возможностей взятия кредитов и возможностей инвестирования. Значения ограничений приближительны и варьируются, за счет чего формируются различные варианты.

5 Модель согласованной оптимизации центром ответственности показателей мероприятий по стратегическим целям

Модель согласованной оптимизации ЦО_i показателей мероприятий по стратегическим целям на множестве компромиссных и кооперативных решений с использованием субъективных оценок ЦО_i по целям и задачам будет иметь следующий вид.

$$CO(\Delta k_{ij}) \cong \sum_{k \in K} CO(\partial k_{ij} / \partial x_{ijk}) \times \Delta x_{ijk} \xrightarrow{\Delta x_{ijk} \geq \Delta_{zn}} \max,$$

$$CO(\Delta k_{ij}) \geq \Delta k_{ij}^*, \Delta s_{ij} \leq \sum_{k \in K} CO(\partial k_{ij} / \partial x_{ijk}) \times \Delta x_{ijk} \geq CO(\Delta_{ij}^s),$$

$$i \in I_w = \{i | i = 1_w, 2_w, \dots, n_w\}, \quad j \in J_{ik} = \{j | j = 1_{ik}, 2_{ik}, \dots, m_{ik}\}.$$

где: $CO(\cdot)$ – субъективная оценка, Δk_{ij} – темп прироста результата по задаче j цели i , $k \in K$ – индекс показателя мероприятия и множество их значений, Δx_{ijk} – темп прироста результата показателя мероприятия k по задаче j цели i , Δ_{zn} – значимое значение прироста, Δ_{ij}^s – желаемый размер финансирования задачи j цели i , $\partial s_{ijk} / \partial x_{ijk}$ – финансирование мероприятия k по задаче j цели i на единицу темпа прироста результата показателя мероприятия k по задаче j цели i .

Задача согласованной оптимизации ЦО_i показателей мероприятий по стратегическим целям на множестве компромиссных и кооперативных решений центра ответственности по целям и задачам заключается в определении таких размеров темпов прироста результата по мероприятиям, которые определяют оптимальное значение темпа прироста социально – экономического состояния по задачам. При этом определяются и перечень мероприятий, для которых $\Delta x_{ijk} \geq \Delta x_{зн}$.

При ЦО_i создаются фокус – группы. Проводятся заседание по сессиям в виде совещаний и определяются в интерактивном процессе субъективные оценки весовых коэффициентов $CO(\partial w_o / \partial w_i)$ целевого показателя развития предприятия (темпа прироста социально – экономического состояния предприятия) Δw_o по целевым показателям стратегических целей (темпы прироста) Δw_i ; весовых коэффициентов $CO(\partial w_i / \partial k_{ij})$ целевых показателей стратегических целей w_i по целевым показателям тактических задач (темпы прироста) Δk_{ij} , целевых значений темпов прироста социально – экономического состояния предприятия по тактическим задачам Δk_{ij}^* ; весовых коэффициентов $CO(\partial k_{ij} / \partial x_{ijk})$ целевых показателей тактических задач по темпам прироста факторных показателей мероприятий x_{ijk} ; субъективные гарантированные оценки $CO(\partial k_{ij} / \partial s_i)$ темпов прироста по задаче ij на единицу финансирования Δs_i ; субъективные гарантированные оценки $CO(\partial s_{ijk} / \partial x_{ijk})$ темпов прироста финансирования на единицу факторного показателя мероприятия x_{ijk} ; условий согласования по целевым показателям целей $CO(\Delta_i^w)$ и задач $CO(\Delta_{ij}^w)$, по финансированию ЦПР $CO(\Delta_o^s)$, целей $CO(\Delta_i^s)$ и задач $CO(\Delta_{ij}^s)$; субъективные оценки весовых коэффициентов критериев оценки ЦПР $CO(\partial W / \partial K_l)$. При этом применяется метод фокус – групп.

После этого на каждом шаге интерактивного процесса осуществляется решение задачи линейного программирования или проводится компьютерное моделирование и групповая оценка, и определение информационного содержания мероприятий, значений Δx_{ijk} .

6 Модель согласованной оптимизации информационного содержания целевой программы на основе оценки ее удовлетворительности и анализа слабых и сильных ее сторон

Задачи согласованной оптимизация информационного содержания ЦПР на основе оценки ее удовлетворительности в плане [1] и анализа слабых и сильных ее сторон будет иметь вид.

$$CO(\Delta w(\Delta P_l(x, z^*))) = \sum_{l \in L} CO(\partial W / \partial P_l) \times CO(\Delta P_l(x, z^*)) \rightarrow \max,$$

$$x \in CO(X_c) \cap CO(X_{ин}) \cap CO(X_p), l \in CO(L_c) \cap CO(L_{ин}) \cap CO(L_p),$$

где: $CO(.)$ – субъективная оценка, $\Delta w(\Delta P_l(x, z^*))$ – функция, определяющая значение темпа прироста социально – экономического результата предприятия от значений X, Z^* информационного содержания ЦПР и требований к ней, $\Delta P_l(x, z^*)$ – функция, определяющая значение темпа прироста критериев социально – экономического результата предприятия $\Delta P_l, l \in L$ от значений x, z^* информационного содержания ЦПР и требования к ней, x, z^* – информационное содержание ЦПР и требования к ней, $\partial W / \partial P_l$ – весовой коэффициент (значимости) по темпу прироста критерия l социально – экономического результата предприятия, равный частной производной функции социально – экономического результата региона по функции критерия l , $X_c, X_{ин}, X_p$ – возможные множества информационных содержаний ЦПР по представлениям персонала, собственников и руководства, $L_c, L_{ин}, L_p$ – возможные множества критериев ЦПР по представлениям собственников, инвесторов и руководства.

Задача согласованной оптимизации информационного содержания ЦПР на основе оценки ее удовлетворительности и анализа слабых и сильных ее сторон заключается в определении таких информационных содержаний ЦПР, которые определяют удовлетворительное для целеустремленных агентов (собственников, инвесторов и руководства) [1] значение темпа прироста социально – экономического состояния предприятия:

$w(x) = W[P(x)]$ - удельная ценность социально – экономического состояния региона или отрасли,
 $P(x)$ - критерии оценки ЦПР,

x, z^* - информационное содержание ЦПР и требования к ней.

Осуществляется оценка удовлетворительности ЦПР центрами ответственности предприятия и потом советом по развитию предприятия с использованием субъективных оценок $CO(X) = CO(X_H) \cap CO(X_3) \cap CO(X_a)$, $CO(L) = CO(L_H) \cap CO(L_3) \cap CO(L_a)$ и $CO(\partial W / \partial P_i)$. Если условие удовлетворительности не выполняется, то корректируются субъективные оценки (в первую очередь условия согласования) и снова осуществляется решение задач оптимизации. Если удовлетворяется, то ЦПР сначала утверждается генеральным директором, а затем принимается собранием собственников. При этом применяется метод фокус – групп.

Заключение

В статье приведены научные результаты, полученные при решении задачи научных исследований согласованной оптимизации целевой программы развития промышленного предприятия. Это формализация и постановки задач согласованной оптимизации. Рассмотрены также концептуальные схемы их решения. Полученные результаты были внедрены на АО «Лихославский завод светотехники».

Литература

1. Акофф Р., Эмери Ф. О целеустремленных системах. – М.: «Советское радио», 1974. – 274 с.
2. Андреева Г.М. Социальная психология. – Изд. 3. М.: Наука, 1994. – 325 с.
3. Арбузов В.А. Согласованное управление проектами ВВТ ВМФ России.// Программные продукты и системы. –2002. - №3. – стр. 45-48.
4. Гаррет Б., Дюссож П. Стратегические альянсы: *Пер. с англ.* – М.: ИНФРА-М, 2002. – 332 с.
5. Бурков В.Н. Основы математической теории активных систем. – М.: «Наука», 1977. – 255 с.
6. Васильев Н.Н. Тренинг преодоления конфликтов. – СПб.: Речь. 2002. –174 с.
7. Введение в практическую социальную психологию. Под ред. Ю.М. Жукова, Л.А. Петровской, О.В. Соловьевой. – М.: Наука, 1994. – 255 с.
8. Гераскин М.И. Согласование интересов в корпоративных структурах. – М.: ИПУ РАН. Изд-во «Анко». 2005. – 293 с.
9. Джофрион А., Дайер Дж., Файнберг А. Решение задач оптимизации при многих критериях на основе человеко-машинных процедур. – В кн.: Вопросы анализа и процедуры принятия решений. М.: Мир, 1976.
10. Кузнецов В.Н. Согласование и оптимизация в иерархических системах с активными элементами. – М.: Институт проблем управления, 1996. – 132 с.
11. Кузнецов В.Н., Ключин А.Ю. Принцип согласованной оптимизации на множестве компромиссных решений в активных системах в условиях неточного знания целевых функций их элементов. // Сборник научных трудов «Программные и технические средства медико-биологических и технических систем». Тверской государственный технический университет. Тверь. 1998, С. 4-13.
12. Кузнецов В.Н., Павлов В.А. Информационное управление процессом нечеткого группового выбора в целеустремленных системах // Вестник Тверского Государственного Университета №27 (55), Тверской государственный университет, 2007г. – С. 43-49
13. Кузнецов В.Н., Пашаев Ф.А. Информационное управление финансово-хозяйственной деятельностью предприятий на множестве компромиссных решений. Задачи, модели и методы развития. Монография. – Тверь, ТГТУ, 2003. – 207 с.
14. Кузнецов В.Н., Пашаев Ф.А., Петухов А.А. Согласованная оптимизация стратегических альянсов // Современные сложные системы управления (СССУ/HTCS 2005): Сб.тр.науч.-прак.конф. – Воронеж, ВГАСУ, 2005. – С.160-164
15. Кузнецов В.Н., Пашаев Ф.А., Толоко В.Б. Информационное управление бюджетными целевыми программами. // Сб. трудов научно-практической конференции «Современные сложные системы управления (СССУ2005/HTCS 2005)». Воронеж, ВГАСУ, 2005. – С.168-173.
16. Кузнецов В.Н., Петухов А.А., Соловьева С.А. Программная система согласованной оптимизации стратегических альянсов // Программные продукты и системы: Научный журнал. 2006, №4 – С.24

17. Кузнецов В.Н., Семенов Н.А. Согласованное управление в интеллектуальных информационных системах. // Труды Международных конференций «Искусственные интеллектуальные системы» (IEEE AIS*02) и «Интеллектуальные САПР». Научное издание. М.: Издательство Физико-математической литературы, 2002. – 609 с. – ISBN 5-94052-031-6. – С.169-173.

18. Кузнецов В.Н., Семенов Н.А., Цибарев М.В. Программная система согласованной оптимизации бюджетных целевых программ. // Программные продукты и системы, №2, 2005 г., – С. 37-39

19. Кузнецов В.Н., Соловьева С.А. Экономический анализ хозяйственной деятельности по проектированию автомобильных дорог в организациях с целеустремленными агентами // Вестник Тверского государственного технического университета: Научный журнал. Тверь: ТГТУ, 2006, Вып.8 – С.120-124.

20. Кузнецов Ю.Н., Кузубов В.И., Волощенко А.Б. Математическое программирование. Учеб. пособие для вузов. – М.: «Высш. школа», 1976. – 352 с..

21. Майер. Э. Контроллинг как система мышления и управления. – М: Финансы и статистика, 1993. – 96 с.

22. Мескон М.Х., Альберт М., Хедоури Ф. Основы менеджмента: Пер. с англ. 2-е изд. – М.: Дело, 2004.– 800 с.

23. Миронов В.А., Спрогис Л.Л. Согласованная оптимизация имущественно-земельного комплекса // Современные сложные системы управления (СССУ/HTCS 2004): Сб. тр. науч.-прак. конф. – Тверь, ТГТУ, 2004. – С. 405-407.

24. Михалевич В.С., Волкович В.Л. Вычислительные методы исследования и проектирования сложных систем. – М.: Наука. Главная редакция физико-математической литературы, 1982. – 286 с.

Виноградов Г.П., Кузнецов В.Н. Согласованная оптимизация целевой программы развития промышленного предприятия. В статье рассматривается проблема согласования интересов в организационно-экономических, эффективность функционирования которых в значительной степени определяется поведением производственного персонала. Рассматриваемая в данной статье задача научных исследований отличается от известных решений тем, что в ней рассматривается согласованная оптимизация, как на множестве компромиссных решений, так и на множестве кооперативных решений. Такой подход позволил разработать методы и алгоритмы согласованного управления, заключающиеся в разработке, принятии и реализации проектов ЦПР промышленного предприятия, обеспечивающих на рассматриваемом периоде времени максимальный гарантированный социально – экономический результат при условии удовлетворения интересов собственников, инвесторов, руководства и персонала предприятия. Кооперация является необходимым элементом совместной деятельности. Алгоритмическое предписание решения дискретной задачи согласованной оптимизации строится на основе алгоритма решения многокритериальной дискретной задачи оптимального управления для функции полезности, ядром которого являются градиентные методы математического программирования.

Ключевые слова: согласованная оптимизация, целевая программа развития, активные субъекты, информационное управление.

The problem of coordination of interests in organizational and economic efficiency of functioning of which is largely determined by the behavior of the production staff. Considered in this article the problem of research is different from the prior art in that it deals with consistent optimization as on the set of trade-offs, as well as on a set of cooperative solutions. This approach allowed us to develop methods and algorithms for consistent management, the development, adoption and implementation of projects EDL industrial enterprise, providing the considered time period, the maximum guaranteed socio - economic results provided to satisfy the interests of owners, investors, management and staff of the enterprise. The cooperation is an essential element of joint activity. Algorithmic discrete problem solving prescription coordinated optimization is based on the algorithm for solving multicriteria discrete optimal control problem for the utility function, the core of which are the gradient methods of mathematical programming.

Keywords: consistent optimization, target of the program, active subjects, information management.

Агентно-ориентированное программирование задач распределённого искусственного интеллекта

О.И.Федяев, Ю.В.Зудикова

Донецкий национальный технический университет
fedyaev@donntu.org

Федяев О.И., Зудикова Ю.В. Агентно-ориентированное программирование задач распределённого искусственного интеллекта. В работе рассматривается построение имитационных моделей агентно-ориентированного типа. С помощью методологии Gaia выполнен агентно-ориентированный анализ процессов моделирования. Разработаны прототипы многоагентных систем с помощью инструментальных сред Zeus и MadKit. Исследования показали, что на основе интеллектуальных агентов можно строить качественные имитационные модели неоднородных и динамических систем, которые трудно поддаются описанию традиционными методами.

Ключевые слова: мультиагентная модель, интеллектуальный агент, программный агент, имитационная модель, распределенная система

Введение

Современные объекты автоматизации часто представляют собой распределённые системы, компоненты которых взаимосвязаны, обладают элементами интеллектуальности и поведением высокой степени сложности. К классу таких объектов относятся крупные производственные и энергетические комплексы с автоматизированным управлением, социально-экономические системы, компьютерные сети, предназначенные для распределённой обработки информации и управления производством. Для анализа динамики функционирования таких систем и управления ими традиционные подходы, основанные на жёсткой централизованной архитектуре, не всегда пригодны [1,2]. Это привело к коренной перестройке традиционных организаций и предприятий, к появлению сетевых организационных структур нового типа и виртуальных предприятий [1,3]. Усложнились не только сами объекты, но и решаемые ими задачи, распределённые как в пространстве, так и функционально.

Теория многоагентных систем предлагает новый подход к проектированию и разработке моделей сложных распределённых систем. Она позволяет строить качественно новые модели сложных систем в виде интеллектуальных организаций, представленных автономными искусственными агентами, обладающими возможностями и средствами адаптации к изменениям среды. Преимуществом многоагентного моделирования является возможность естественного описания сложных систем в терминах поведения и функционирования их компонент [5-9]. Эти обстоятельства привели к появлению эволюционной парадигмы развития программного обеспечения на основе автономных и взаимодействующих программных модулей – агентов [1,2,4]. Соответственно абстракция «объект» была заменена ещё более высокой абстракцией – «агент». Таким образом, в программировании наметилась тенденция, когда при распределённом решении задач программисты пишут программы в терминах всё более высокоуровневых абстракций, которые находятся всё дальше от принципов процедурного и объектно-ориентированного программирования, а всё ближе к понятиям, отражающим основные абстракции предметной области [2]. Поэтому с применением и развитием агентных технологий на основе методов искусственного интеллекта связываются самые серьёзные перспективы перехода от пространства данных к пространствам знаний [10].

Однако технология агентно-ориентированного программирования пока ещё не полностью апробирована и является предметом пристального изучения [2]. Есть примеры разработки многоагентных систем, которые не демонстрируют достоинства агентного подхода (задача не адекватна, мало агентов и т. д.). Поэтому целью данной работы является анализ опыта применения агентного подхода к разработке агентно-ориентированных моделей реальных динамических распределённых систем.

Основы агентно-ориентированного проектирования и программирования

Теория многоагентных систем (МАС) возникла на пересечении теории систем, теории коллективного поведения и распределённого искусственного интеллекта [3]. До этого в искусственном интеллекте использовался классический подход, при котором разрабатывался только один интеллектуальный агент –

экспертная система, обладающая всем объёмом знаний для решения трудно формализуемой задачи. Напротив, при создании MAS предполагается, что отдельный агент может иметь лишь частичное представление о задаче и способен решить лишь некоторую её подзадачу. Поэтому для решения сложной проблемы, как правило, требуется взаимодействие агентов, т.е. применение MAS [3].

С некоторыми упрощениями MAS можно описать следующей алгебраической формой [3]:

$$MAS = (A, E, R, ORG),$$

где A – множество агентов; E – среда, в которой находится MAS; R – множество взаимодействий между агентами; ORG – множество базовых организационных структур, соответствующих конкретным функциям (ролям) агентов.

Ключевым элементом MAS является агент, для которого есть много различных определений. Автор отдаёт предпочтение следующему определению. Искусственный агент – это высокоуровневая абстракция для формализации и структурирования сложных сущностей предметной области в виде программно-аппаратного средства, способного автономно действовать в интересах достижения заложенных в него целей и активно взаимодействовать с внешней средой (см. рис.1).

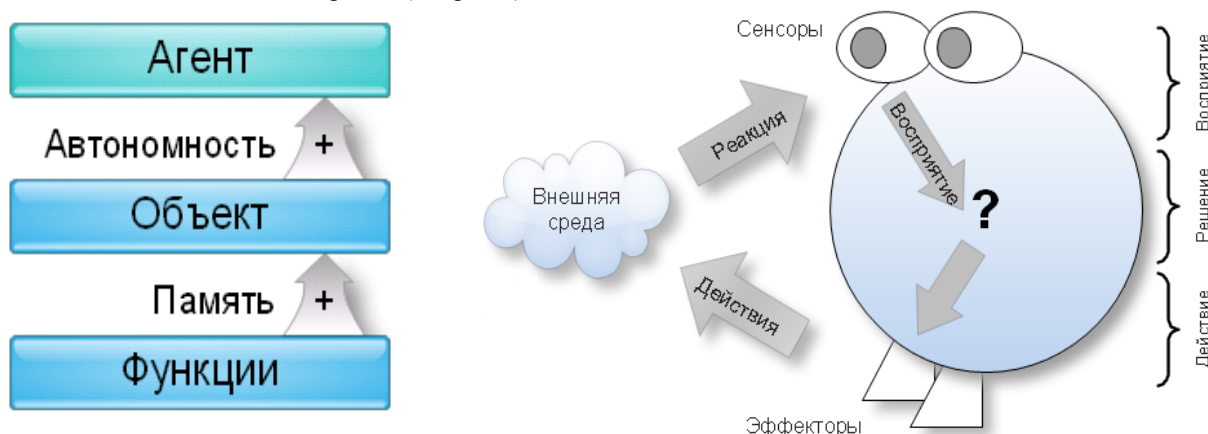


Рисунок 1 – Взаимодействие агента с внешней средой и уровни абстракции понятий агента и объекта

В настоящее время предложено много разных архитектур искусственных агентов [1,2,3]. В архитектурах интеллектуальных агентов (основанные на знаниях), которые представляют наибольший интерес, заложено некоторое подмножество так называемых «ментальных свойств», к которым относятся:

- *убеждения (Beliefs)* – это знания агента о среде, в частности о других агентах; это те знания, которые могут изменяться во времени и становиться неверными, однако агент может не иметь об этом информации и продолжать оставаться в убеждении, что на них можно основывать свои выводы;
- *желания (Desires)* – это состояния, ситуации, достижение которых для агента является желательным;
- *намерения (Intention)* – это постоянная часть знаний агента о себе, среде и других агентах, т.е. та часть, которая не изменяется в процессе его функционирования.

На основе интеллектуальных агентов с BDI-архитектурой можно создавать виртуальные миры, по сложности взаимоотношений сопоставимые с человеческими [2].

Агент хоть и является автономной сущностью, но он функционирует в некоторой среде E, т.е. *окружении*. Внешняя среда агента – это всё, что вне его и при этом доступно для восприятия и изменения. Среда может состоять из других агентов и пассивных объектов (ресурсов).

Взаимодействие между агентами R означает установление двусторонних и многосторонних динамических отношений между субъектами. Это главная черта MAS, которой она отличается от других интеллектуальных систем. К базовым видам взаимодействий между агентами относятся: кооперация (сотрудничество), конкуренция (конфронтация), компромисс (учёт интересов других агентов) и другие [3]. Взаимодействие агентов обусловлено целым рядом причин, важнейшими среди которых являются следующие: совместимость целей (общая цель), отношение к ресурсам, обязательства агентов друг перед другом, опыт агентов, связанный с некоторой проблемной областью. Агенты обмениваются друг с другом информацией посредством переговоров по соответствующему протоколу. Чаще всего общение агентов ведётся на языках KQML и KIF [2].

В настоящее время не существует языка программирования, который в полной мере отвечал бы потребностям технологии многоагентных систем [11]. Разрабатываемые в настоящее время агентные системы используют большой спектр различных базовых языков (более 10 языков), но, к сожалению, ни один из них не может рассматриваться как истинно «агентно-ориентированный». Как правило, расширяют существующие языки программирования, но есть и проекты по разработке новых специализированных агентских языков [2].

Наиболее распространённым языком программирования агентов является язык Java.

Для проектирования MAC созданы методологии и на их основе различные технологии, являющиеся развитием существующих методов инженерии знаний и объектно-ориентированного программирования. К объектно-ориентированному классу относится технология моделирования систем на основе BDI агентов [2]. В сфере совместной работы применяется многоагентный метод Multi-Agent Scenario-Based Method (MASB), основанный на сценарии [12]. Для моделирования промышленных предприятий предложена специализированная агентно-ориентированная методология, объединяющая объектно ориентированный метод Object Oriented Software Engineering (OOSE) и метод моделирования предприятий IDEF и CIMOSA [13]. Методология Gaia применяется для агентно-ориентированного анализа и проектирования MAC, разрабатываемых для предметных областей широкого назначения [14]. Позже методология Gaia была доработана с тем, чтобы сделать её применимой для разработки агентно-ориентированных Internet приложений [15].

Для поддержки процессов проектирования агентов и MAC разработаны специальные инструментальные средства: Jade [16], AgentBuilder [10], Bee-gent [10], Jack [17], MadKit [20], Zeus [18] и др. В следующих разделах рассмотрим примеры агентно-ориентированного программирования реальных задач с распределенным искусственным интеллектом.

Многоагентная модель процесса производства промышленных смесей

В качестве примера рассмотрим технологический процесс производства сухих промышленных смесей [9]. Данный технологический процесс (рис. 2) как объект управления является распределенной системой, в которой взаимодействуют отдельно работающие устройства и персонал. К структурным составляющим процесса производства относятся здания, технологическое оборудование, персонал, сырье, готовая продукция и транспорт. Участники производства распределены территориально и функционируют автономно. Работа персонала заключается в координировании хода производства: управлении работой устройств и принятии решений по обеспечению нормального протекания производственных процессов.

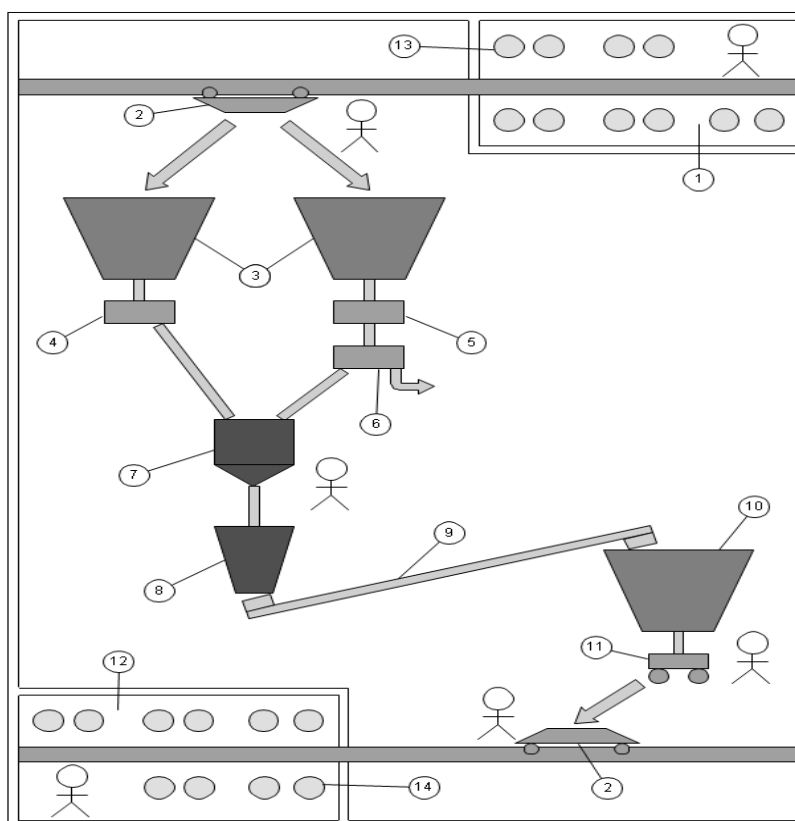


Рисунок 2 – Схема технологического процесса производства промышленных смесей: 1 – склад сырья; 2 – кран-балка; 3 – бункер; 4 – грохот; 5 – дробилка; 6 – сепаратор; 7 – дозатор; 8 – смеситель; 9 – винтовой конвейер; 10 – накопительная емкость; 11 – тележка; 12 – склад готовой продукции; 13 – сырье; 14 – готовые смеси

С позиции системного анализа такой технологический процесс производства является сложной системой

с распределённым интеллектом, которая состоит из автономных, взаимосвязанных, целенаправленно функционирующих элементов и находится во взаимодействии с внешней средой. Элементы системы функционируют во времени и пространстве. Поэтому каждый участник динамического процесса производства описан как подсистема (рис. 4), характеризующаяся входными и выходными данными, параметрами и выполняемыми функциями. Применение традиционных способов моделирования таких сложных объектов автоматизации затруднительно в силу сложности формализации человеко-машинного поведения.

Каждый из агентов наделяется полномочиями того участника производственного процесса, которого он представляет, и обладает поведением, определяемым этими полномочиями. Взаимодействуя с другими агентами и оценивая текущее состояние процесса производства, каждый агент самостоятельно принимает решение о своих дальнейших действиях. Коллективная работа и коммуникация агентов имитируют протекание технологического процесса.

На уровне агентно-ориентированного анализа по методологии Gaia разработаны следующие концептуальные модели производственного процесса: ролей, взаимодействий, агентов, услуг и связей (рис.3) [3, 14].

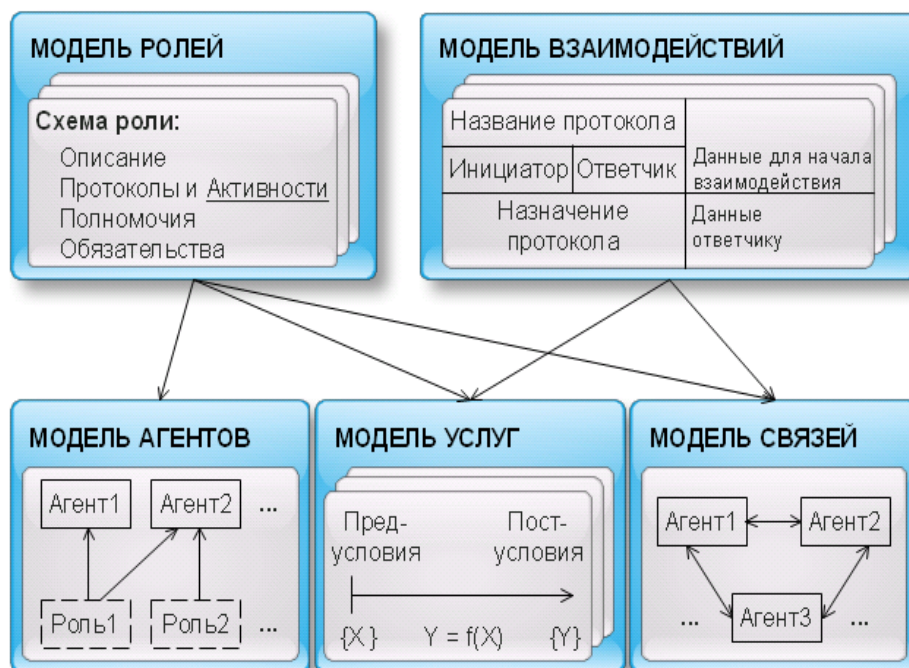


Рисунок 3 – Взаимосвязь абстрактных моделей при агентно-ориентированном проектировании по методологии Gaia

В рамках этой методологии MAC представлена совокупностью взаимодействующих ролей. Роль рассматривается как абстрактное описание свойств и функций объекта. Она задаётся тремя элементами: обязательствами, полномочиями и протоколами. Каждая роль связана с определённым числом протоколов, которые определяют пути взаимодействия её с другими ролями.

В результате проведения агентно-ориентированного анализа процесса производства выделены следующие роли: Кладовщик_сырья, Кладовщик_готовой_продукции, Кран-балка_сырья, Кран-балка_готовой_продукции, Бункер, Грохот, Дробилка, Дозатор, Смеситель, Винтовой_конвейер, Накопительная_ёмкость. Некоторые из ролей (Кран-балки, Дозатора, Накопительной_ёмкости) объединяют в себе функции как устройства (кран-балки, дозатора и накопительной ёмкости соответственно), так и управляющего им работника. Для всех выделенных ролей составлены схемы их описания [9].

Разработанные агентно-ориентированные модели позволили системно перейти от этапа проектирования к этапу программной реализации MAC процесса производства промышленных смесей. В качестве инструментальной среды для разработки MAC был выбран инструментарий Zeus [19], который оперирует схожими с методологией агентно-ориентированного анализа Gaia спецификациями сообщества агентов. Инструментальная среда Zeus помимо средств программной реализации предоставляет свой набор абстракций (моделей) для описания структуры агентов и их поведения. Ключевым вопросом в иерархическом проектировании многоагентной системы является трансформация моделей одного уровня описания в другой. Среда Zeus в отличие от методологии Gaia оперирует другими базовыми понятиями: онтология, факты, агенты, задачи, правила, координация и организация агентов (рис. 4).

Установленная связь концептов среды Zeus и моделей методологии Gaia позволила составить в среде Zeus адекватные спецификации онтологии, структуры и поведения агентов, по которым сгенерирован

программный код онтологии, агентов, базы правил каждого агента, шаблоны задач на языке Java. К полученным компонентам добавлен программный код, реализующий логику выполнения задач агентов, и программный код связанных с агентами внешних программ, предоставляющих интерфейс взаимодействия с пользователем. Для визуализации динамики моделирования был разработан визуализатор процесса, который показывает взаимодействие агентов-устройств и агентов-персонала в ходе моделирования производства промышленных смесей (рис.5). Красный индикатор рядом с графическим изображением агента обозначает, что в данный момент времени агент выполняет задачу, зелёный индикатор – агент завершил выполнение задачи. «Песочные» стрелки показывают направление взаимодействия между агентами. С каждым из агентов связано окно спецификации его характеристик.

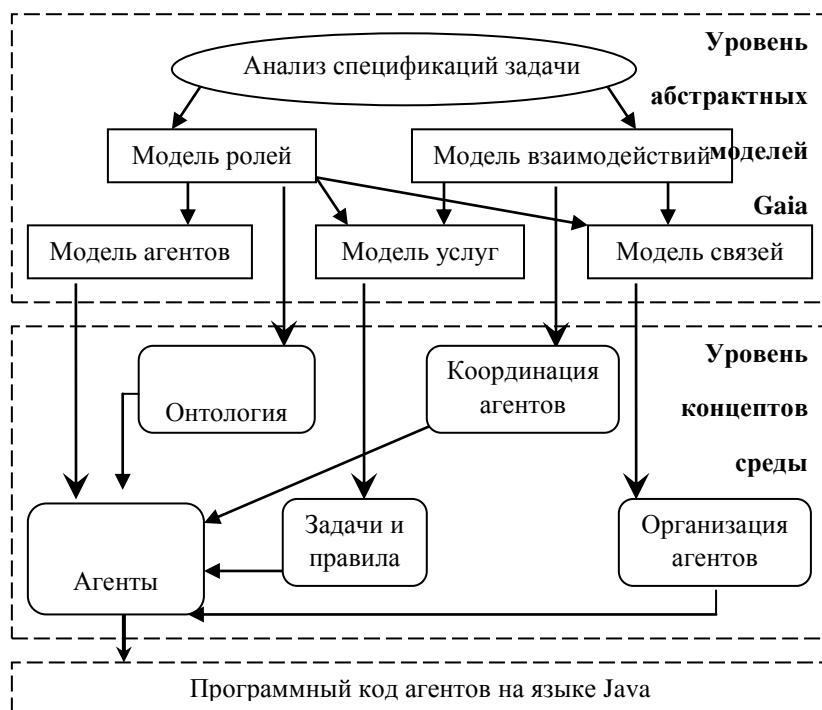


Рисунок 4 – Взаимосвязь моделей методологии Gaia и инструментария Zeus

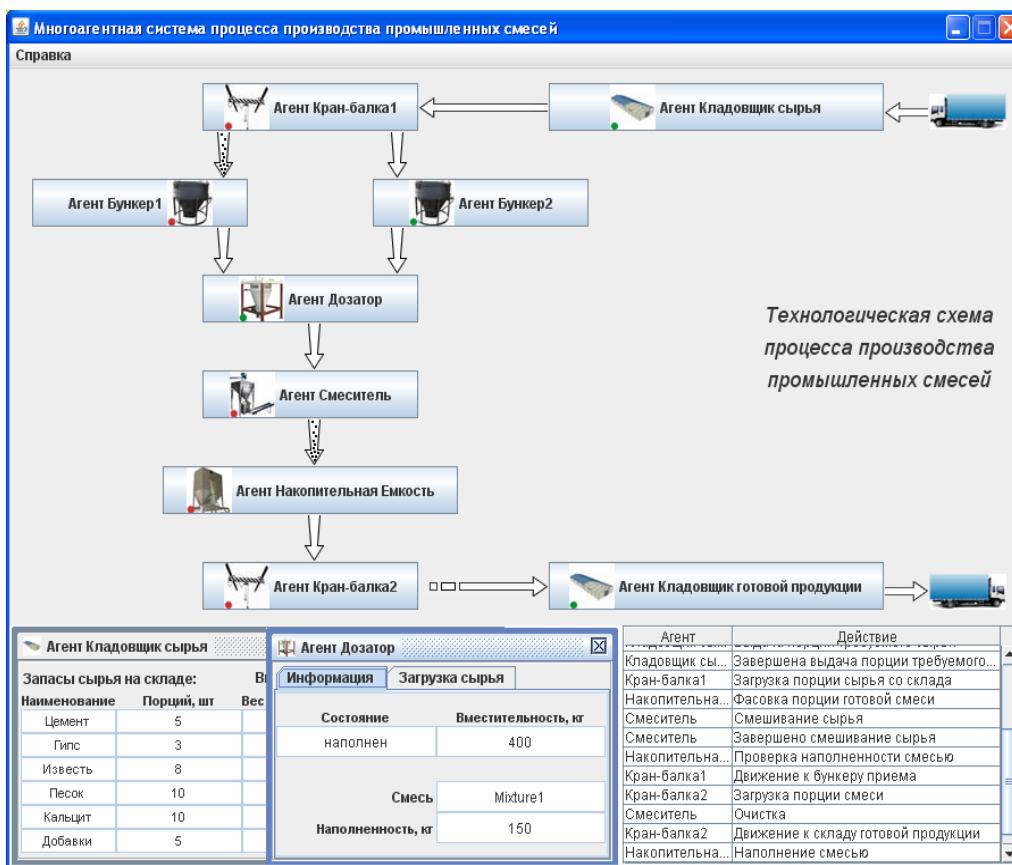


Рисунок 5 – Окно визуализатора процесса моделирования

На основе разработанной многоагентной программной модели можно выполнять анализ, реинжиниринг и управление производством промышленных смесей. В частности были проведены модельные эксперименты по определению рациональной структуры производства [9] и минимизации непроизводительных простоев.

Многоагентная модель прогнозирования трудоустройства выпускников вуза

В данном разделе рассматривается построение другой мультиагентной модели, которая позволит анализировать процесс подготовки молодых специалистов и прогнозировать возможность их трудоустройства. В системе подготовки и трудоустройства молодых специалистов можно выделить следующие элементы: студенты (будущие молодые специалисты), учебное заведение и фирмы (предприятия). Перечисленные элементы системы территориально удалены друг от друга, неоднородны по структуре, их деятельность интеллектуальна и динамична по своей природе. Все вместе они образуют сложную распределённую систему, в рамках которой решаются известные задачи по подготовке кадров.

Эта особенность обуславливает целесообразность применения теории интеллектуальных агентов к разработке имитационной модели для анализа и управления процессами подготовки кадров и их трудоустройства. При этом возникает сложная, но очень актуальная задача формализации поведения людей, успешное решение которой откроет перспективы в создании качественно новых моделей социально-экономических систем [3].

Агентно-ориентированный анализ процесса подготовки и трудоустройства молодых специалистов показал, что его субъекты взаимосвязаны, образуют распределённую, неоднородную и интеллектуальную систему. Поэтому имитационная модель такого процесса была представлена в виде совокупности взаимодействующих интеллектуальных агентов (рис.6).

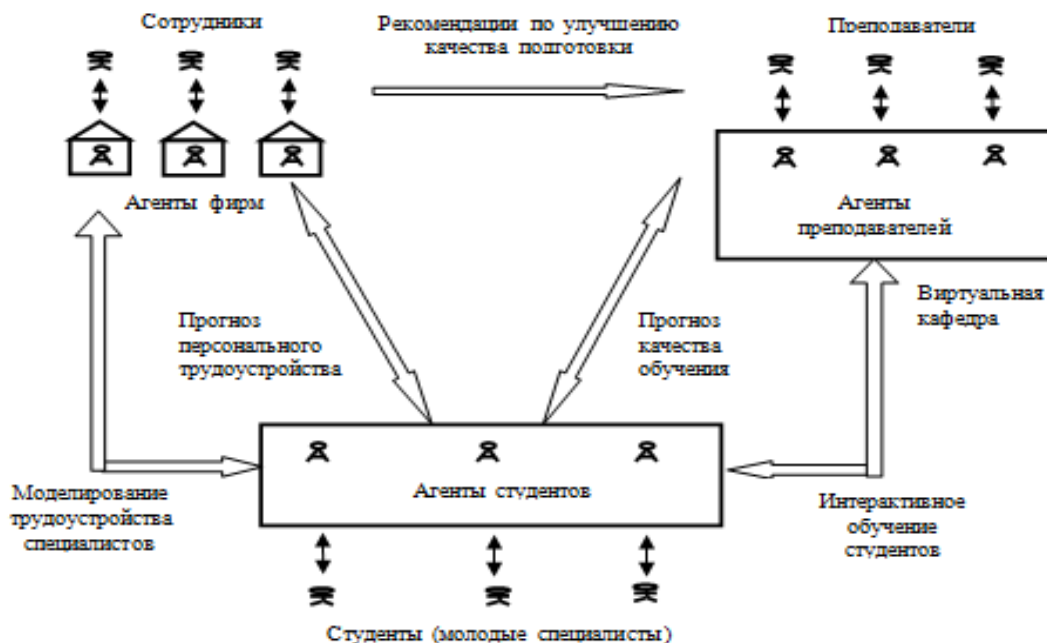


Рисунок 6 - Схема мультиагентной системы моделирования подготовки и трудоустройства специалистов: — Ⓐ — искусственный агент; Ⓒ — реальные сущности системы подготовки кадров

В рамках этой схемы можно решать следующие задачи:

- прогнозирование качества профессионального обучения студентов в зависимости от их личностных характеристик и других факторов;
- оценивание перспективы трудоустройства студентов на этапе их учёбы по прогнозным данным об остаточных знаниях и навыках по профильным дисциплинам;
- моделирование процесса трудоустройства выпускников вуза;
- обеспечение интерактивного обучения студентов, как в компьютерном классе, так и в дистанционном режиме.

Рассмотрим третью задачу, которая решается с помощью имитационной модели процесса трудоустройства выпускников вуза. Как известно, отбор претендентов на работу происходит в процессе собеседования, которое организует каждая фирма (предприятие) с желающими на ней работать. Фирма при собеседовании даёт выпускнику анкету, в которой он заполняет данные о себе и решает тестовые задания, после чего фирма оценивает анкету. Выпускник в то же время получает информацию о требованиях к соискателям, условиях работы и вырабатывает оценку о привлекательности фирмы. После завершения этих процессов фирма и выпускник должны принять общее соглашение о заключении или не заключении трудового договора.

Рассмотрим формальную постановку задачи распределения молодых специалистов на фирмы. Предположим, что в процессе трудоустройства участвуют n выпускников ВУЗа и k фирм, каждая из которых имеет свой профиль работы, квоты и требования к кандидатам на работу.

На первой встрече фирма информирует всех претендентов о предлагаемых условиях работы. По этой информации каждый i -й выпускник для себя формирует показатель социально-экономической привлекательности каждой j -й фирмы

$$P_{ij} = \varphi_{ij}(y_i) ,$$

где $\varphi_{ij}(\cdot)$ — субъективная функция оценки i -м выпускником привлекательности j -й фирмы по социально-экономическим показателям y_i , важным для i -го выпускника.

Если отбросить трудно формализуемые ситуации в собеседовании, которые могут возникнуть, а рассмотреть только наиболее важные аспекты этого процесса, то со своей стороны фирма (в лице её руководства) оценивает профессиональные знания и умения каждого претендента по профилю своей работы и формирует показатель (оценку) «ценности» кандидата для фирмы

$$c_{ij} = f_{ij}(x_j) ,$$

где $f_{ij}(\cdot)$ – субъективная функция оценивания профессионализма претендента по направлению деятельности фирмы; x_j – набор типовых контрольных заданий по направлению деятельности j -й фирмы.

Процедуры реализации функций ϕ_{ij} и f_{ij} позволяют найти следующие матрицы: $\|p_{ij}\|$ – матрица ($n \times k$) значений привлекательности j -й фирмы по оценкам i -го выпускника, ($p_{ij} \geq 0$); $\|c_{ij}\|$ – матрица ($n \times k$) оценок знаний и умений i -го выпускника по результатам собеседования, проводимого j -й фирмой, ($c_{ij} \geq 0$); $\Pi = (1, 2, \dots, n)$ – общий список претендентов на работу, элементами которого являются номера выпускников; S_j – список кандидатов на j -ю фирму (перед собеседованием $S_j := \emptyset$, $j = [1, k]$; \emptyset – пустой список).

В процессе отбора каждый (не распределённый) i -й выпускник определяет для себя самую привлекательную фирму j , куда он пойдёт на собеседование

$$m_i = \arg \max_{1 \leq j \leq k} (p_{ij}) , \quad \forall i \in \Pi , \quad (1)$$

где $m_i \in \{1, 2, \dots, k\}$ – номер фирмы, у которой максимальная привлекательность для i -го выпускника. По m_i формируется новый список кандидатов на j -ю фирму

$$\Delta S_j = \{ i \mid m_i = j \} , \quad i \in \Pi , \quad (2)$$

$$S_j := S_j \oplus \Delta S_j ,$$

где \oplus – операция добавления списка ΔS_j , состоящего из новых претендентов, в уже отобранный ранее список S_j кандидатов на работу в j -ю фирму.

Кроме того, проводится коррекция матрицы $\|p_{ij}\|$: $p_{ij} := 0$, для $\forall j \in \Delta S_j$, $j = [1, k]$. Чтобы отобрать наиболее достойных, j -я фирма ранжирует (сортирует) свой список S_j по убыванию оценок c_{ij} профессиональных знаний кандидатов

$$S_j := \underset{c_{ij}, \forall i \in S_j}{\text{Sort}} (S_j) , \quad j = [1, k] ,$$

где $\text{Sort}(S_j)$ – процедура сортировки элементов списка S_j по убыванию соответствующих каждому элементу значений оценок знаний c_{ij} . Каждая фирма j оставляет в упорядоченном списке S_j только h_j лучших кандидатов

$$S_j := \text{Head}^{+h_j}(S_j) , \quad j = [1, k] ,$$

где $\text{Head}^{+h_j}(S_j)$ – операция выделения первых (головных) h_j элементов в списке S_j ; h_j – выделенная квота приёма молодых специалистов в j -й фирме. Затем проводится коррекция общего списка оставшихся претендентов на работу, т.е. «отбракованные» кандидаты возвращаются в общий список претендентов Π

$$\Pi := (\Pi \ominus \text{Head}^{+h_j}(S_j)) \oplus \text{Tail}^{+h_j}(S_j) , \quad j = [1, k]$$

где $\text{Tail}^{+h_j}(S_j)$ – операция выделения конца (хвоста) списка S_j , начиная с (h_j+1) элемента; \ominus – операция удаления из списка Π элементов списка S_j . Процесс отбора повторяется циклически, начиная с формулы (1).

Таким образом, задача состоит в том, чтобы сформировать для каждой j -й фирмы список S_j принятых на работу выпускников, которые подходят по знаниям фирме и которым нравится работать на ней. Динамический процесс отбора на работу заканчивается тогда, когда все списки претендентов (2) перестанут обновляться, т.е. $\Delta S_j = \emptyset$ для $\forall j \in \{j \mid j = [1, k]\}$.

Исходя из схемы диалога во время собеседования выпускника с представителем фирмы, была построена структура агентно-ориентированной модели (рис.7). При построении искусственных агентов была выбрана нейросетевая архитектура. Интеллектуальными задачами, которые решаются нейросетевым способом, являются: оценивание выпускником условий труда на фирме, решение студентом типовых заданий по профилю фирмы, оценивание фирмой ответов выпускника на тестовые задания. Для передачи профессиональных навыков от молодого специалиста (источника знаний) к нейросетевому программному агенту использовались коммуникативные методы извлечения знаний из реальных студентов-выпускников (это третья задача из списка в разделе 1) и алгоритм настройки нейросети по стратегии «обучение с учителем».

Для извлечения знаний из выпускников были составлены опросные анкеты по каждому разделу программной инженерии (например, «Системы искусственного интеллекта», «Базы данных и т. д.), из которых формировались обучающие множества для нейросетей. В анкету входил набор типовых заданий по каждому из намеченных разделов учебного плана специальности и правильные ответы к ним в виде номеров необходимых знаний и умений из предлагаемого списка для их решения.

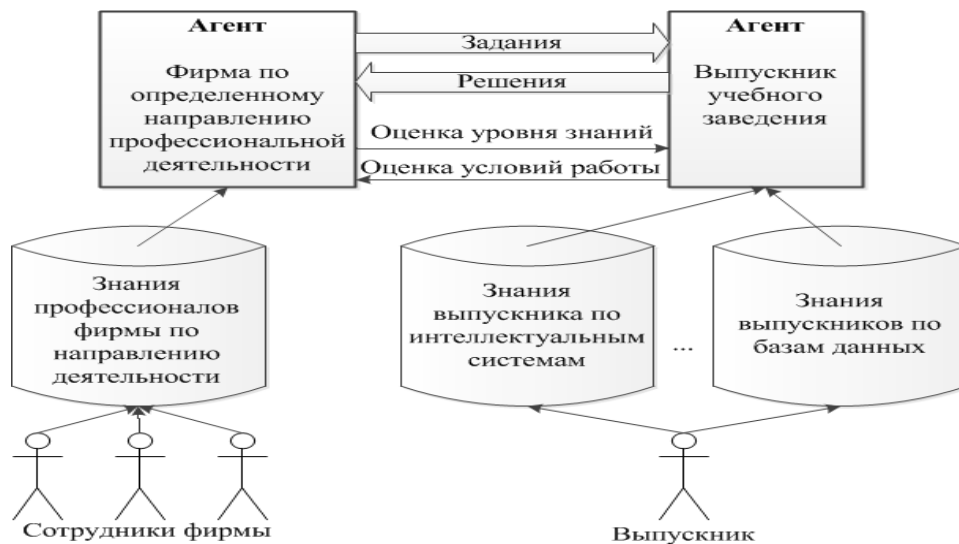


Рисунок 7 - Структура агентно-ориентированной модели собеседования при трудоустройстве выпускника на фирму

Многоагентная система моделирования разработана в инструментальной среде MadKit. Архитектура платформы MadKit основана на AGR (Agent/Group/Role) модели, которая базируется на трёх ключевых понятиях: агент, группа и роль [20].

Рассмотрим один из экспериментов по моделированию процесса трудоустройства выпускников с низким уровнем профессиональной подготовки [21,22]. В эксперименте запланировано участие трёх фирм, у которых имеется определённое количество вакансий: первый агент фирмы (company-1) имеет 2 вакансии, второй (company-2) – 4 и третий (company-3) – 1. В качестве кандидатов на работу в этих фирмах запланировано участие 15 выпускников ВУЗа с низким уровнем знаний по профилю данных фирм. Условия труда, которые предлагают фирмы в данном эксперименте, оценивались выпускниками по 8 показателям: заработная плата, предоставление жилья, форма собственности и т. д.

Динамика процесса визуализируется с помощью специальных окон (рис.8), в которых на каждом шаге моделирования отображается состояние трудоустройства, т.е. сколько и кого уже отобрала каждая фирма, какие выпускники ещё проходят собеседование и т. д.

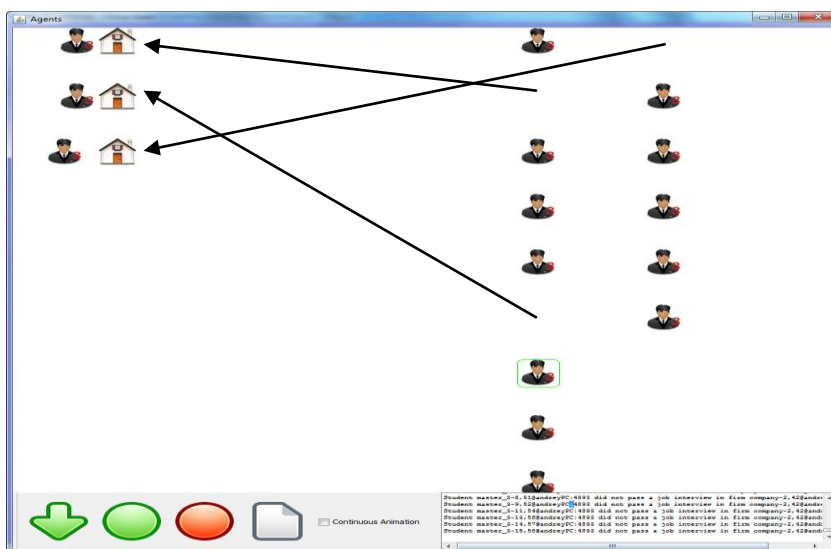


Рисунок 8 - Конечное состояние моделирования трудоустройства

Эксперимент показал, что разработанная агентно-ориентированная модель системы трудоустройства работает правдоподобно и может быть использована для анализа проблем подготовки кадров и прогнозирования перспектив в трудоустройстве студентов старших курсов университета.

Заключение

Исследования, проведенные на примере моделирования процесса производства промышленных смесей, показали, что искусственные агенты – качественно новое средство представления сложных сущностей. Агентно-ориентированная концепция, основанная на использовании знаний в виде продукционных правил и различных протоколов взаимодействий автономных искусственных агентов, порождает новый подход к моделированию неоднородных систем с распределённым интеллектом.

Предложен агентно-ориентированный подход к нейросетевому моделированию трудно формализуемого процесса профессионального обучения студентов, основанный на имитации процесса передачи профессиональных навыков и знаний в зависимости от личностных характеристик студентов.

Разработана многоагентная модель, реалистично отражающая поведение людей и фирм при поиске работы в зависимости от социальных, профессиональных и экономических факторов, делегированных членам искусственных обществ модели системы трудоустройства.

Таким образом, искусственные агенты не только воплощают в себе преимущества традиционных объектов и систем, основанных на знаниях, но и обладают новым качеством, обусловленным возможностью коммуникации и кооперации. Это придаёт новые качества агентным моделям, с помощью которых можно строить модели предметных областей, трудно описываемых традиционными методами.

Литература

11. Городецкий В.И., Грушинский М.С., Хабалов А.В. Многоагентные системы: Обзор // *Новости искусственного интеллекта*. – 1998. - №2. – С.3-19.
12. Болотова Л.С. Системы искусственного интеллекта: модели и технологии, основанные на знаниях: учебник / ФГБОУ ВПО РГУИТТ; ФГАУ ГНИИ ИТТ «Информатика». – М.: Финансы и статистика, 2012. – 664 с..
13. Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002. – 352 с.
14. Тарасов В.Б. Новые стратегии реорганизации и автоматизации предприятий: на пути к интеллектуальным предприятиям, // *Новости искусственного интеллекта*. – 1996. - №4. – С. 40-84.
15. Bussmann S. On the identification of agents in the design of production control systems / S. Bussmann, N. Jennings, M. Wooldridge [Электронный ресурс] – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.15.5827&rep=rep1&type=pdf>.
16. Bussmann S. An agent-based approach to the control of flexible production systems / S. Bussmann, K. Schild [Электронный ресурс] – Режим доступа: <http://www.stefan-bussmann.de/downloads/ETFA2001.pdf>.
17. Parunak H.V.D. Applications of distributed artificial intelligence in industry / H.V.D. Parunak [Электронный ресурс] – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.34.1502&rep=rep1&type=pdf>.
18. Nwana H.S. ZEUS: An advanced tool-kit for engineering distributed multi-agent systems / H.S. Nwana, D.T. Ndumu, L.C. Lee [Электронный ресурс] – Режим доступа: <http://www.agent.ai/doc/upload/200302/nwan98.pdf>.
19. Федяев О.И., Зудикова Ю.В. Иерархическая разработка многоагентной модели процесса производства промышленных смесей. В сб. научн. тр. ДонНТУ. Серия «Информатика, кибернетика и вычислительная техника». – 2011. – Вып. 13(185). - С. 216-223.
20. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем / Т.А.Гаврилова, В.Ф.Хорошевский – СПб.: Питер, 2000. – 384 с.
21. Городецкий В.И. Многоагентные системы: основные свойства и модели координации поведения. // *Информационные технологии и вычислительные системы*. – 1998. - №1. – С. 22-34.
22. Bernard Moulin and Mario Brassard. A scenario-based design method and an environment for the development of multiagent systems. In D. Lukose and C. Zhang, editors, *First Australian Workshop on Distributed Artificial Intelligence, (LNAI volumen 1087)*, pages 216–231. Springer-Verlag: Heidelberg, Germany, 1996.
23. Elisabeth A. Kendall, Margaret T. Malkoun, and Chong Jiang. A methodology for developing agent based systems for enterprise integration. In D. Luckose and Zhang C., editors, *Proceedings of the First Australian Workshop on DAI, Lecture Notes on Artificial Intelligence*. Springer-Verlag: Heidelberg, Germany, 1996.
24. Wooldridge M. J., Jennings N. R. and Kinny D. The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, September 2000.
25. Zambonelli F., Jennings N. R., Omicini A. and Wooldridge M. J. *Coordination of Internet Agents: Models, Technologies and Applications*, chapter 13. Springer-verlag, 2000. Agent-Oriented Software Engineering for Internet Applications.
26. Глибовец Н.Н. Использование JADE (Java Agent Development Environment) для разработки компьютерных систем поддержки дистанционного обучения агентного типа// *Educational Technology&Society* 8(3) 2005 ISSN 1436-4522 pp. 325-345.
27. JACK Intelligent Agents. – Mode of access: <http://www.agent-software.com/products/jack/index.html> .

28. Wooldridge M. The Gaia methodology for agent-oriented analysis and design / M. Wooldridge, N. Jennings, D. Kinny [Электронный ресурс] – Режим доступа: <http://www.csc.liv.ac.uk/~mjw/pubs/jaamas2000b.pdf>
29. Building electronic marketplaces with the ZEUS agent toolkit [Электронный ресурс] / J.C. Collis, L.C. Lee. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.3018>
30. MadKit Development Guide [Электронный ресурс]. – URL: <http://www.madkit.net/documentation/devguide/devguide.html>
31. Федяев О.И., Лукина Ю.Ю., Стропалов А.С. Анализ и прогнозирование процесса трудоустройства молодых специалистов с помощью мультиагентной имитационной модели / Интеллектуальный анализ информации, ИАИ-2013, Международная научная конференция имени Т.А.Таран, Киев, 15-17 мая 2013. – К.: Просвіта, 2013. – С. 47-53.
32. Федяев О.И. Модель системы подготовки и трудоустройства специалистов на основе программных агентов с нейросетевой архитектурой // Пятнадцатая национальная конференции по искусственному интеллекту с международным участием КИИ-2016 (3-7 октября 2016 г., г.Смоленск, Россия).Труды конференции. В 3-х томах. Том 2. – Смоленск: Универсум, 2016. – С. 372-381.

Федяев О.И., Зудикова Ю.В. Агентно-ориентированное программирование задач распределённого искусственного интеллекта. В работе рассматривается построение имитационных моделей агентно-ориентированного типа. С помощью методологии Gaia выполнен агентно-ориентированный анализ процессов моделирования. Разработаны прототипы многоагентных систем с помощью инструментальных сред Zeus и MadKit. Исследования показали, что на основе интеллектуальных агентов можно строить качественные имитационные модели неоднородных и динамических систем, которые трудно поддаются описанию традиционными методами.

Ключевые слова: мультиагентная модель, интеллектуальный агент, программный агент, имитационная модель, распределенная система.

Fedyayev O., Zudikova I. Agent-oriented programming tasks distributed artificial intelligence. The paper deals with the construction of simulation models of agent-oriented type. With the help of Gaia methodology is made of agent-oriented analysis modeling processes. A prototype multi-agent systems using tools and environments Zeus MadKit. Studies have shown that on the basis of intelligent agents can build a high-quality simulation models of inhomogeneous and dynamic systems that are difficult to describe traditional methods.

Keywords: multi-agent model, intelligent agent software agent simulation model, a distributed system.

СЕКЦИЯ «ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ, РАСПОЗНАВАНИЕ И КЛАССИФИКАЦИЯ»

УДК 004.421.4, 004.421.6

Компьютерный тезаурус для метапоисковой ГРИД-системы «Программирование для ОС «Android»

И.А. Коломойцева, Т.В. Завадская
Донецкий национальный технический университет
bolatiger@gmail.com

Коломойцева И.А., Завадская Т.В. Компьютерный тезаурус для метапоисковой ГРИД-системы «Программирование для ОС Android». В данной работе рассматривается решение задачи метапоиска с помощью ГРИД-системы для предметной области «Программирование для ОС Android». Приведена общая схема ГРИД-системы для решения этой задачи. Предложена структура тезауруса для представления естественно-языковой информации, используемой для синтеза запросов метапоисковой системы, и формирования результата её работы.

Ключевые слова: метапоиск, ГРИД-система, тезаурус, Android, объекты, семантические отношения.

Введение

Наше время можно назвать эпохой мобильных устройств. Они вошли в нашу жизнь и стали её неотъемлемой частью. Большинство мобильных устройств работают под управлением операционной системы Android. По данным сайта <http://4pda.ru> их доля на рынке превысила 86%.

Растёт также количество Android-разработчиков. Основную информацию о программировании под Android они берут из Интернета. Учитывая разнообразие мобильных устройств (смартфоны, планшеты, часы, очки и т.п.), их широкие аппаратные возможности (наличие GPS, акселерометров, NFC и т.п.), частоту выхода версий ОС Android (в среднем раз в два-три месяца), разработчикам приложений приходится очень часто обращаться с запросами к всемирной паутине. С другой стороны, существует огромное количество Интернет-ресурсов, содержащих сведения о программировании для ОС Android. Информация на них часто повторяется и, чтобы найти нужные сведения иногда приходится тратить много времени, просматривая ресурсы, перефразируя запросы, обращаясь к разным поисковым машинам.

Облегчить Android-разработчикам поиск сведений об особенностях программирования под Android может метапоисковая система.

Метапоисковая система (метапоисковая машина) — это поисковая система, которая в отличие от классических поисковых машин не имеет собственной базы данных и собственного поискового индекса, а формирует поисковую выдачу за счет смешивания и переранжирования результатов поиска других поисковых систем [2].

Среди разработанных метапоисковых систем следует отметить Web Scout (ищет новости, конференции, аукционы), 1 SECOND, search.da.ru, exactus [1].

Задача метапоиска ресурсозатратная, обладает естественным распараллеливанием. Кроме этого, при реализации автоматических запросов к стандартным поисковым машинам возникают проблемы, связанные с ограничением, установленным этими машинами. Они ограничивают количество автоматических запросов, которые можно выполнить в единицу времени с одного IP-адреса.

Снять это ограничение и увеличить скорость работы метапоисковой системы можно помощью ГРИД-системы.

Направление развития ГРИД-технологии происходит по трём направлениям [2]:

- 1) вычислительные ГРИД;
- 2) семантические ГРИД;
- 3) ГРИД для интенсивной обработки данных.

ГРИД первого направления предназначены для решения вычислительных задач, например, с

использованием численных методов (решение уравнение математической физики) [2].

Семантические ГРИД предназначены для оперирования данными из различных баз данных.

ГРИД третьего направления предназначены для обработки огромных объёмов данных несложными программами, которые можно реализовать на персональном компьютере. Сложность этого направления – доставка данных для обработки и пересылка результатов [2].

ГРИД-системы используются для решения задач прогноза погоды, обработка космических, астрофизических, геофизических данных и для БАК, индексирование новостных БД, в физике высоких энергий, биомедицинских науках, вычислительной химии, ядерного синтеза, финансово-экономических исследований [2, 3, 4, 5].

Задача метапоиска информации о программировании для ОС Android требует обработки большого объёма информации. Поэтому её решение с помощью ГРИД-системы является актуальной задачей.

Целью данной статьи является описание структурной схемы ГРИД-системы, решающей задачу метапоиска информации о программировании для ОС Android, а также структуры текстовой информации, которой она оперирует.

Общая постановка задачи

ГРИД-система для решения задачи метапоиска делится на следующие подсистемы (рисунок 1):

- подсистема интерфейса пользователя, которая обеспечивает ввод запроса от пользователя и вывод результатов поиска;
- подсистема генерации запросов, которая по введённому пользователем генерирует запросы, семантически идентичные исходному;
- подсистема работы брокера (планировщика), которая распределяет запросы между узлами и принимает результат работы;
- подсистема работы узлов Грид-системы, выполняющих поиск;
- подсистема отбора релевантных результатов, которая получает их от брокера и после обработки передаёт подсистеме интерфейса пользователя.

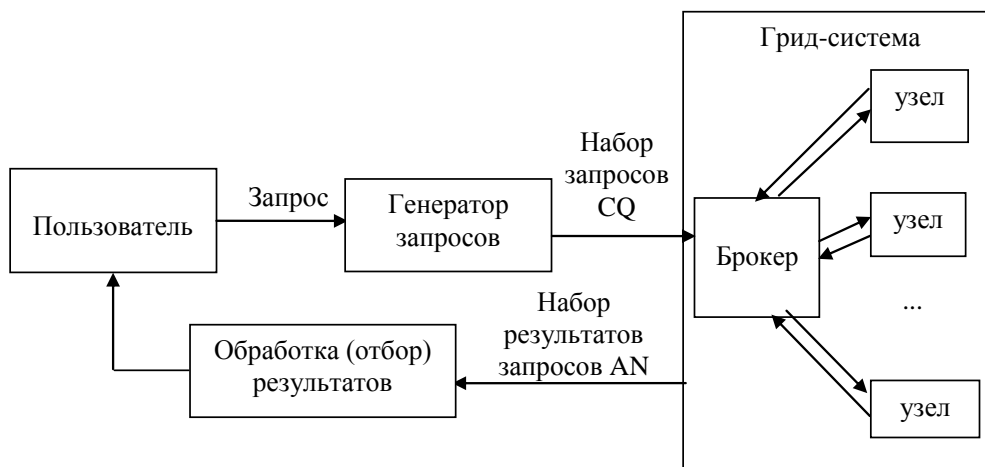


Рисунок 1 – Общая схема ГРИД-системы, решающей задачу метапоиска

Базой для генерации запросов для подсистемы генерации запросов является тезаурус. Тезаурус обычно привязан к некоторой предметной области. Например, в работе [6] предложена объектная модель семантического анализа естественно-языкового текста, используемая для тезауруса запросов в таких предметных областях, как «заболевания» [6] и «лекарства» [7]. Описание тезауруса для предметной области «Программирование для ОС Android» приведено в этой статье дальше.

Узлы ГРИД-системы при решении задачи метапоиска могут выполнять две задачи:

- запускать стандартную поисковую машину с конкретными запросами и получать результат;
- в БД текстов, которая хранится на узлах, осуществлять поиск по конкретному запросу и выдавать список документов, ему удовлетворяющих.

В [8, 9] предлагается структура подсистемы, которая на основе объектной модели из [6], выполняет анализ текстовой БД и отбирает нужные документы. Отобранные документы вместе с результатами работы стандартных поисковых машин, отправляются в подсистему подсистема отбора релевантных результатов. Работа брокерами с узлами, работающими с полнотекстовой БД, не является предметом данной статьи.

Отбор результатов в подсистеме подсистема отбора релевантных результатов может осуществляться на основе двух подходов:

- объектного;
- функционального.

В работах [6-9] описан метод отбора релевантных результатов на основе анализа выдач с помощью тезауруса, построенного посредством объектной модели.

Альтернативой является функциональный подход, представленный в работе [9]. В соответствии с этим подходом выдачи результатов запросов можно представить в виде функций, аргументы которых должны быть сопоставлены с образцами. Однако этот подход имеет более высокую вычислительную сложность.

Алгоритм работы брокера приведен в [10].

Объекты и семантические отношения в тезаурусе «Программирование для ОС Android»

Чтобы использовать естественный язык в качестве основы для построения языка представления знаний, в нем предлагается выделить несколько классов-элементов. Эти классы можно разделить на две категории: семантически значимые объекты предложения и семантические отношения. Объекты еще называют именами [11] и именованными сущностями [12]. Примеры объектов, представленных в ЕЯ-текстах с описанием элементов программирования для ОС Android, приведены в таблице 1.

Таблица 1. Примеры объектов, представленных в ЕЯ-текстах с информацией о программировании для ОС Android

№ п/п	Название объекта	Примеры объектов
1	КОМПОНЕНТ	Деятельность (Activity), служба (Service), приёмник широковещательных намерений (Broadcast Receiver), контент-провайдер (Content Provider)
2	СТАТУС_ПРОЦЕССА	Активный процесс, фоновый процесс, пустой процесс
3	РАЗРЕШЕНИЕ	Permission-tree, permission-group
4	РЕСУРС	Изображение, звук
5	РАЗМЕТКА	Фрейм (FrameLayout), линейная (LinearLayout), табличная (TableLayout), относительная (RelativeLayout)
6	ВИДЖЕТ	Текстовые поля, списки, кнопки, индикаторы
7	ИНДИКАТОР	ProgressBar, RatingBar, SeekBar
8	УВЕДОМЛЕНИЕ	Всплывающее уведомление, уведомление в строке состояния
9	ДИАЛОГОВОЕ_ОКНО	AlertDialog, DatePickerDialog, TimePickerDialog, ProgressDialog
10	МЕНЮ	Меню опций, контекстное меню, подменю
11	ДАТЧИК	SensorManager
12	ПЕРЕДАЧА_ДАННЫХ	Bluetooth, Wi-Fi, NFC
13	ТЕЛЕФОНИЯ	Звонок, SMS
14	ДААННЫЕ	SQLite, геолокационные данные

Часть из приведенных в таблице 1 объектов имеют сложную иерархическую структуру. К таким объектам можно отнести ВИДЖЕТЫ. Из таблицы 1, например, видно, что к ВИДЖЕТАМ относятся ИНДИКАТОРЫ, а ИНДИКАТОРЫ, в свою очередь, делятся на ProgressBar, RatingBar, SeekBar. Имеют свои разновидности СПИСКИ, ТЕКСТОВЫЕ_ПОЛЯ, КНОПКИ и другие ВИДЖЕТЫ.

Объекты связываются между собой с помощью семантических отношений. Выдвинута гипотеза, согласно которой множество отношений, в отличие от множеств объектов (имен), конечно [11]. В [11] выделено около 200 не сводимых к друг другу отношений. В [13] 200 отношений из [11] сведены к семнадцати.

Классы-объекты можно представить в виде древовидной структуры. Особенностью данного дерева является то, что в узлах дерева находятся названия классов, а листьями являются понятия данного класса, что позволяет достаточно четко их определять. Все листья, которые определены в данном классе, являются синонимами.

В естественно-языковых текстах из предметной области «Программирование для ОС Android» можно выделить следующие семантические связи: генеративную, результативную, инструментальную, каузальную, комитативную [13].

Генеративная связь имеет место, когда один компонент обозначает лицо или предмет, принадлежащий некоторой совокупности, категории, обозначаемой вторым компонентом.

Результативная присутствует в тех предложениях, где один компонент выражает следствие действия второго.

Инструментальная означает, что один компонент обозначает орудие действия, обозначаемого другим компонентом.

Каузальная имеет место, когда один компонент обозначает причину появления другого компонента спустя какое-то время.

Комитативная встречается в тех предложениях, когда один компонент обозначает сопровождающее другой компонент действие, сопутствующий предмет, сопровождающее лицо.

Примеры объектов естественно-языковых текстов из предметной области «Программирование для ОС Android», связываемых семантическими отношениями, представлены в таблице 2.

Таблица 2. Семантические отношения и связываемые ими объекты

Семантическая связь	Связываемые объекты
Результативная	СОБЫТИЕ → ДЕЙСТВИЕ
Инструментальная	РАЗРЕШЕНИЕ → ДЕЯТЕЛЬНОСТЬ
Каузальная	УВЕДОМЛЕНИЕ → ДЕЯТЕЛЬНОСТЬ
Комитативная	МЕНЮ → ПОДМЕНЮ

Описание тезауруса предметной области «Программирование для ОС Android»

Знания о предметной области «Программирование для ОС Android» необходимо представить в виде, пригодной для автоматической обработки. Онтологии являются именно такой формой представления знаний.

Онтология – соглашение об общем использовании понятий, которое содержит средства представления предметных знаний и договоренности о методах соображений. Она может рассматриваться как определенное описание взгляда на мир в конкретной сфере интересов, который состоит из набора терминов и правил использования этих терминов, которые ограничивают их значение в рамках конкретной предметной области [11].

Согласно «Современному словарю иностранных слов» тезаурус определяется как полный систематизированный набор данных о какой-либо области знаний, позволяющий человеку или вычислительной машине в ней ориентироваться. Таким образом, тезаурус можно рассматривать как частный случай онтологии.

Существует множество стандартов на формат представления тезаурусов. Основными документами, регламентирующими формат представления тезауруса, являются стандарты ISO 2788-1986 для описания одноязычных тезаурусов и ISO 5964-1985. При разработке тезауруса предметной области «Программирование для ОС Android» будем опираться на первый из перечисленных стандартов.

Стандарт ISO 2788-1986 определяет тезаурус как набор терминов, связанных между собой соответствующими отношениями.

Каждый термин характеризуется комментарием и ссылкой на понятие верхнего уровня.

Основные понятия, представленные в тезаурусе «Программирование для ОС Android»: компонент, статус_процесса, разрешение, ресурс, разметка, виджет, уведомление, диалоговое_окно, меню, датчик, передача_данных, телефония, данные.

Связи, которые представлены в тезаурусе «Программирование для ОС Android»:

– USE – связь с очень близким по смыслу синонимом (термины А и В, представляющие некоторое понятие, обозначают практически одно и то же; например, у одного виджета может быть несколько названий, как минимум английское и русское);

– UF – связь, обратная USE;

– BT – связь термина с понятием высокого уровня (например, термин «текстовое поле» есть уточнение понятия «виджет»);

– NT – связь, обратная BT;

– Res – результативная;

– Ins – инструментальная;

– Caus – каузальная;

– Com – комитативная.

Программно тезаурус представлен таблицами termin, comment и links.

Таблица termin содержит поля, в которых присутствует информация о названии понятия, ссылка на комментарий, ссылка на информацию о связях, идентификатор уровня.

Таблица comment содержит словесное представление комментария.

Таблица links содержит информацию о типе связи, идентификаторы связываемых понятий.

Разработанный тезаурус можно использовать для генерации запросов и формирования окончательного

результата для вывода пользователю, в метапоисковой системе для предметной области «Программирование для ОС Android».

Выводы

В данной работе рассмотрена организация ГРИД-системы, решающей задачу метапоиска для предметной области «Программирование для ОС Android». В частности, выделены основные её структурные элементы.

Разработан тезаурус для предметной области «Программирование для ОС Android». Этот тезаурус будет использован для синтеза запросов метапоисковой системы и для формирования результата её работы, который будет предоставлен пользователю.

Литература

1. Пасько В.П. Энциклопедия ПК. Аппаратура. Программы. Интернет. – Киев, издательская группа BHV; СПб: Питер, 2004. – 800 с.
2. Котляр В.В. Применение Грид-технологий для задач интенсивной обработки данных – Режим доступа: <http://litcey.ru/geografiya/5431/index.html>
3. Castellano M. Biomedical Text Mining Using a Grid Computing Approach / M. Castellano, G. Mastronardi, Decataldo G., Pisciotto L., Tarricone G., Cariello L., Bevilacqua V. – Режим доступа <http://cdn.intechopen.com/pdfs-wm/12928.pdf>.
4. Hughes B. Grid-based Indexing of a Newswire Corpus / B. Hughes, S. Venugopal, R. Buyya. – Режим доступа: <http://www.cloudbus.org/papers/nlp-newswire-grid.pdf>.
5. Li Q. The Future-Oriented Grid-Smart Grid / Q. Li, M. Zhou // Journal of computers, Vol 6. – №1. – 2011. – p.98-105.
6. Коломойцева И.А. Объектная модель семантического анализа естественно-языкового медицинского текста / И.А. Коломойцева // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ – 2007). Выпуск 8 (120). – Донецк: ДонНТУ, 2007. – С. 141-150.
7. Коломойцева И.А. Компьютерный тезаурус для предметной области «ЛЕКАРСТВА» / И.А. Коломойцева // Моделирование и компьютерная графика: Материалы 4-й международной научно-технической конференции, г. Донецк, 5-8 октября 2011 г. — Донецк, ДонНТУ, Министерство науки и образования, молодёжи и спорта, 2011. – с. 161-165.
8. Коломойцева И.А. Объектная модель естественно-языкового медицинского текста на примере системы «ФармАналитик» / И.А. Коломойцева // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ – 2009). Выпуск 10 (153). – Донецк: ДонНТУ, 2009. – С. 303-306.
9. Коломойцева И.А. Функциональная модель медицинского естественно-языкового текста / И.А. Коломойцева // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ – 2008). Выпуск 9 (132). – Донецк: ДонНТУ, 2008. – С. 237-241.
10. Коломойцева И.А., Дацун Н.Н. Алгоритм работы брокера ГРИД-системы для решения задачи метапоиска // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ-2014). – Донецк: ГВУЗ «ДонНТУ», 2014. – Вып. 1 (19) – С. 92-96.
11. Поспелов Д. А. Логико-лингвистические модели в системах управления. М.: Энергоиздат, 1981. 232 с.
12. Хорошевский В.Ф. Оценка систем извлечения информации из текстов на естественном языке: кто виноват, что делать // Труды Десятой национальной конференции по искусственному интеллекту с международным участием (КИИ-2006). М.: Физматлит, 2006. Т. 2. С. 464-478.
13. Осипов Г.С. Приобретение знаний интеллектуальными системами: Основы теории и технологии. М.: Наука. Физматлит, 1997. 112 с.

Kolomoitseva I.A., Zavadskaja T.V. Computer thesaurus for metasearch GRID-system "Programming for Android". This article describes the solution metasearch tasks using GRID-system for domain "Programming for Android". The General scheme of the GRID to solve this problem was developed. The structure of a thesaurus to represent natural language information was developed. This thesaurus used for the synthesis queries of metasearch system, and the formation of the result of work this system.

Keyword metasearch, GRID-system, thesaurus, Android, objects, semantic relations.

Применение метапоиска к решению задач поиска цитат

Серёженко О. А., Коломойцева И. А.
Донецкий национальный технический университет
seryozhenko-94@mail.ru, bolatiger@gmail.com

Серёженко О. А., Коломойцева И. А. Применение метапоиска к решению задач поиска цитат. В данной статье описаны проблема поиска текста в сети Интернет, в частности цитат, инструментарий мета-поисковых систем на основе примеров Exactus и Dogpile. Также рассмотрены и описаны общая схема обработки естественно-языковых текстов, реляционно-ситуационный и латентно-семантический методы анализа текста. Как результат анализа представлена структура мета-поисковой системы цитат.

Ключевые слова: цитата, мета-поисковая система, естественно-языковой текст, синтаксический анализ, семантический анализ.

Введение

На сегодняшний день насущной проблемой для миллионов людей каждый день является проблема поиска информации в Интернете. Последние годы прослеживается явная тенденция к появлению узкоспециализированных сервисов поиска: например, yandex-блоги, yandex-картинки, yandex-новости, google-maps, google-video и т. д. Разделение задачи поиска информации на подзадачи позволяет внедрять новые методы поиска и значительно повышать его эффективность. Но, несмотря на это, на сегодняшний день существует крайне мало решений, обеспечивающих поиск в Интернете именно текстов.

Если текст не такой распространенный, и/или в известный пользователь сетевых библиотеках его найти не удалось, то он вынужден воспользоваться услугами поисковых систем. Пользователь вводит мета-текст в интерфейс поисковой системы (далее ИПС), и получает в ответ несколько сотен или тысяч ссылок, часть из которых ведет на сайты магазинов, в которых можно купить соответствующую книгу, часть этих ссылок ведет в библиографию и/или упоминание, часть этих ссылок просто информационный шум, и, наконец, часть ссылок может ввести к самому тексту. Даже с таким довольно приблизительным разбиением результатов, очевидно, что задача обработки полученного результата возлагается на пользователя. Специализированные же ИПС отсекают значительную часть заведомо нерелевантных результатов, поэтому создание специализированных ИПС под конкретную задачу является более эффективным решением.

Целью данной работы является анализ существующих мета-поисковых систем, методов синтаксического и семантического разборов естественно-языковых текстов. Также на основе этого анализа необходимо определить структурную модель мета-поисковой системы цитат.

Метапоисковая система

Метапоисковая система – это поисковый инструмент, посылающий запрос пользователя одновременно на несколько поисковых систем, каталогов [1].

Принцип работы мета-поисковика заключается в следующем: запрос пользователя преобразуется в запросы, отформатированные синтаксически и логически в конструкции, оптимальные для каждого отдельного, "традиционного" поисковика, т. е. из одного запроса мета-поисковый механизм делает ряд запросов, которые адресуются нескольким "обычным" поискам [2]. Собрав результаты, мета-поисковая система удаляет дублированные ссылки и, в соответствии со своим алгоритмом, объединяет результаты в общем списке.

В рамках одной мета-поисковой системы можно осуществлять поиск информации различного типа. Мета-поисковые системы не предназначены для индексирования и накопления данных, их назначение – чистый поиск и обработка результатов поиска.

Метапоисковые системы позволяют взглянуть на результаты поиска, по ключевым словам, подобрать новые ключевые слова с помощью облаков связанных понятий. Можно однозначно рекомендовать мета-поисковые системы для обзорного поиска. Обзорный поиск полезен при первом подходе к изучению материалов по новой для пользователя теме или же необходим для включения в поле зрения как можно большего числа интернет-источников. Даже одно ключевое слово может в некоторых случаях дать полезный, наводящий результат [3].

Примерами мета-поиска могут служить такие системы, как www.exactus.ru, www.dogpile.com, www.info.com, [vivisimo](http://vivisimo.com) и т. д.

Интеллектуальная поисковая система Exactus

Интеллектуальная поисковая система Exactus позволяет искать документы в сети Интернет. Система Exactus функционирует на высокопроизводительной кластерной установке под управлением операционной системы Unix. Архитектура Exactus построена по модульному расширяемому принципу с возможностью наращивания мощности кластерной установки. Exactus является кроссплатформенной системой, реализованной на языке C++, что позволяет запускать ее на широком спектре Unix-подобных операционных систем [4].

Отличительные особенности системы Exactus:

- полный лингвистический анализ текстов;
- сравнение текстов по семантике;
- нечувствительность к перефразированию – перестановке слов и предложений местами, замене слов и словосочетаний синонимами.

Ознакомиться с интеллектуальной системой Exactus можно по адресу <http://www.exactus.ru>. На рисунке 1 показан пример работы с вышеуказанной интеллектуальной системой.



Рисунок 1 – Пример работы с интеллектуальной системой Exactus

Интеллектуальная поисковая система Dogpile

Метапоисковая система Dogpile использует для поиска несколько средств, среди которых: поисковые машины, веб-каталоги, расширенные средства поиска. Система позволяет искать сайты, изображения, аудио– и видеофайлы, а также проводить поиск в новостях. Есть специальные вкладки «yellow page» («жёлтые страницы») и «white pages» («белые страницы»). В разделе «yellow pages» проводится поиск компаний по их названию, в запросе можно задать отрасль или вид бизнеса, а также город и штат. Раздел «white pages» предназначен для поиска людей по указанным имени, фамилии, городу и штату. Дополнительные возможности, предоставляемые Dogpile: поиск пути между двумя указанными городами, поиск электронного адреса и просмотр карты.

Доступна сортировка полученных результатов по релевантности или поисковому средству. В левой части окна отображается список предыдущих запросов. В Dogpile не предусмотрена кластеризация результатов, вместо этого система формирует список скорректированных запросов, содержащих данное поисковое слово.

Отличительные особенности системы Dogpile:

- «чистый» интерфейс, приятный дизайн;
- наличие функции контроля содержимого;
- наличие 3 варианта фильтрации: отсутствие фильтрации, отбрасывание наиболее нескромных страниц и, наконец, максимальная фильтрация, после которой, по идее, должны остаться только безупречно благопристойные ссылки;
- возможность настройки способа сортировки результатов поиска изображений, новостей и медиа-файлов;
- организация реальных акций в помощь животным.

Ознакомиться с интеллектуальной системой Dogpile можно по адресу <http://www.dogpile.com>. На рисунках 2-3 показаны главная страница вышеуказанной интеллектуальной системой и результат её работы соответственно.



Рисунок 2 – Главная страница интеллектуальной системы Dogpile

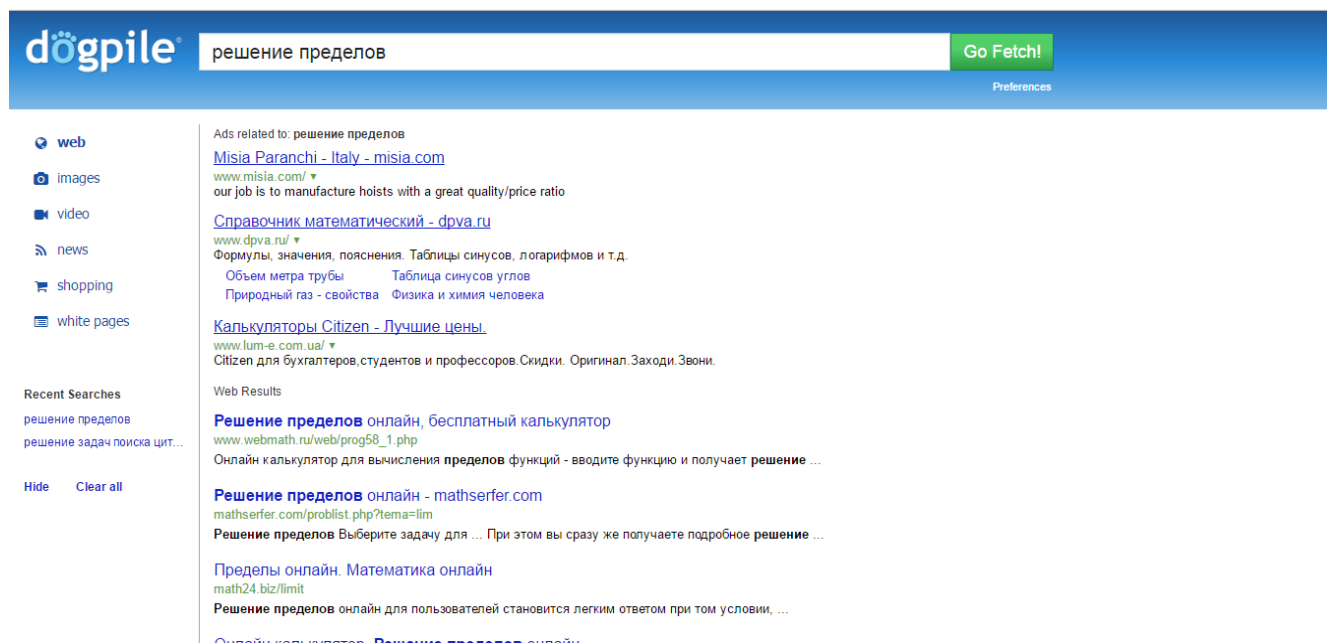


Рисунок 3 – Результат работы интеллектуальной системы Dogpile

Общая схема обработки естественно-языковых текстов

Общая схема обработки текстов инвариантна по отношению к выбору естественного языка. Независимо от того, на каком языке написан исходный текст, его анализ проходит одни и те же стадии. Первые две стадии (разбиение текста на отдельные предложения и на слова) практически одинаковы для большинства естественных языков. Единственное, где могут проявиться специфичные для выбранного языка черты, - это обработка сокращений слов и обработка знаков препинания.

Последующие две стадии (определение характеристик отдельных слов и синтаксический анализ), напротив, сильно зависят от выбранного естественного языка. Последняя стадия (семантический анализ) также мало зависит от выбранного языка, но это проявляется только в общих подходах к проведению анализа.

Семантический анализ основывается на результатах работы предыдущих фаз обработки текста, которые всегда специфичны для конкретного языка. Следовательно, способы представления их результатов тоже могут сильно варьироваться, оказывая большое влияние на реализацию методов семантического анализа.

Результаты анализа, произведенного на ранних стадиях, могут быть многозначны: для выходных параметров указывается не одно, а сразу несколько возможных значений. В таких случаях последующие стадии должны выбирать наиболее вероятные значения результатов ранних стадий анализа и уже на их основе проводить дальнейший анализ текста [5].

На рисунке 4 показана общая схема обработки текстов.

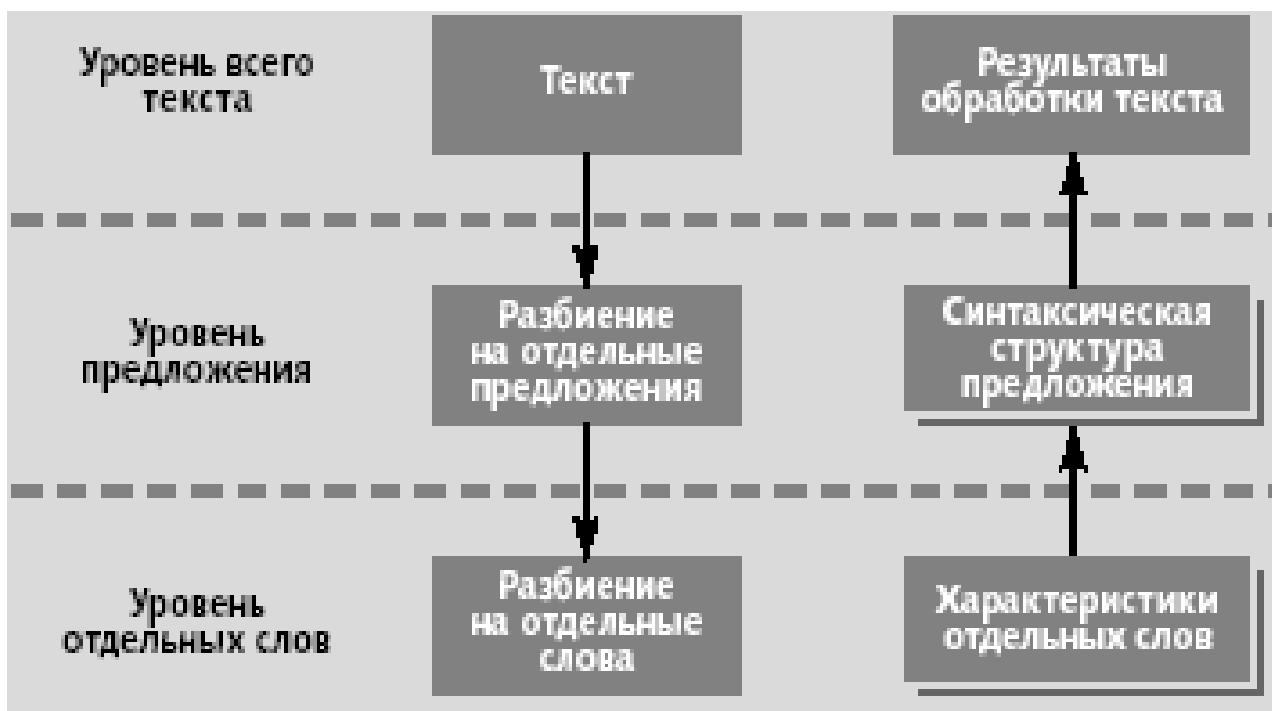


Рисунок 4 – Схема обработки текстов

Реляционно-ситуационный метод анализа текста

Коммуникативная грамматика русского языка отвергает традиционное противопоставление синтаксиса семантике, которое предполагает разделение знаний о законах формирования связной речи на два уровня: знания о форме (синтаксис) и знания о значении (семантика). Главная идея коммуникативной грамматики заключается в том, что синтаксис должен изучать осмысленную речь, а синтаксические правила должны учитывать категориальные значения слов, чтобы иметь возможность определять обобщенный смысл любой синтаксической конструкции – от слова до словосочетания и простого предложения [6, 7].

Формируя и изучая связную речь, синтаксис имеет дело, прежде всего, с осмысленными единицами, несущими свой не индивидуально лексический, а обобщенный, категориальный смысл в конструкциях разной степени сложности. В основе коммуникативной грамматики лежит понятие синтаксем, как минимальной единицы смысла высказывания. Синтаксемой называется минимальная синтактико-семантическая единица языка, несущая обобщенный категориальный смысл и характеризующаяся взаимодействием морфологических, семантических и функциональных признаков.

Синтаксемой называется минимальная синтактико-семантическая единица языка, несущая обобщенный категориальный смысл и характеризующаяся взаимодействием морфологических, семантических и функциональных признаков.

Несмотря на сложность описания, синтаксема является интуитивно понятной конструкцией для любого носителя языка и используется им повсеместно для построения различного рода высказываний. Примеры значений синтаксем:

- аблатив – исходная точка движения (выйти из комнаты);
- агенс – производитель действия (закон подписан президентом);
- адресат – лицо или реже предмет, к которому обращено информативное, донативное или эмотивное действие (обратиться к президенту);
- дестинатив – назначение действия или предмета (выступить в защиту животных; поехать на лечение);
- транзитив – компонент со значением пути движения (Корейский лидер проехал почти по всей России на поезде);
- инструментив – орудие действия (Что написано пером, того не вырубишь топором);

- каузат – объект каузирующего воздействия (способствовать вступлению России в ВТО);
- каузатор – воздействующий фактор (от ученья нравам лишь вред).

Основной задачей реляционно-ситуационного анализа является выявление значений синтаксем и семантических связей между ними. Главную роль здесь играют глаголы, имеющие, как правило, центральное положение в семантической структуре предложения и оказывающие решающее влияние на именные словосочетания и предложения. Сведения о синтаксической сочетаемости каждого глагола с синтаксемами заносятся лингвистами в словарь предикатных слов. Кроме того, в словаре имеются указания на то, как могут быть связаны между собой именные синтаксемы. Набор заносимых в словарь бинарных связей на множестве ролей также является специфичным для каждого типа предикатных слов и определяется априорно [8].

Приведем пример правил для синтаксем родительного падежа: если встречается синтаксема в падеже <родительный> с предлогом <для>, имеющая категориальный класс <личное>, а до неё встречается синтаксема в падеже <именительный>, имеющая категориальный класс <предметное>, то полагается, что первая синтаксема имеет значение <дестинатив - назначение предмета или действия>.

Латентно-семантический метод анализа текста

Латентно-семантический анализ (далее ЛСА) используется для выявления латентных (скрытых) ассоциативно-семантических связей между терминами путем сокращения факторного пространства термины-документы. Терминами могут выступать как слова, так и их комбинации – в идеале: наборы тематически однородных текстов, либо просто любой желателен объемный текст (несколько млн. словоформ), произвольно разбитый на куски, например, абзацы.

Основная идея латентно-семантического анализа состоит в следующем: если в исходном вероятностном пространстве, состоящим из векторов слов (вектором могут выступать предложение, абзац, документ и т.п.), между двумя любыми словами из двух разных векторов может не наблюдаться никакой зависимости, то после некоторого алгебраического преобразования данного векторного пространства эта зависимость может появиться, причем величина этой зависимости будет определять силу ассоциативно-семантической связи между этими двумя словами.

Например, рассмотрим два простых сообщения из разных источников:

1-ый источник реклама: «Этот замечательный ноутбук ХХХ имеет мощный аккумулятор!»

2-ой источник блога: «Кстати, у устройства ХХХ неплохая батарейка».

Поскольку лексика блогов и рекламы не сильно пересекается, то слова «аккумулятор» и «батарейка» получат разный вес, скажем, первое маленький, а второе, наоборот, большой. Тогда эти сообщения можно объединить только на основе названия «ХХХ» (сильный критерий), но подробности про батарею (назовем его слабым критерием) пропадет.

Однако, если мы проведем ЛСА, то веса у «аккумулятора» и «батарейки» выровняются, и эти сообщения можно будет объединить на основе хотя и слабого критерия, но наиболее важного для товара критерия.

Таким образом, ЛСА «стягивает» вместе слова разные по написанию, но близкие по смыслу.

Очевидным недостатком ЛСА является ненормальность вероятностного распределения слов в любом естественном языке. Но эту проблему можно решить сглаживанием выборки (например, применив фонетические слова: распределение становится более «нормальным»).

Другим, менее очевидным недостатком ЛСА (и ему подобных методов) применительно к обработке неструктурированной информации можно отнести «туманность» самого метода и интерпретации результата, не говоря уже о проблеме сбалансированности обучающего текста.

Интерпретируемость результатов ЛСА также затруднительна: человек еще может понять, что за тематику будет содержать текст, полученный в результате анализа, а вот машине не понять тематику без привлечения большого числа хороших и разных тезаурусов.

Таким образом, несмотря на трудоемкость и непрозрачность ЛСА, он может успешно применяться для разного рода задач, где важно поймать семантику сообщения, обобщить или расширить «смыслы» поискового запроса [9].

Применение метапоисковой системы в решении задачи поиска цитат

Поиск цитат — это поиск текста по заданному фрагменту.

Пользователь, задавший такой запрос, вероятнее всего, хочет найти происхождение цитаты — то есть либо увидеть произведение, из которого она взята (в таком случае на найденной странице будет представлен достаточно широкий оригинальный контекст цитаты, что и проверяется анализатором), либо хотя бы узнать автора и название этого произведения.

Рассмотрим эту проблему подробнее, а также введем некоторые ограничения и определения.

1. Под поиском текстов в Интернете понимается ситуация, когда пользователю известно название произведения и/или его автор (имя-фамилия), а в качестве результата пользователь хочет получить полный текст этого произведения в электронном виде.

2. «Текстом» будем считать законченное языковое произведение, характеризующееся наличием автора и названия. «Мета-текстом» будем считать любую непустую комбинацию имени-фамилии автора и названия текста. «Адресом» текста будет считаться веб-адрес, по которому этот текст доступен в Интернете. Если такого адреса не существует, то будем считать адрес этого текста «нулевым».

3. Для примеров будем использовать литературные произведения на русском языке, хотя методы поиска применимы к текстам любого жанра и тематики (технические, публицистические и др.).

Стандартным решением проблемы поиска текстов в Интернете является создание систем, индексирующих найденные в Интернете тексты. По сути дела, данные системы являются базами данных, в которых содержатся ссылки на тексты в Интернете. В таких системах пользователь вводит мета-текст в стандартный поисковый интерфейс и, если данный текст проиндексирован, то пользователь получает адрес текста, по которому данный текст был найден в процессе индексирования. Данные системы обладают, как минимум, одним серьезным недостатком: при таком построении текст привязывается к определенному адресу в Интернете. Но Интернет-ресурсы имеют свойство закрываться, переезжать с одного домена на другой, а расположенные на них документы часто меняют название, удаляются, реорганизуются. Следовательно, проиндексированные адреса в любой момент могут перестать быть актуальными. Второй недостаток подобного метода состоит в охвате представленных адресов. Ручное пополнение баз данных не гарантирует попадание всех адресов текста, автоматическое же индексирование «по адресу» серьезно повышает уровень информационного шума.

Существует иной способ поиска текстов в Интернете, назовем его поиском по цитате или цитатным поиском. Основная идея заключается в том, что ИПС общего назначения предоставляют возможность в качестве запроса указывать целую фразу, и результатом такого запроса будут только те документы, в которых присутствует эта фраза целиком с сохранением порядка слов. Таким образом, если пользователю вместо мета-текста известна цитата из текста, то дальнейший процесс поиска текста сокращается до ввода этой цитаты в кавычках в Google или Yandex, и поисковик либо выдает ссылки непосредственно на текст, либо однозначно сигнализирует о том, что такого текста в Интернете нет. Очевидным образом, мы сталкиваемся с двумя проблемами: предварительное извлечение цитат из текстов и релевантность выдаваемых ссылок. Первая проблема решается построением базы данных цитат, которые будут выдаваться в обмен на мета-текст. Вторая же проблема связана с понятием «целостности текста» (мы можем «попасть» не в целый текст, а в часть текста, например, в случае цитирования текста или ознакомительной страницы) и «минимальной релевантной цитаты» (есть вероятность, что цитата может встретиться в нескольких различных текстах). Обе эти проблемы активно изучаются и решаются авторами. Процесс поиска текстов в Интернете можно разбить на два этапа: пользователь должен по мета-тексту получить цитату, затем по полученной цитате отыскивается полный текст. Таким образом, для эффективного разрешения проблемы поиска текстов в Интернете необходимо создать специализированную ИПС, ядром которой будет база цитат, предварительно извлеченных из текстов. Эта система должна работать в качестве посредника между пользователем и ИПС общего назначения.

Мета-поисковая система поиска цитат будет включать несколько этапов.

1. Пользователь делает запрос с цитатой.
2. Система проводит синтаксический анализ запроса.
3. Результаты синтаксического анализа попадают в блок семантического анализа.
4. На основе результатов синтаксического и семантического анализа, используя словари ассоциаций, синонимов, система генерирует несколько запросов, являющихся вариациями исходного.
5. Система посылает полученные запросы стандартным поисковым системам, например, google, yandex.
6. Система анализирует результат работы поисковиков, выбирая самые подходящие источники цитат, и выводит их на экран пользователю.

На рисунке 5 показана структурная схема алгоритма поиска цитат.

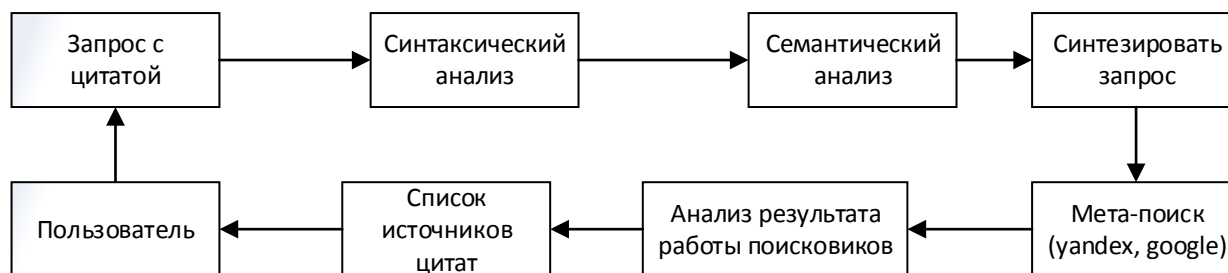


Рисунок 5 – Структурная схема алгоритма поиска цитат

Выводы

В ходе описания статьи были проанализированы:

- проблемы поиска цитат в сети Интернет;
- инструментарии существующих мета-поисковых систем, приведены их характеристики;
- существующие методы обработки естественно-языковых текстов.

Также была составлена схема собственного алгоритма поиска цитат по его фрагментарному заданию.

Литература

1. Мета-поисковые системы [электронный ресурс]. – Режим доступа: http://catalysis.ru/link/index.php?ID=12&SECTION_ID=54
2. Мета-поисковые системы [электронный ресурс]. – Режим доступа: http://www.vsepoisk.ru/2009/07/blog-post_23.html
3. Метапоисковые системы: принципы работы, опыты кластеризации поисковых результатов [электронный ресурс]. – Режим доступа: http://life-prog.ru/2_10898_metapoiskovie-sistemi-printsipi-raboti-opiti-klasterizatsii-poiskovich-rezultatov.html
4. Поисковая система Exactus [электронный ресурс]. – Режим доступа: http://www.isa.ru/index.php?option=com_content&view=article&id=55%3Aexactus&catid=68%3Aon-line-&lang=ru
5. Обработка текстов на естественном языке [электронный ресурс]. – Режим доступа: <http://www.osp.ru/os/2003/12/183694/>
6. Золотова Г.А., Онипенко Н. К., Сидорова М. Ю. Коммуникативная грамматика русского языка. Институт русского языка РАН им. В. В. Виноградова, М. 2004 –544 с.
7. Золотова Г.А. Синтаксический словарь: Репертуар элементарных единиц русского синтаксиса. – М.: Наука, 1988 – 440 с.
8. Г.С. Осипов, И.В. Смирнов, И.А. Тихомиров. Реляционно-ситуационный метод поиска и анализа текстов и его приложения. Отделения нанотехнологий и информационных технологий Российской академии наук, 2008, № 2, 4-6.
9. Латентно-семантический анализ и искусственный интеллект (ЛСА и ИИ) [электронный ресурс]. – Режим доступа: <https://geektimes.ru/post/230075/>

Серёженко О. А., Коломойцева И. А. Применение метапоиска к решению задач поиска цитат. В данной статье описаны проблема поиска текста в сети Интернет, в частности цитат, инструментарий мета-поисковых систем на основе примеров Exactus и Dogpile. Также рассмотрены и описаны общая схема обработки естественно-языковых текстов, реляционно-ситуационный и латентно-семантический методы анализа текста. Как результат анализа представлена структура мета-поисковой системы цитат.

Ключевые слова: цитата, мета-поисковая система, естественно-языковой текст, синтаксический анализ, семантический анализ.

Serozhenko O. A., Kolomoitseva I. A. Application the meta-search to the solution of problem of searching the quotes. This paper describes the problem of searching for text on the Internet, in particular quotes, tools meta-search engine based on Exactus and Dogpile examples. The authors considered and described the scheme of processing of natural language text, relationally-situational and latent-semantic text methods of analysis. The result of analysis is the structure of meta-search engine quotes.

Keywords: quote, meta-search engine, natural-language text, syntactic analysis, semantic analysis.

Методы интеллектуального анализа данных

О.А. Логинов, Э.В. Кулиев, Н.Э. Страхов Н.Э.
Южный федеральный университет
olezhka_94@inbox.ru

Логинов О.А., Кулиев Э.В., Страхов Н.Э. Методы интеллектуального анализа данных. В работе описаны различные методы и представлены решения с помощью существующего на сегодняшний день программного обеспечения. Представлены существующие средства интеллектуального анализа данных. Помимо этого, речь здесь пойдет о нахождении технических решений, относящихся к интеллектуальному анализу данных – к его инструментарию, к структуре и функциям этого инструментария и его организации. Также рассматриваются пути разработки программных систем, могущих повысить эффективность большого количества данных для таких целей, как интеллектуальный анализ, к которому относятся прогнозирование, обучение, классификация.

Ключевые слова: интеллектуальный анализ данных, Ассоциация, Классификация, Кластеризация, Прогнозирование, Последовательные модели

Введение

Стремительное развитие информационных технологий, в частности, прогресс в методах сбора, хранения и обработки данных позволил многим организациям собирать огромные массивы данных, которые необходимо анализировать. Объемы этих данных настолько велики, что возможностей экспертов уже не хватает.

На сегодняшний день интенсивно развивается направление, связанное с интеллектуализацией методов обработки и анализа данных. Интеллектуальные системы анализа данных (ИСАД) призваны минимизировать усилия лица, принимающего решения (ЛПР), в процессе анализа данных, а также в настройке алгоритмов анализа. Многие ИСАД позволяют не только решать классические задачи принятия решения, но и способны выявлять причинно-следственные связи, скрытые закономерности в системе, подвергаемой анализу.

По сути, интеллектуальный анализ данных — это обработка информации и выявление в ней моделей и тенденций, которые помогают принимать решения. Принципы интеллектуального анализа данных известны в течение многих лет, но с появлением больших данных они получили еще более широкое распространение.

Большие данные привели к взрывному росту популярности более широких методов интеллектуального анализа данных, отчасти потому, что информации стало гораздо больше, и она по своей природе и содержанию становится более разнообразной и обширной. При работе с большими наборами данных уже недостаточно относительно простой и прямолинейной статистики. Имея 30 или 40 миллионов подробных записей о покупках, недостаточно знать, что два миллиона из них сделаны в одном и том же месте. Чтобы лучше удовлетворить потребности покупателей, необходимо понять, принадлежат ли эти два миллиона к определенной возрастной группе, и знать их средний заработок.

Инструменты интеллектуального анализа данных

Интеллектуальный анализ данных — это не только используемые инструменты или программное обеспечение баз данных. Интеллектуальный анализ данных можно выполнить с относительно скромными системами баз данных и простыми инструментами, включая создание своих собственных, или с использованием готовых пакетов программного обеспечения. Сложный интеллектуальный анализ данных опирается на прошлый опыт и алгоритмы, определенные с помощью существующего программного обеспечения и пакетов, причем с различными методами ассоциируются разные специализированные инструменты.

Например, IBM SPSS®, который уходит корнями в статистический анализ и опросы, позволяет строить эффективные прогностические модели по прошлым тенденциям и давать точные прогнозы. IBM InfoSphere® Warehouse обеспечивает в одном пакете поиск источников данных, предварительную обработку и интеллектуальный анализ, позволяя извлекать информацию из исходной базы прямо в итоговый отчет.

В последнее время стала возможна работа с очень большими наборами данных и кластерная/крупномасштабная обработка данных, что позволяет делать еще более сложные обобщения

результатов интеллектуального анализа данных по группам и сопоставлениям данных. Сегодня доступен совершенно новый спектр инструментов и систем, включая комбинированные системы хранения и обработки данных.

Можно анализировать самые разные наборы данных, включая традиционные базы данных SQL, необработанные текстовые данные, наборы "ключ/значение" и документальные базы. Кластерные базы данных, такие как Hadoop, Cassandra, CouchDB и Couchbase Server, хранят и предоставляют доступ к данным такими способами, которые не соответствуют традиционной табличной структуре.

В частности, более гибкий формат хранения базы документов придает обработке информации новую направленность и усложняет ее. Базы данных SQL строго регламентируют структуру и жестко придерживаются схемы, что упрощает запросы к ним и анализ данных с известными форматом и структурой.

Документальные базы данных, которые соответствуют стандартной структуре типа JSON, или файлы с некоторой машиночитаемой структурой тоже легко обрабатывать, хотя дело может осложняться разнообразной и переменчивой структурой. Например, в Hadoop, который обрабатывает совершенно "сырые" данные, может быть трудно выявить и извлечь информацию до начала ее обработки и сопоставления.

Основные методы

Несколько основных методов, которые используются для интеллектуального анализа данных, описывают тип анализа и операцию по восстановлению данных. К сожалению, разные компании и решения не всегда используют одни и те же термины, что может усугубить путаницу и кажущуюся сложность.

Рассмотрим некоторые ключевые методы и примеры того, как использовать те или иные инструменты для интеллектуального анализа данных.

Ассоциация

Ассоциация (или отношение), вероятно, наиболее известный, знакомый и простой метод интеллектуального анализа данных. Для выявления моделей делается простое сопоставление двух или более элементов, часто одного и того же типа. Например, отслеживая привычки покупки, можно заметить, что вместе с клубникой обычно покупают сливки.

Создать инструменты интеллектуального анализа данных на базе ассоциаций или отношений нетрудно. Например, в InfoSphere Warehouse есть мастер, который выдает конфигурации информационных потоков для создания ассоциаций, исследуя источник входной информации, базис принятия решений и выходную информацию.

Классификация

Классификацию можно использовать для получения представления о типе покупателей, товаров или объектов, описывая несколько атрибутов для идентификации определенного класса. Например, автомобили легко классифицировать по типу (седан, внедорожник, кабриолет), определив различные атрибуты (количество мест, форма кузова, ведущие колеса). Изучая новый автомобиль, можно отнести его к определенному классу, сравнивая атрибуты с известным определением. Те же принципы можно применить и к покупателям, например, классифицируя их по возрасту и социальной группе.

Кроме того, классификацию можно использовать в качестве входных данных для других методов. Например, для определения классификации можно применять деревья принятия решений. Кластеризация позволяет использовать общие атрибуты различных классификаций в целях выявления кластеров.

Кластеризация

Исследуя один или более атрибутов, или классов, можно сгруппировать отдельные элементы данных вместе, получая структурированное заключение. На простом уровне при кластеризации используется один или несколько атрибутов в качестве основы для определения кластера сходных результатов. Кластеризация полезна при определении различной информации, потому что она коррелируется с другими примерами, так что можно увидеть, где подобию и диапазоны согласуются между собой.

Метод кластеризации работает в обе стороны. Можно предположить, что в определенной точке имеется кластер, а затем использовать свои критерии идентификации, чтобы проверить это. График, изображенный на рисунке 1, демонстрирует наглядный пример. Здесь возраст покупателя сравнивается со стоимостью покупки. Разумно ожидать, что люди в возрасте от двадцати до тридцати лет (до вступления в брак и появления детей), а также в 50-60 лет (когда дети покинули дом) имеют более высокий располагаемый доход.

В этом примере видны два кластера, один в районе \$2000/20-30 лет и другой в районе \$7000-8000/50-65 лет. В данном случае мы выдвинули гипотезу и проверили ее на простом графике, который можно построить с помощью любого подходящего ПО для построения графиков. Для более сложных комбинаций требуется полный аналитический пакет, особенно если нужно автоматически основывать решения на информации о ближайшем соседе.

Такое построение кластеров является упрощенным примером так называемого образа ближайшего соседа. Отдельных покупателей можно различать по их буквальной близости друг к другу на графике. Весьма вероятно, что покупатели из одного и того же кластера разделяют и другие общие атрибуты, и это предположение можно использовать для поиска, классификации и других видов анализа членов набора данных.

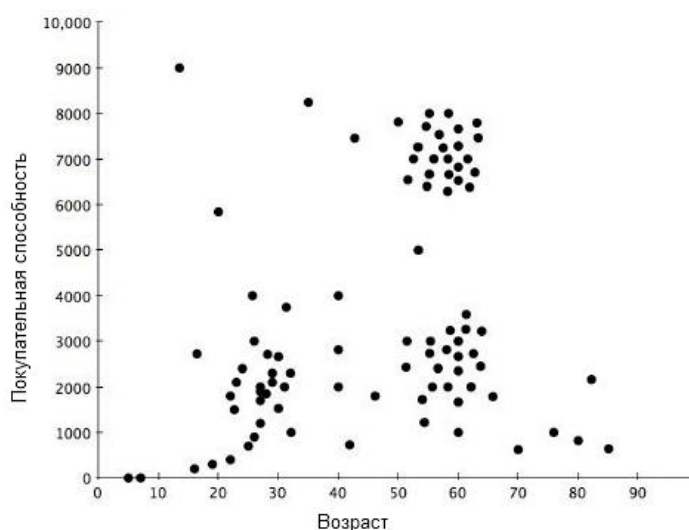


Рисунок 9 – Кластеризация

Метод кластеризации можно применить и в обратную сторону: учитывая определенные входные атрибуты, выявлять различные артефакты. Например, недавнее исследование четырехзначных PIN-кодов выявило кластеры чисел в диапазонах 1-12 и 1-31 для первой и второй пар. Изобразив эти пары на графике, можно увидеть кластеры, связанные с датами (дни рождения, юбилей).

Прогнозирование

Прогнозирование — это широкая тема, которая простирается от предсказания отказов компонентов оборудования до выявления мошенничества и даже прогнозирования прибыли компании. В сочетании с другими методами интеллектуального анализа данных прогнозирование предполагает анализ тенденций, классификацию, сопоставление с моделью и отношения. Анализируя прошлые события или экземпляры, можно предсказывать будущее.

Например, используя данные по авторизации кредитных карт, можно объединить анализ дерева решений прошлых транзакций человека с классификацией и сопоставлением с историческими моделями в целях выявления мошеннических транзакций. Если покупка авиабилетов в США совпадает с транзакциями в США, то вполне вероятно, что эти транзакции подлинны.

Последовательные модели

Последовательные модели, которые часто используются для анализа долгосрочных данных, — полезный метод выявления тенденций, или регулярных повторений подобных событий. Например, по данным о покупателях можно определить, что в разное время года они покупают определенные наборы продуктов. По этой информации приложение прогнозирования покупательской корзины, основываясь на частоте и истории покупок, может автоматически предположить, что в корзину будут добавлены те или иные продукты.

Деревья решений

Дерево решений, связанное с большинством других методов (главным образом, классификации и прогнозирования), можно использовать либо в рамках критериев отбора, либо для поддержки выбора определенных данных в рамках общей структуры. Дерево решений начинают с простого вопроса, который имеет два ответа (иногда больше). Каждый ответ приводит к следующему вопросу, помогая классифицировать и идентифицировать данные или делать прогнозы.

Деревья решений часто используются с системами классификации информации о свойствах и с системами прогнозирования, где различные прогнозы могут основываться на прошлом историческом опыте, который помогает построить структуру дерева решений и получить результат.

Получение и подготовка данных

Сам интеллектуальный анализ данных опирается на построение подходящей модели и структуры, которые можно использовать для обработки, выявления и создания необходимой информации. Независимо от

формы и структуры источника данных, информация структурируется и организуется в соответствии с форматом, который позволяет выполнять интеллектуальный анализ данных с максимально эффективной моделью.

Подумайте о комбинировании бизнес-требований по интеллектуальному анализу данных с выявлением существующих переменных (покупатель, стоимость, страна) и созданием новых переменных, которые можно использовать для анализа данных на подготовительном этапе.

Аналитические переменные для данных, полученных из множества различных источников, можно составить в единую, определенную структуру (например, создать класс покупателей определенных уровней и возрастов или класс ошибок определенного типа).

В зависимости от источника данных важно выбрать правильный способ построения и преобразования этой информации, каким бы ни был метод окончательного анализа данных. Этот шаг также ведет к более сложному процессу выявления, сбора, упрощения или расширения информации в соответствии с входными данными (рисунок 2).



Рисунок 10 - Подготовка данных

Опора на SQL

Наиболее простым из всех подходов часто служит опора на базы данных SQL. SQL (и соответствующая структура таблицы) хорошо понятен, но структуру и формат информации нельзя игнорировать полностью. Например, при изучении поведения пользователей по данным о продажах в модели данных SQL (и интеллектуального анализа данных в целом) существуют два основных формата, которые можно использовать: транзакционный и поведенческо - демографический.

При работе с InfoSphere Warehouse создание поведенческо - демографической модели в целях анализа данных о покупателях для понимания моделей их поведения предусматривает использование исходных данных SQL, основанных на информации о транзакциях, и известных параметров покупателей с организацией этой информации в заранее определенную табличную структуру. Затем InfoSphere Warehouse может использовать эту информацию для интеллектуального анализа данных методом кластеризации и классификации с целью получения нужного результата. Демографические данные о покупателях и данные о транзакциях можно скомбинировать, а затем преобразовать в формат, который допускает анализ определенных данных.

Документальные базы данных и MapReduce

Обработка с помощью функции MapReduce многих современных документальных и NoSQL баз данных, таких как Hadoop, нацелена на очень большие наборы данных и информацию, которая не всегда соответствует табличному формату. При работе с программным обеспечением интеллектуального анализа данных эта система может принести пользу — и вызвать проблемы.

Основная проблема данных на основе документов — это неструктурированный формат, который может потребовать дополнительной обработки. Много различных записей могут содержать аналогичные данные. Сбор и согласование этой информации в целях упрощения ее обработки зависит от этапов подготовки и применения MapReduce.

В системе, основанной на MapReduce, на этапе преобразования исходные данные нормализуются — приводятся к стандартной форме. Этот шаг может быть относительно простым (определение ключевых полей или точек данных) или сложным (анализ и обработка информации для создания выборки данных). В процессе преобразования данные приводятся к стандартизированному формату, который можно использовать в качестве базы.

Сокращение — это суммирование или количественная оценка информации с последующим выводом этой информации в стандартизованную структуру, основанную на итогах, суммах, статистике или других результатах анализа, выбранных для вывода данных.

Запросы к этим данным часто бывают сложными — даже при использовании специализированных инструментов. Идеальный подход к интеллектуальному анализу данных заключается в использовании этапа MapReduce в рамках подготовки данных.

Например, при выполнении интеллектуального анализа данных методом ассоциации или кластеризации на первом этапе лучше всего построить подходящую статистическую модель, которую впоследствии можно будет применять для выявления и извлечения необходимой информации. Используйте этап MapReduce для извлечения и вычисления этой статистической информации с ее последующим вводом в остальную часть процесса интеллектуального анализа данных, что ведет к созданию структуры, представленной на рисунке 3.

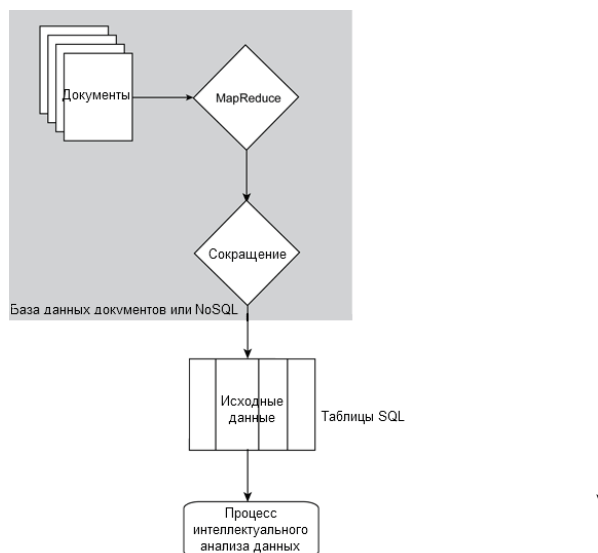


Рисунок 11 - Структура MapReduce

В предыдущем примере мы выполнили обработку (в данном случае посредством MapReduce) исходных данных в документальной базе данных и преобразовали ее в табличный формат в базе данных SQL для целей интеллектуального анализа данных.

Для работы с этой сложной и даже неструктурированной информацией может потребоваться более тщательная подготовка и обработка. Существуют сложные типы и структуры данных, которые нельзя обработать и подготовить в нужном вам виде за один шаг. В этом случае можно направить выход MapReduce либо для последовательного преобразования и получения необходимой структуры данных, как показано на рисунке 4, либо для индивидуального изготовления нескольких таблиц выходных данных.

Например, за один проход можно взять исходную информацию из документальной базы данных и выполнить операцию MapReduce для получения краткого обзора этой информации по датам. Хорошим примером последовательного процесса является регенерирование информации и комбинирование результатов с матрицей решений (создается на втором этапе обработки MapReduce) с последующим дополнительным упрощением в последовательную структуру. На этапе обработки MapReduce требуется, чтобы весь набор данных поддерживал отдельные шаги обработки данных.

Независимо от исходных данных, многие инструменты могут использовать неструктурированные файлы, CSV или другие источники данных. Например, InfoSphere Warehouse в дополнение к прямой связи с хранилищем данных DB2 может анализировать неструктурированные файлы.

Заключение

Интеллектуальный анализ данных — это не только выполнение некоторых сложных запросов к данным, хранящимся в базе данных. Независимо от того, используете ли вы SQL, базы данных на основе документов, такие как Hadoop, или простые неструктурированные файлы, необходимо работать с данными, форматировать или реструктурировать их. Требуется определить формат информации, на котором будет основываться ваш метод и анализ. Затем, когда информация находится в нужном формате, можно применять различные методы (по отдельности или в совокупности), не зависящие от требуемой базовой структуры данных или набора данных.

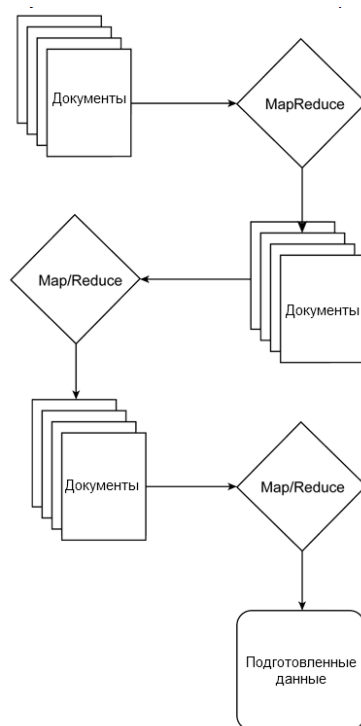


Рисунок 12 - Последовательная цепочка вывода результатов обработки MapReduce

Литература

1. Интеллектуальные информационные системы: Интеллектуальная информационная технология. Экспертные системы: Учеб.пособие / Д. В. Гаскаров, Д. В. Сикунер, В. В. Фомин, И. К. Фомина. СПб: СПГУВК, 2004. 362 с.
2. Искусственный интеллект: В 3 кн. Кн. 1. Системы общения и экспертные системы: Справочник / Под ред. Э. В. Попова. М.: Радио и связь, 1990. 464 с.
3. Марманис Х., Бабенко Д. Алгоритмы интеллектуального Интернета: Пер. с англ. СПб: Символ-Плюс, 2011. 480 с.
4. Фомин В. В. Автоматизация логического моделирования программного обеспечения с применением формального аппарата семиотических систем. СПб: Энергоатомиздат., Санкт-Петербургское отделение, 2000. 250 с.
5. Сахаров А. А. Концепция построения и реализации информационных систем, ориентированных на анализ данных // СУБД. - 1996. - № 4. - С. 55-70.

Loginov O.A, Kuliev E.V., Strahov N.E. Methods of data mining. The paper describes various methods and solutions are presented using existing to date software. Presents the existing means of data mining. In addition, we are going to find technical solutions related to data mining - to his toolbox, to the structure and functions of the instruments and its organization. It also discusses ways to develop software systems that could improve the efficiency of large amounts of data for purposes such as intellectual analysis, which includes forecasting, training, classification.

Keywords: data mining, association, classification, clustering, forecasting, model, serial

Разработка подсистемы интеллектуального анализа данных учебного процесса

И.И. Казмина

Южный федеральный университет, РФ
kazmina.i.i@mail.ru

Казмина И.И. Разработка подсистемы интеллектуального анализа данных учебного процесса. В работе представлена модификация Априорного алгоритма для интеллектуального анализа данных учебного процесса, описаны проектирование и интерфейс подсистемы интеллектуального анализа данных учебного процесса. Предложенная модификация Априорного алгоритма использует основные показатели, участвующие в формировании правил взаимосвязей в оригинальном алгоритме, а также вводит показатель информативности для характеристики выявленных правил с точки зрения информационных потребностей пользователя подсистемы. В рамках проектирования подсистемы интеллектуального анализа данных учебного процесса представлены ее структура, функциональная модель, схема БД, а также алгоритм работы пользователя с подсистемой. Представлен интерфейс разработанной подсистемы.

Ключевые слова: интеллектуальный анализ данных, Априорный алгоритм, модификация Априорного алгоритма, подсистема интеллектуального анализа, учебный процесс, проектирование.

Введение

Интеллектуализация является одним из перспективных направлений развития современных информационных систем (ИС). Интеллектуальные технологии предоставляют инструменты для решения сложных задач, решение которых традиционными методами может быть затруднительно или невозможно. В число таких задач входит интеллектуальный анализ данных (ИАД), распознавание образов и др. Интеллектуализация ИС перспективна в различных областях человеческой деятельности, в том числе в образовании. Поддержка процессов индивидуализации и интеллектуализации обучающих систем, программ и технологий является важнейшей особенностью современных компьютерных технологий обучения [1-3]. Для поддержки эффективности обучения в интеллектуальных обучающих системах могут использоваться знания о предмете обучения, о стратегиях и методах обучения, знания об обучаемом, которые выделяются эксплицитно и реализуются с помощью различных методов и технологий искусственного интеллекта [1,4-6] и др. Поводом для интеллектуального взаимодействия могут быть практические задачи, учебные проекты, разрабатываемые территориально-разобщенными группами обучающихся, сложные решения, принимаемые в условиях неопределенности, вызванной недостаточной формализацией области знаний при проведении учебных и научных исследований, решения, принимаемые в условиях частично не совпадающих интересов и целей обучения [7] и др.

Использование ИАД позволяет выявлять новые сведения и закономерности в больших объемах данных, а также осуществлять прогнозирование. В работе предлагается использовать ИАД для анализа данных учебного процесса. Целью такого использования является выявление зависимостей между успеваемостью обучаемых и различными характеристиками учебного процесса. Данные для анализа будут включать успеваемость студентов за определенный рассматриваемый промежуток времени и особенности учебного процесса, характеризующие его с точек зрения используемых учебно-методических материалов, технологий обучения и контроля, характеристик преподавателя и др. В связи с этим результаты ИАД, представляющие собой зависимости в данных, будут индивидуальны для каждой процедуры анализа.

Исследование возможностей использования ИАД для анализа учебного процесса осуществлялось автором в работах [8-9].

Целью данной работы является разработка подсистемы ИАД учебного процесса. Для достижения данной цели решены следующие задачи: разработан алгоритм, используемый для ИАД учебного процесса, осуществлено проектирование подсистемы, а также разработан ее интерфейс.

Модификация Априорного алгоритма ИАД учебного процесса

В основе подсистемы ИАД учебного процесса лежит алгоритм выявления взаимосвязей в данных учебного процесса с учетом информационных потребностей пользователя. Данный алгоритм является модификацией известного Априорного алгоритма ИАД [10]. Априорный алгоритм ИАД позволяет выявлять закономерности в больших объемах данных, за счет выявления групп элементов, часто встречающихся вместе, и формирования на их основе ассоциативных правил. Априорный алгоритм включает показатель поддержки (support), характеризующий частоту встречаемости групп элементов, и показатель достоверности (confidence), характеризующий точность сформированных правил. Правила формируются из наборов элементов, которые имеют достаточно высокий показатель поддержки. Для каждого сформированного правила рассчитывается показатель достоверности.

Предложенная модификация Априорного алгоритма, кроме вышеуказанных, также использует показатель информативности, характеризующий актуальность сформированных правил. Показатель информативности рассчитывается на основании информационных потребностей и предпочтений пользователя подсистемы.

Модифицированный Априорный алгоритм для анализа данных учебного процесса включает следующие основные шаги:

1. Задание параметров алгоритма (пороговые значения поддержки и достоверности, параметры, отражающие информационные потребности пользователя системы).
2. Выявление повторяющихся наборов элементов в данных.
3. Расчет показателя поддержки для каждого выявленного набора элемента.
4. Отсевание наборов элементов, показатель поддержки (S_i) которых меньше заданного порогового значения (S_{min}), т.е. тех, для которых выполняется условие:

$$S_i < S_{min}, \quad (1)$$

5. Формулировка правил, отражающих зависимость успеваемости от характеристик учебного процесса, из тех наборов элементов, которые прошли отсев по условию (1).

6. Расчет показателя достоверности для каждого сформулированного правила.

7. Отсевание правил, показатель достоверности (C_i) которых меньше заданного порогового значения (C_{min}), т.е. тех, для которых выполняется условие:

$$C_i < C_{min}, \quad (2)$$

8. Расчет показателя информативности для правил, прошедших отбор по условию (2), в соответствии с информационными предпочтениями пользователя системы, заданными с помощью соответствующих параметров.

9. Вывод правил и их показателей достоверности и информативности пользователю системы.

Так как информационные потребности и предпочтения пользователя подсистемы зависят от целей и ожидаемых результатов в каждом отдельном случае ИАД, то они фиксируются перед процедурой запуска алгоритма ИАД по средствам использования специальных параметров. Эти параметры отражают степень значимости тех или иных данных для пользователя, которая учитывается при расчете информативности правил.

Приведенная модификация Априорного алгоритма позволяет выявлять взаимосвязи между успеваемостью студентов и характеристиками учебного процесса, с учетом информационных потребностей пользователя. Полученные правила могут быть использованы при корректировке учебного процесса с целью повышения его эффективности.

Проектирование подсистемы ИАД учебного процесса

Процесс проектирование ИС включает ряд этапов, на которых осуществляется разработка моделей будущей системы, характеризующих ее с различных точек зрения. Состав моделей должен быть достаточен для описания системы с необходимой степенью детализации, достаточной для дальнейшей программной реализации системы. В рамках проектирования подсистемы ИАД учебного процесса были разработаны структура, функциональная модель, схема БД и алгоритм работы пользователя подсистемы.

Структура подсистемы ИАД учебного процесса (см. рис.1) отражает ее элементы и связи между ними. В составе подсистемы выделены Модуль ввода данных, Модуль выполнения алгоритма ИАД, Модуль вывода результатов и БД. Параметры алгоритма и данные для анализа вводятся в подсистему из внешней среды, а именно пользователем подсистемы.

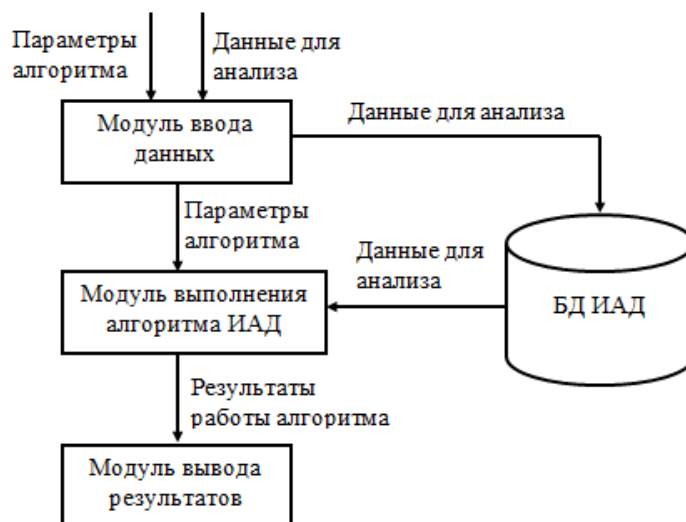


Рисунок 1 – Структура подсистемы ИАД учебного процесса

Функциональная модель подсистемы ИАД (см. рис.2) отражает основные функциональные процессы, БД, пользователя и протекающие между ними потоки данных. Данная модель построена с использованием методологии DFD (Data Flow Diagrams). Внешней сущностью (источником данных) является преподаватель, который порождает информационные потоки, входящие в подсистему. К таким информационным потокам относятся данные учебного процесса и параметры алгоритма. Функциональные процессы преобразуют входящую в них информацию и порождают новые потоки. После выполнения всех функциональных процессов, предусмотренных подсистемой, итоговая информация передается к ее потребителю, которым выступает преподаватель. В данном случае преподавателю передаются результаты работы подсистемы, которые представляют собой выявленные в исходных данных правила, отражающие зависимость успеваемости от различных характеристик учебного процесса.

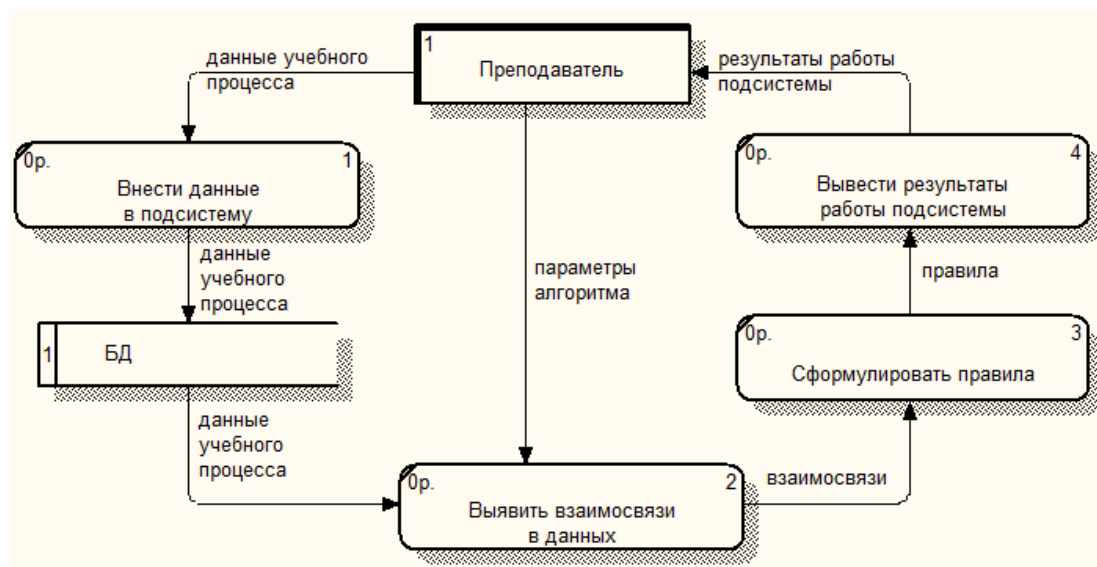


Рисунок 2 – Функциональная модель подсистемы ИАД учебного процесса

Логическая модель БД (см. рис.3) отражает необходимые понятия предметной области, которые соответствуют данным используемым подсистемой. Представленная логическая модель БД разработана средствами ER-диаграмм. ER-диаграммы отражают сущности предметной области и связи между ними. Сущности соответствуют понятиям предметной области и имеют ряд атрибутов, которые характеризуют их. Например, сущность Обучаемый может иметь атрибут Успеваемость. Связи отражают логические отношения между сущностями (один к одному, один ко многим и многие ко многим).

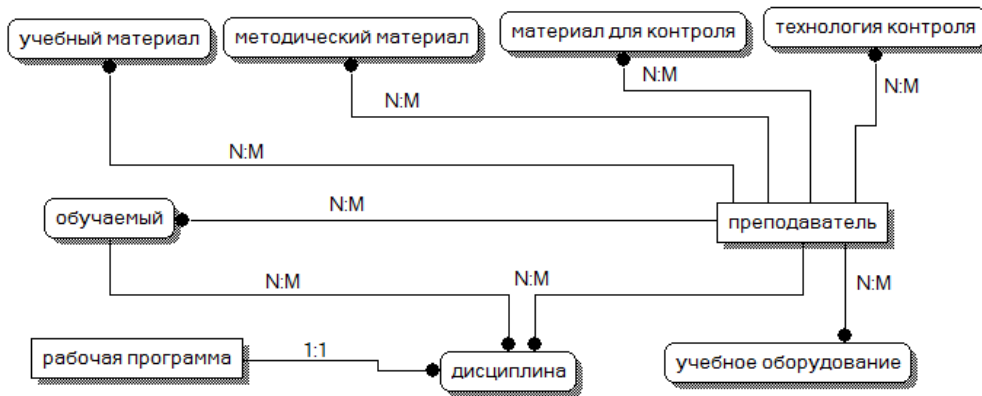


Рисунок 3 – Логическая модель БД подсистемы ИАД учебного процесса

Алгоритм работы пользователя подсистемы ИАД учебного процесса (см. рис.4) отражает основные шаги, которые необходимо пройти пользователю при работе с подсистемой. После запуска подсистемы пользователю необходимо ввести данные, которые будут анализироваться, и параметры алгоритма. Далее следует запустить алгоритм ИАД и просмотреть результаты его работы. Если результаты работы алгоритма ИАД являются неприемлемыми по тем или иным причинам, например, выявлено слишком много правил с невысокой достоверностью, то необходимо изменить параметры алгоритма и повторить процедуру запуска алгоритма. Если параметры алгоритма ИАД изменять не требуется, можно произвести анализ других данных или закончить работу с подсистемой.



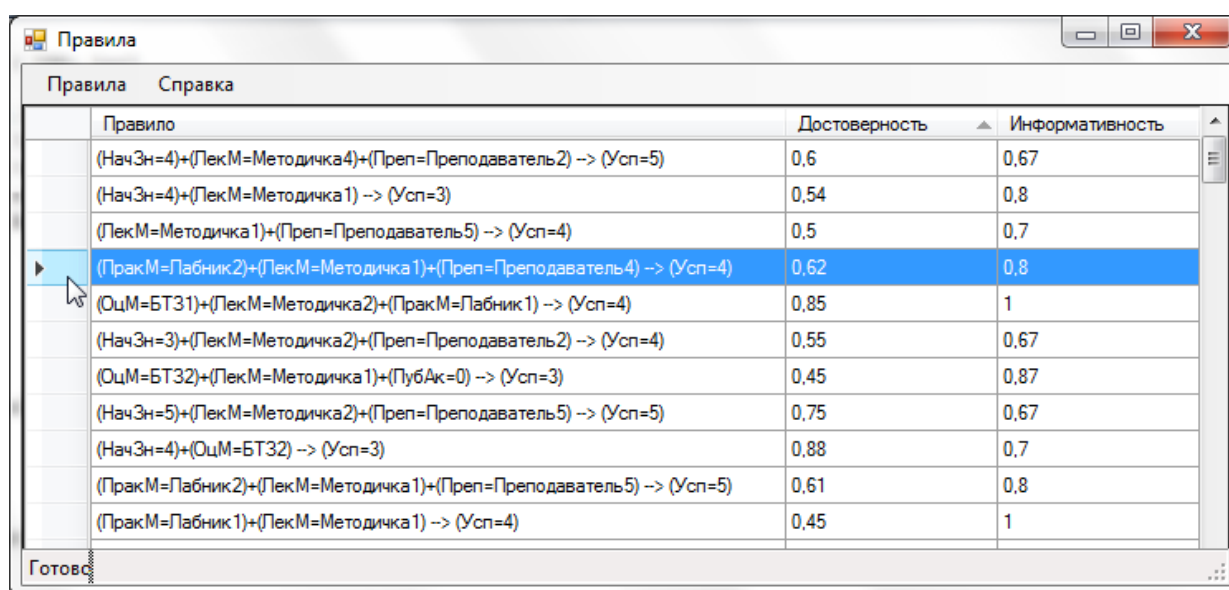
Рисунок 4 – Алгоритм работы пользователя подсистемы ИАД учебного процесса

Представленные модели и алгоритмы описывают различные аспекты подсистемы ИАД учебного процесса со степенью достаточной для ее программной реализации.

Интерфейс подсистемы ИАД учебного процесса

Проектирование графического интерфейса пользователя (ГИП) включает разработку экранных форм, которые будут соответствовать задачам, решаемым подсистемой. При разработке ГИП подсистемы ИАД учебного процесса учитывались следующие основные принципы: унифицированность, простота, эргономичность. Унифицированность интерфейса выражена в использовании типичных элементов управления и стандартного вида окон программы. Так главное меню расположено в верхней части окна, элементы управления размером окна и его закрытия – в правом верхнем углу и т.д. Простота выражается в том, что интерфейс является интуитивно понятным. Все надписи имеют минимальный объем и максимальную смысловую нагрузку, имеется элемент меню Справка, содержащий вспомогательные сведения для пользователя и т.д. Эргономичность заключается в представлении информации в виде, удобном для восприятия человеком. Так для всех окон выбрана нейтральная цветовая гамма, размер и начертание шрифта являются удобными для прочтения и т.д.

В качестве примера приведено окно с результатами работы алгоритма ИАД (см.рис.5).



Правило	Достоверность	Информативность
(НачЗн=4)+(ЛекМ=Методичка4)+(Преп=Преподаватель2) -> (Усп=5)	0,6	0,67
(НачЗн=4)+(ЛекМ=Методичка1) -> (Усп=3)	0,54	0,8
(ЛекМ=Методичка1)+(Преп=Преподаватель5) -> (Усп=4)	0,5	0,7
(ПракМ=Лабник2)+(ЛекМ=Методичка1)+(Преп=Преподаватель4) -> (Усп=4)	0,62	0,8
(ОцМ=БТЗ1)+(ЛекМ=Методичка2)+(ПракМ=Лабник1) -> (Усп=4)	0,85	1
(НачЗн=3)+(ЛекМ=Методичка2)+(Преп=Преподаватель2) -> (Усп=4)	0,55	0,67
(ОцМ=БТЗ2)+(ЛекМ=Методичка1)+(ПубАк=0) -> (Усп=3)	0,45	0,87
(НачЗн=5)+(ЛекМ=Методичка2)+(Преп=Преподаватель5) -> (Усп=5)	0,75	0,67
(НачЗн=4)+(ОцМ=БТЗ2) -> (Усп=3)	0,88	0,7
(ПракМ=Лабник2)+(ЛекМ=Методичка1)+(Преп=Преподаватель5) -> (Усп=5)	0,61	0,8
(ПракМ=Лабник1)+(ЛекМ=Методичка1) -> (Усп=4)	0,45	1

Рисунок 5 – Окно с результатами работы алгоритма ИАД

Окно с результатами работы алгоритма ИАД выводит выявленные правила и их показатели достоверности и информативности.

Интерфейс подсистемы ИАД отвечает всем предъявляемым к нему требованиям: соответствует решаемым подсистемой задачам, интуитивно понятен, эргономичен и унифицирован.

Заключение

В работе описан процесс разработки подсистемы ИАД учебного процесса. Приведены основные шаги модифицированного Априорного алгоритма для анализа данных учебного процесса. Данная модификация отличается использованием показателя информативности, характеризующего правила с точки зрения их актуальности для пользователя подсистемы. Приведена структура подсистемы ИАД учебного процесса, отображающая состав ее модулей и их связи между собой и с БД. Функциональная модель отражает особенности функционирования подсистемы, в том числе связи с пользователем. Логическая модель БД отражает понятия предметной области и их взаимосвязи. Приведенный алгоритм работы пользователя отражает основные шаги работы с подсистемой. В работе также описан ГИП подсистемы. Приведенные модели и алгоритмы иллюстрируют функциональность и особенности работы и устройства подсистемы ИАД учебного процесса.

Использование подсистемы ИАД учебного процесса в образовании позволит производить анализ данных учебного процесса, с целью выявления зависимости успеваемости студентов от различных факторов учебного процесса. Полученные правила взаимосвязей могут быть использованы с целью корректировки учебного процесса таким образом, чтобы минимизировать влияние факторов, приводящих к низкой успеваемости и максимизировать влияние факторов, приводящих к высокой успеваемости.

Литература

1. Бова В.В., Лещанов Д.В., Запорожец Ю.Ю., Курейчик Л.В. Онтологическое моделирование разнородных предметных знаний в интеллектуальных обучающих системах // Информатика, вычислительная техника и инженерное образование. -2016. - № 4 (24). - С. 60-70.
2. Башмаков А.И., Старых В.А. Принципы и технологические основы создания открытых информационно-образовательных сред. – ФГУ ГНИИ ИТТ «Информика». – М.: БИНОМ. Лаборатория знаний, 2010. – 719 с.
3. Бова В.В. Концептуальная модель представления знаний при построении интеллектуальных информационных систем // Известия ЮФУ. – Таганрог: Изд-во ЮФУ, 2014. – № 7. – С. 109-117.
4. Курейчик В.В., Бова В.В. Моделирование процесса представления знаний в интеллектуальных обучающих системах на основе компетентностного подхода. // Открытое образование. – М., 2014. – № 3 (104). – С. 42-48.
5. Бова В.В. Представление динамических моделей сложных объектов в интеллектуальных обучающих системах // Известия ЮФУ. – Таганрог: Изд-во ЮФУ. – 2010. – № 12. – С. 37-43.
6. Кравченко Ю.А., Запорожец Д.Ю., Лежебоков А.А. Способы интеллектуального анализа данных в сложных системах // Известия КБНЦ РАН. – Нальчик: Изд-во КБНЦ РАН, 2012. – № 3 (47). – С. 52-57.
7. Бова В.В., Кулиев Э.В., Новиков А.А. Особенности использования интерактивных заданий в современных средствах компьютерного обучения // Открытое образование. - М.: 2014. – № 3 (104). – С. 18-24.
8. Казмина И.И., Нужнов Е.В. Применение интеллектуального анализа данных в интегрированной инструментальной среде поддержки образовательных процессов // Материалы Третьей Международной конференции «Автоматизация управления и интеллектуальные системы и среды». Научный сборник. – Нальчик: Издательство КБНЦ РАН, 2012. – Т.2. – С. 111-114.
9. Казмина И.И., Нужнов Е.В. Интеллектуальный анализ данных в образовательной деятельности на основе алгоритма взаимосвязей // Информатика, вычислительная техника и инженерное образование. -2015. - № 1 (21). - С. 58-64.
10. R. Agrawal, and R.Srikant, Fast algorithms for mining association rule, in Proceedings of the 20th International Conference on Very Large Databases. – 1994. – pp. 487-499.

Казмина И.И. Разработка подсистемы интеллектуального анализа данных учебного процесса. В работе представлена модификация Априорного алгоритма для интеллектуального анализа данных учебного процесса, описаны проектирование и интерфейс подсистемы интеллектуального анализа данных учебного процесса. Предложенная модификация Априорного алгоритма использует основные показатели, участвующие в формировании правил взаимосвязей в оригинальном алгоритме, а также вводит показатель информативности для характеристики выявленных правил с точки зрения информационных потребностей пользователя подсистемы. В рамках проектирования подсистемы интеллектуального анализа данных учебного процесса представлены ее структура, функциональная модель, схема БД, а также алгоритм работы пользователя с подсистемой. Представлен интерфейс разработанной подсистемы.

Ключевые слова: Интеллектуальный анализ данных, Априорный алгоритм, модификация Априорного алгоритма, подсистема интеллектуального анализа, учебный процесс, проектирование.

Kazmina I.I. Development of educational process data mining subsystem. In this paper modification of A priori algorithm for educational process data mining is presented, design and interface of educational process data mining subsystem are described. The modification of A priori algorithm uses basic indicators, which used in A priori algorithm for rules generation, and new indicator – information capability, which used for description generated rules with a view to subsystem user information needs. Within the scope of educational process data mining subsystem design its structure, functional model, database scheme and user work algorithm are presented. Interface of designed subsystem is presented.

Keywords: Data mining, A priori algorithm, modification of A priori algorithm, data mining subsystem, educational process, design.

Анализ основных технологий распознавания речи

Д.А. Капустин И.С. Антонюк А.А. Гречина Е.Ю. Суворова В.В. Швыров
Луганский национальный университет имени Тараса Шевченко
kap-kapchik@mail.ru

Д.А. Капустин, И.С. Антонюк, А.А. Гречина, Е.Ю. Суворова, В.В. Швыров. Анализ основных технологий распознавания речи. Статья посвящена обзору и анализу систем и технологий по распознаванию речи. Были исследованы вопросы выявления наиболее перспективных технологий и методик, используемых в этой области. Анализовалась возможность их комбинированного использования и совершенствования за счет внедрения передовых разработок в области распознавания.

Ключевые слова: распознавание, речь, технология, библиотека, приложение.

Постановка проблемы

В последнее время находится под наблюдением стремительное развитие информационных технологий. Одним из самых важных направлений исследований в данной области можно отнести задачи хранения, обработки и передачи мультимедийных данных. Однако, в многочисленных задачах анализа мультимедиа-данных компьютеру так и не удалось в корне заменить человека. К ним относятся: параллельный перевод, автоматическая сегментация изображений и видео последовательностей, автоматическая стенография. Одной из ключевых задач обработки мультимедиа информации является цель распознавания и анализа обыденной речи человека.

В задачу анализа распознавания речи входит обширный спектр задач. Классически они делятся на три подкласса: задачи идентификации, классификации и диагностики [1].

К задачам идентификации относят задачи верификации и идентификации дикторов. К задачам классификации относят распознавания ключевых слов, распознавания слитной речи и задачи семантического анализа речи. К классу задач диагностики относят задачи определения психофизического состояния диктора.

На сегодняшний день задача распознавания речи становится все более популярна. Кругозор для использования полученных решений довольно обширен: автоматические стенографы, автоматизированные справочные терминалы с речевым управлением, синхронные переводчики, системы сжатия и передачи речевого сигнала с высоким качеством, системы сегментации, индексации и поиска мультимедиа информации.

Обработка в системах распознавания речи заключается в следующих задачах:

- фильтрация и подавление шума;
- сегментация на информативные участки;
- определение информативных параметров;
- распознавание.

Практика речевого интерфейса несет в себе актуальность для таких задач, как распознавание и анализ речи, повышение эффективности преобразователей «речь-текст». Со временем все больше и больше набирает популярность использования речевых технологий для организации диалога человек-машина [2].

Основополагающим направлением современных речевых технологий можно отнести к разработкам единых алгоритмов параметризации речевого сигнала, основанных на физической модели речевого тракта и природе речевого сигнала – определение существенных параметров и разработка математических моделей, которые применяются как для синтеза, так и для анализа речи. Все это осложняется тем, что для речевого сигнала присуще иметь двойственную природу: с одной стороны, это акустический сигнал, отражающий процесс распространения энергии акустических колебаний в упругой среде, с другой стороны, речевой сигнал несет смысловую нагрузку в виде информации об эмоциональном состоянии диктора, а также содержит субъективные параметры, позволяющие найти различия одного диктора от другого. Исходя из этого можно утверждать, что не существует единого мнения о природе формы спектра речевых сигналов, методах их анализа и синтеза [3].

Современное развитие науки и техники позволяет создавать системы, основанные на сложной математической конструкции, но имеют свойство работать в режиме реального времени.

В качестве отдельной задачи распознавания следует выделить распознавание математических формул. Для преобразования звуков речи и перевода в формульное обозначение необходимо использовать их классификацию. При создании подобной системы разработчик сталкивается с определенными проблемами. Первая: отсутствие математической модели семантики речевого сигнала. Следовательно, для определения семантики речевого сигнала могут применяться только вероятностные и эвристические методы, не дающие точного результата, и точность которых обратно пропорциональна количеству смысловых единиц, на которые они рассчитаны. Вторая: индивидуальные характеристики говорящего, а именно специфика произношения, акценты, ударения, хезитации. Третья: необходимость работы со спонтанной речью и определение присутствия ключевого слова. Четвертая: различия в акустической обстановке, шумы. Из этого следует, что полное распознавание математических формул на сегодняшний день не достижимо. Актуальность исследований в этой области несомненна.

Анализ последних исследований и публикаций показал, что речевое распознавание находит все более новые и актуальные области применения, от приложений, исполняющих преобразование речевой информации в текст и завершая управление автомобилем на аппаратном уровне [4]. Условно распознавание речи можно поделить на следующие группы:

- программные ядра для аппаратных реализаций систем распознавания речи;
- наборы библиотек, утилит для разработки приложений, использующих речевое распознавание;
- независимые пользовательские приложения, осуществляющие речевое управление и/или преобразование речи в текст;
- специализированные приложения, использующие распознавание речи;
- устройства, выполняющие распознавание на аппаратном уровне;
- теоретические исследования и разработки.

Цель статьи: определение наиболее перспективной технологии преобразования речи в текст.

Программные ядра для аппаратных реализаций

В основе речевой технологии лежит так называемый «engine» или ядро программы – пакет данных и правил, с помощью которых в дальнейшем осуществляется обработка данных. По назначению ядра делятся на TTS и ASR engine. TTS (Text-to-Speech) синтезирует речь по тексту, а ASR (Automatic Speech Recognition) engine – распознаёт речь.

Существует несколько крупных производителей, которые занимаются разработкой ASR ядер, такие, как SPIRIT, Advanced Recognition Technologies, IBM. В 2010 году компания SPIRIT вывела на рынок программное обеспечение для веб-видеоконференций VideoMost [5]. К преимуществам этой технологии относится распознавание речевых команд в сильно зашумленной среде; дикторонезависимость системы; обработка речи в реальном времени и неограниченная длина словарного потока. К недостаткам: неравномерное качество распознавания.

Компания ProVox Technologies основываясь на программное ядро ViaVoice от IBM разработала дикторозависимую систему для отчетов врачей-радиологов VoxReports [6]. По результатам тестирований система справилась на 95-98% при обычном темпе распознавания (до 180 слов в минуту) в независимости от диктора. Главным недостатком может послужить набор специфических медицинских терминов в словаре.

IBM занялась созданием машины, которая способна распознать разговорную речь на 20 языках, а также воспринимать различные диалекты, акценты и контексты, что даст возможность в совершенстве переводить юридические и медицинские документы, и даже свидетельские показания в зале суда. По словам IBM, идея состоит в том, чтобы заставить машины сравниваться, а затем и превзойти возможности человека в распознавании речи. По задумке IBM технология должна вплотную приблизиться и даже превзойти человеческое мышление.

«Сакрамент» – белорусская компания с многолетней историей работы в сфере синтеза и распознавания речи. Sakrament ASR Engine – программная разработка, рассчитанная на применение в различных аппаратных системах и программных приложениях, использующих технологии распознавания речи.

Преимущества: точность распознавания 95-98%, дикторонезависимость, языконезависимость, распознавание слитной речи в виде выражений и небольших предложений.

Недостатки: медленный темп речи, плохая стыковка звуков, монотонность.

Наборы библиотек для разработки приложений

Исходя из лингвистических особенностей человеческой речи, дополнительные артикуляционные данные дают возможность для более точного анализа речи диктора и в последующем автоматически разделить звуковую волну на отдельные участки. Также, при анализе аудиовизуального голосового сигнала во временной

динамике наблюдается перспектива фиксирования открытых и закрытых слогов, звонких, шипящих, ударных, и другие речевые единицы.

Не так давно компания Google предоставила доступ к программным интерфейсам Cloud Speech API. Это дало возможность распознавание речи через микрофон или аудиозаписей из файлов. Система выдаёт распознанный текст мгновенно в процессе работы [7]. Google утверждает, что Speech API работает даже в сильно зашумленном месте, так что материал не придется очищать от лишних помех или удалять шум [8].

Главные преимущества сервиса: наличие русскоязычного интерфейса; возможность просмотра и выбора вариантов распознавания; наличие голосовых подсказок; автоотключение записи после длительной паузы. Единственным недостатком сервиса является не совсем привычный для подобных сервисов алгоритм работы.

Большинство коммерческих приложений для Windows используют для синтеза и распознавания речи Microsoft Speech Application Interface (SAPI) – интерфейс программирования речи от «Майкрософт» [9]. SAPI снижает общее количество написания кода для продуктов, использующих распознавание и синтез текста, делая речевые технологии более приемлемыми и устойчивыми для решения широкого круга задач. Что касается движков, то их можно поделить на два типа, а именно SAPI - это движки синтеза (TTS – Text-to-Speech) и распознавания речи. TTS системы синтезируют из текстовых строк или файлов звуки синтетической речи. Распознавание речи преобразует человеческую речь в читабельные текстовые строки или файлы. Известная всем корпорация IBM уже более 30 лет заинтересована в автоматическом распознавании речи и добились больших успехов. В своем блоге Microsoft сообщили, что их технология распознавания речи на сегодняшнее время уже может сравниться по точности с человеком. Согласно данным ученых, сейчас показатель ошибок системы в распознавании слов составляет 5,9%. Это практически тот же результат, что показывают люди, которые профессионально расшифровывают надиктованный текст. Система и люди проходили испытания на одной базе записанных разговоров [10].

Преимущества: длинные предложения могут быть обработаны намного легче и быстрее на стороне клиента; интеграция с любой программой; высокая точность распознавания.

Недостатки: работает только на платформе Windows; потребуется изменить параметры безопасности в Internet Explorer, чтобы разрешить инициализации скриптов и элементов управления ActiveX.

2013 год для Яндекса стал представлением собственной технологии распознавания речи Yandex SpeechKit [11]. Тогда же была выпущена библиотека Mobile SDK, с помощью которой легко можно встраивать распознавание в приложения для Android, iOS и Windows Phone. Сейчас эту технологию можно встретить в смартфонах, планшетах. Каждый день Yandex SpeechKit обрабатывает миллионы запросов. Однако умение понимать голос может пригодиться не только смартфонам и планшетах. Поэтому Яндекс запустил SpeechKit Cloud — облачный сервис распознавания речи. Он позволяет добавлять функции распознавания речи в любые разработки и устройства — от компьютерной игры до робота [12]. Качество распознавания речи зависит от тематики запроса. К примеру, если вы диктуете смс или заметку, оно составляет 82%, если произносите короткий запрос – 85%. Распознавание осуществляется не посредственно на серверах Яндекса, потому что для анализа и обработки требуются много ресурсов, рассчитанных на высокие нагрузки. Преимущества: хорошая совместимость с русским языком, бесплатное использование программы. Недостатки: отсутствие библиотек для offline разработок.

Независимые пользовательские приложения

Dragon NaturallySpeaking Preferred фирмы Dragon Systems— единственная программа, приблизившаяся к тому, чтобы соответствовать заявленным характеристикам. В целом она весьма близко подходит к достижению заявленной безошибочности распознавания - 95% [13]. Преимущества: высокая безошибочность распознавания, простота использования. Недостатки: неудобный ввод чисел, посредственное управление экраном.

Компания М.С. Технолоджи разработала программу «Микросервис» для управления функциями операционных систем Windows 98/Me/2000/XP и ввода текста в любой редактор. Программа поддерживает русский и английский языки и содержит словарь порядка 10000 слов. Компания IC приобрела права на это ПО и выпустила его под названием «Диктограф».

Однако, по данным тестирований, «Микросервис» от М.С. Технолоджи показал неудовлетворительные результаты – 30-50% правильно распознанных слов и команд.

К сожалению, российский рынок программных средств распознавания речи представлен единичными разработками [14, 15, 16]. Из всех программ, изначально разрабатываемых для русского языка, только ПО от белорусской компании «Сакрамент» может конкурировать по качеству распознавания с зарубежными аналогами, так как имеет поддержку русского и английского языков, содержательную базу словаря, интегрируется с другими программами. К недостаткам относится небольшой результат распознаваемой речи.

Специализированные приложения

Распознавание речи встречается не только для ввода текста или подачи команд, но и для более специфичных целей. Так компания «Центр Речевых Технологий» производит программные продукты, технологии и образцы техники для подразделений МВД, ФСБ, МЮ, МЧС, МО, служб экстренной помощи, центров обработки вызовов и для других пользователей, в деятельности которых особое значение придается регистрации и обработке речевой информации. Компания успела создать следующие приложения: «ИКАР Лаб» – инструментальный комплекс для криминалистического исследования фонограмм речи, «Трал» – рассчитан для распознавания телефонных разговоров, «Территория» – для диагностики диалектов и акцентов русской устной речи [17]. Преимущества:

- множество алгоритмов для идентификационного исследования;
- широкий набор параметров и настроек для максимизации эффективности;
- высокая надежность биометрического движка.

Недостатки: сравнение происходит один-к-одному; для эффективной работы необходим квалифицированный специалист; режим распознавания происходит на полуавтоматическом режиме.

Устройства, выполняющие распознавание на аппаратном уровне

С целью применения функций речевого распознавания в различных устройствах, роботах, играх, разрабатываются аппаратные способы решения данной проблемы. Для этого американская компания Sensory Inc. разработала интегральную схему Voice Direct 364, исполняющую дикторозависимое распознавание небольшого числа команд (около 60) после предварительного обучения [18]. Преимущества: применяется в различных устройствах. Недостатки: большая вычислительная сложность, так как команды могут включать слова, которые плохо распознаются с помощью алгоритма.

Петербургская компания «Центр речевых технологий», целенаправленно занимающаяся технологиями распознавания речи, еще в 2008 г. создала технологию распознавания слитной русской речи «Руссограф», для создания которой был создан уникальный для России набор речевых баз данных, в который входят записи более чем 3000 дикторов общей длительностью около 300 часов, собранных с учетом 5 диалектных групп русского языка. Уникальность данной технологии заключается в том, что многочисленные системы распознавания речи, применяемые к другим языкам, не обеспечивают такого же качества распознавания при работе с русским языком. Преимущества: распознавание русской слитной речи; большой набор речевой базы данных. Данная технология быстро справляется с редкой диалектной группой слов, но плохо распознают более или менее сложные предложения из-за отсутствия четкой структуры в русской речи.

Выводы и перспективы следующих исследований

Анализ области технологий речевого распознавания показал, что, хотя подобные технологии и получили широкое распространение в различных областях науки и техники, существует множество нерешенных проблем, связанных прежде всего с подавлением уровня шума и выделением необходимого звукового канала, зависимостью уровня распознавания от конкретного диктора, языка, и предметной области распознавания. Кроме того, на уровень распознавания влияет несовершенство применяемых технологий. К основным недостаткам сетевых технологий распознавания речи следует отнести необходимость постоянного и широкополосного доступа к сетевым ресурсам. Особенно остро это проявляется при распознавании в режиме реального времени. К недостаткам консольных решений следует отнести повышенные требования к ПК пользователя, и задержка при обновлении библиотек распознавания.

На наш взгляд к наиболее перспективным следует отнести комбинированные технологии, которые включает консольную часть и имеют возможность оперативного перевода сложных участков, в том числе и с помощью сетевых ресурсов, что приведет к снижению сетевого трафика и повышению качества распознавания. Кроме того, необходимо ввести диалог между системой распознавания и пользователем даже в случае распознавания слитной речи. Что позволит системе более точно подбирать шаблон (в случае необходимости) и повысит качество распознавания.

Установлено, что на сегодняшний день отсутствуют комплексные решения по распознаванию математических формул. Следовательно, это является актуальной проблемой для исследований в отрасли технических и естественных наук.

Литература

1. Винцюк Т.К. "Анализ, распознавание и интерпретация речевых сигналов." -Киев: Научн. мысль, 1987. -262 с.
2. Чучупал В.Я., Чичагов А.В., Маковкин К.А. Алгоритмы и программы для предобработки зашумленных речевых сигналов. /Тезисы 8-й Всероссийской конференции "Математические методы распознавания образов". Москва, 1997, с.222.

3. Э. П. Нейбург, Т. Б. Мартин, Дж. Р. Уэлч, В. У. Зу, Р. М. Шварц, Дж. Е. Шуп, А. Р. Смит, М. Р. Самбур, Ф. Хейс-Роз, Г. Гудмэн, Р. Редди. Методы автоматического распознавания речи: В 2-х книгах. Пер. с англ. /Под ред. У. Ли. – М.: Мир, 1983. – Кн. 1. 328 с.
4. Чесебиев, И. А. Компьютерное распознавание и порождение речи / И.А. Чесебиев. - М.: Спорт и Культура-2000, 2008. - 128 с.
5. Программный продукт для веб-видеоконференций VideoMost [Электронный ресурс] – Дата обновления 23 октября 2016, Режим доступа: <http://www.videomost.com/companу> – Заглавие с экрана.
6. Компания ProVox Technologies [Электронный ресурс] – Дата обновления 23 октября 2016, Режим доступа: <http://www-03.ibm.com/software/products/en/ibm-mobilefirst-foundation> - Заглавие с экрана.
7. Современные технологии распознавания речи [Электронный ресурс] – Дата обновления 23 октября 2016, Режим доступа: dialog-21.ru/media/2458/leonovich-a.pdf – Заглавие с экрана.
8. Google открывает API для распознавания речи на 80 языках [Электронный ресурс] – Дата обновления 24 октября 2016, Режим доступа: <https://geektimes.ru/post/273282/> – Заглавие с экрана.
9. Microsoft Speech Application Programming Interface [Электронный ресурс] – Дата обновления 24 октября 2016, Режим доступа: <http://speech-soft.ru/info/speech-api> – Заглавие с экрана.
10. Технология распознавания речи сравнялась по точности с человеком – Microsoft. [Электронный ресурс] – Дата обновления 23 октября 2016, Режим доступа: <http://odnako.su/news/science/-553467-tehnologiya-raspoznavaniya-rechi-sravnyalas-po-tochnosti-s-chelovekom-microsoft/> – Заглавие с экрана.
11. Яндекс. SpeechKit API [Электронный ресурс] – Дата обновления 24 октября 2016, Режим доступа: <http://api.yandex.ru/speechkit/> – Заглавие с экран.
12. Речевые технологии SpeechKit Cloud [Электронный ресурс] – Дата обновления 24 октября 2016, Режим доступа: <https://tech.yandex.ru/speechkit/cloud/> – Заглавие с экрана.
13. Распознавание речи Dragon NaturallySpeaking Preferred [Электронный ресурс] – Дата обновления 24 октября 2016, Режим доступа: <http://speech-soft.ru/prog/dragon> – Заглавие с экрана.
14. Хабрахабр. Распознавание речи [Электронный ресурс] – Дата обновления 24 октября 2016, Режим доступа: <http://habrahabr.ru/post/64572/> – Заглавие с экрана.
15. Кленяева Г.В. Современные проблемы речевой акустики и построения систем автоматического распознавания речи / Г.В. Кленяева, А.П. Преображенский // Вестник Воронежского института высоких технологий. 2007. Т. 1. № 2-1. С. 071-074.
16. Русскоязычный синтез речи [Электронный ресурс] – Дата обновления 24 октября 2016, Режим доступа: <http://www.softkey.info/reviews/review1140.php> – Заглавие с экрана.
17. Синтез и распознавание речи. Центр речевых технологий [Электронный ресурс] – Дата обновления 24 октября 2016, Режим доступа: <http://www.speechpro.ru> – Заглавие с экрана.
18. Центр речевых технологий [Электронный ресурс] – Дата обновления 24 октября 2016, Режим доступа: <https://habrahabr.ru/company/speechpro/profile/> – Заглавие с экрана.
19. Kennedy P.R., Guenther F.H. (2010). Brain-computer interfaces for speech communication. *Speech Communication* 52:367–379. 2010
20. Голубцов С. В. Задачи и перспективы распознавания речи. — Тез. докл. 6-го Всесоюз. семинара «Автоматическое распознавание слуховых образов», Таллин, 1972, с. 64—73.

Д.А. Капустин, И.С. Антониук, А.А. Гречина, Е.Ю. Суворова, В.В. Швыров. Анализ основных технологий распознавания речи. Статья посвящена обзору и анализу систем и технологий по распознаванию речи. Были исследованы вопросы выявления наиболее перспективных технологий и методик, используемых в этой области. Анализировалась возможность их комбинированного использования и совершенствования за счет внедрения передовых разработок в области распознавания.

Ключевые слова: *распознавание, речь, технология, библиотека, приложение.*

D. A. Kapustin, I. S. Antoniuk, A. A. Grechina, E. Y. Suvorova, V. V. Shvyrov. Analysis of the core technologies of speech recognition. The article is devoted to consideration and analysis of systems and technologies for speech recognition. Questions were investigated of identifying the most of perspective technologies and methods used in this field. To analyze the possibility of their combined use and improvement through the introduction of advanced developments in the field of recognition.

Keywords: *recognition, speech, technology, library, the program.*

СЕКЦИЯ «КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В ФУНДАМЕНТАЛЬНЫХ И ПРИКЛАДНЫХ ИССЛЕДОВАНИЯХ»

УДК 004.7

Анализ трафика сети с беспроводным доступом

Д.В. Бельков, Е.Н. Едемская

Донецкий национальный технический университет

belkov65@list.ru

Бельков Д.В., Едемская Е.Н. Статистический анализ трафика сети с беспроводным доступом. Недавние исследования трафика пакетов в сети показали, что это - самоподобный процесс и его Марковские (без последствия) модели приводят к некорректным решениям. Поэтому важной задачей является дальнейшее исследование современного трафика. Данная статья представляет результаты статистического анализа временных рядов задержки в сети с GPRS-доступом на стороне отправителя и UMTS-доступом со стороны получателя.

Ключевые слова: трафик, задержка, джиттер, функция автокорреляции, спектральная плотность, параметр Херста.

Введение

В связи с тенденциями объединения различных телекоммуникационных приложений на базе универсальной сетевой инфраструктуры актуальной проблемой является разработка методов анализа и синтеза информационно-управляющих систем. Данная работа посвящена одной из важных задач развития этого направления – исследованию процессов в компьютерных сетях с беспроводным доступом.

Использование в сетях интегрированных информационных приложений и пакетной коммутации сопровождается сложными явлениями, изучение которых может быть выполнено с помощью вероятностных подходов.

Характерным для процессов пакетной передачи является обнаруженное на практике свойство фрактальности. В процессе передачи возникают большие всплески при относительно низком среднем уровне трафика. Данное явление, которое значительно влияет на потери и задержки пакетов, не учитывается в теории телетрафика. Расчет параметров системы распределения информации по классическим формулам дает некорректные, неоправданно оптимистические результаты. Алгоритмы обработки трафика, созданные для работы с простейшим потоком неэффективны для фрактальных потоков [1-6].

Появление новых свойств сетевого трафика, необходимость обеспечения высокого качества обслуживания приложений, делают актуальным статистический анализ трафика сетей.

Целью настоящей работы является исследование структуры трафика сети с беспроводным доступом, направленное на выявление его характерных особенностей. Изучается один из основных сетевых процессов – процесс RTT-задержки, который служит для получения информации о состоянии сети методом “черного ящика”, когда через сеть пропускается последовательность пакетов, и на основании времени их прохождения до удаленного узла и обратно делаются выводы о загрузке сети.

В работе решаются следующие задачи: оцениваются плотности распределения, автокорреляционные функции (АКФ), энергетические спектры изучаемых временных рядов, определяется степень фрактальности (показатель Херста). В исследовании использовались программы Fractan и Statistika.

Реализации сетевого трафика

Для изучения выбраны шесть реализаций сетевого трафика [7], полученные в университете города Наполи (Италия). Согласно лицензии, данные свободно доступны для анализа. Измерения проводились каждые 10 миллисекунд, получено свыше 3000 отсчетов. Отправитель имел GPRS-доступ, получатель - UMTS-доступ, скорость передачи 100 rps, операционная система Windows32, на каждой из сторон. Для процесса RTT-задержки на маршруте прохождения пакета можно выделить постоянную минимальную составляющую,

обусловленную отсутствием очередей и переменную составляющую, возникающую из-за задержек в очередях. RTT-задержка измеряется на стороне источника и складывается из времени обработки, времени распространения, времени коммутации, времени ожидания в очереди. Изучаемые временные ряды показаны на рис. 1-3. Нижний уровень задержки соответствует отсутствию очередей. Условные обозначения, принятые в работе, показаны в табл. 1.

Таблица 1. Условные обозначения

Обозначение	Описание
TCP_d64	Ряд измерений задержки TCP-пакетов объемом 64 байт
TCP_d256	Ряд измерений задержки TCP-пакетов объемом 256 байт
TCP_d1024	Ряд измерений задержки TCP-пакетов объемом 1024 байт

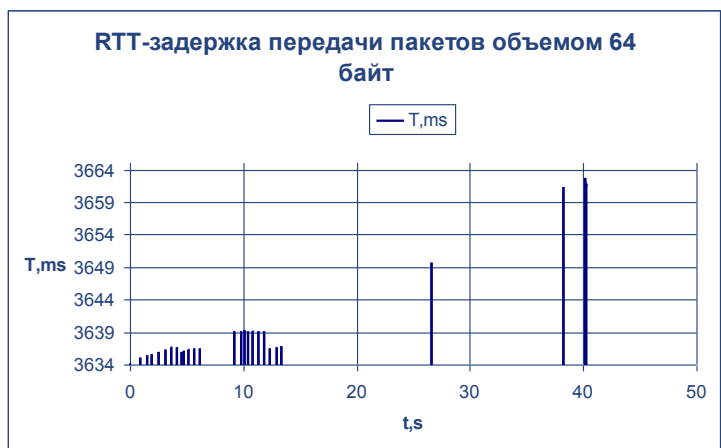


Рисунок 1 - Задержка при передаче пакетов объемом 64 байт



Рисунок 2 - Задержка при передаче пакетов объемом 256 байт

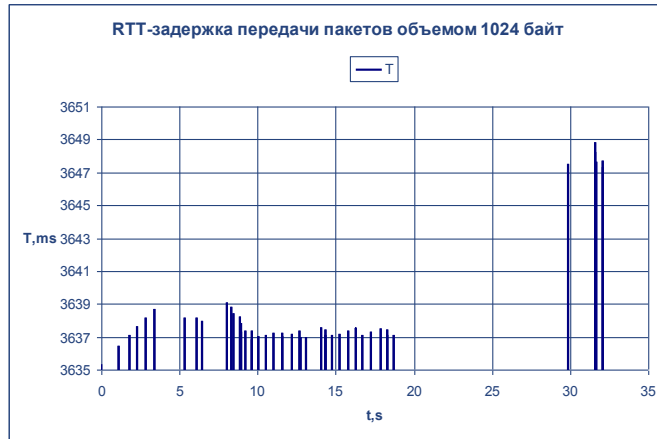


Рисунок 3 - Задержка при передаче пакетов объемом 1024 байт

Анализ плотности распределения

В этом разделе выполнен анализ плотности распределения. Оценка проводится на основании гистограмм относительных частот, показанных на рис. 4-6. Визуальный анализ позволяет сделать вывод, что исследуемые ряды не подчиняются нормальному распределению.

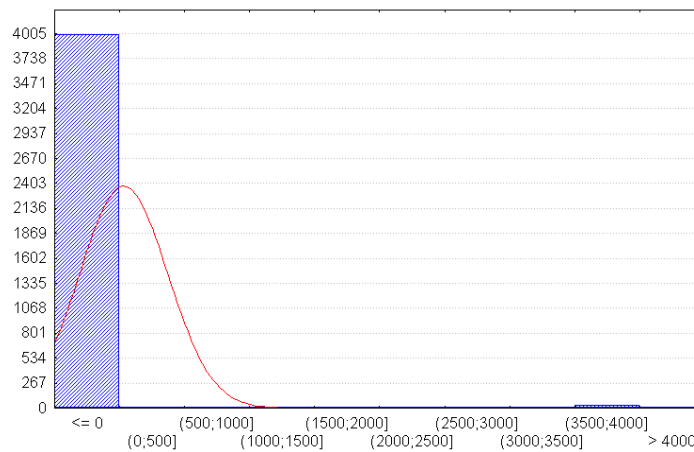


Рисунок 4 - Плотность распределения временного ряда TCP_d64

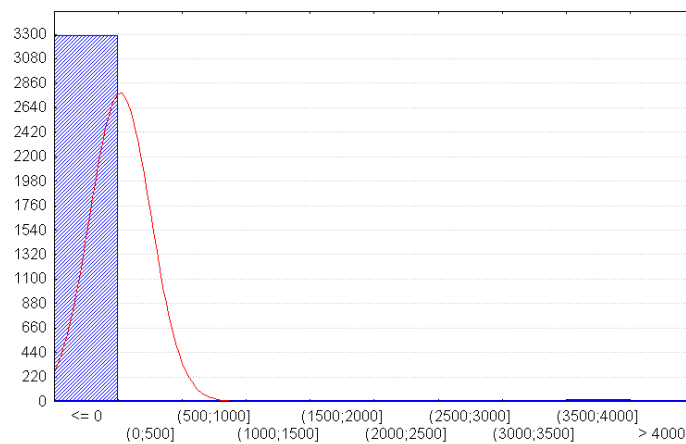


Рисунок 5 - Плотность распределения временного ряда TCP_d256

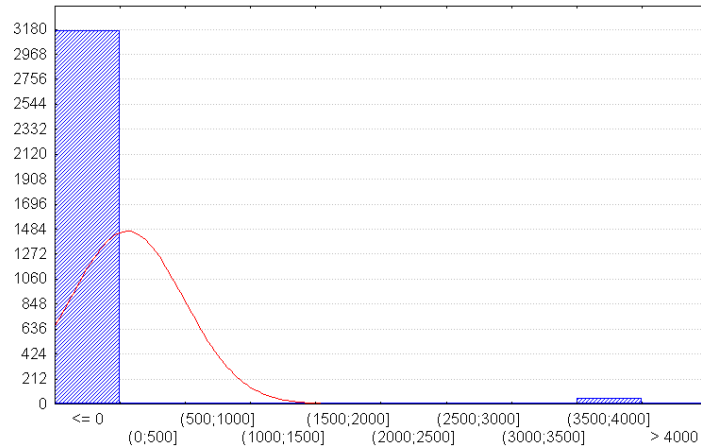


Рисунок 6 - Плотность распределения временного ряда TSP_d1024

Анализ автокорреляционных функций

В этом разделе необходимо определить обладают ли временные ряды медленно убывающей зависимостью или быстро убывающей зависимостью. По определению процесс X обладает медленно убывающей зависимостью (МУЗ) [long-range dependence], если для его АКФ выполняется условие $r(k) \sim k^{-\beta} \cdot L_1(k), k \rightarrow \infty$, где $0 < \beta < 1$ и L_1 - медленно меняющаяся на бесконечности функция: $\lim_{t \rightarrow \infty} \frac{L_1(tx)}{L_1(t)} = 1$ для всех x . Процессы с

МУЗ характеризуются автокорреляционной функцией, которая убывает по степенному закону при увеличении временной задержки (лага). В отличие от процессов с МУЗ, процессы с быстро убывающей зависимостью (БУЗ) [short-range dependence] обладают экспоненциально спадающей АКФ вида $r(k) \sim e^{-k}, k \rightarrow \infty$.

Вычисления АКФ выполнены с помощью программы Fractan 4.4 по формуле

$$r(k) = \frac{\sum_{i=1}^{N-\tau} (X_i - \bar{X})(X_{i+k} - \bar{X})}{(N - \tau)\sigma^2(X)}, \text{ где } \bar{X} - \text{выборочное среднее ряда } X, \sigma^2(X) - \text{выборочная дисперсия ряда } X,$$

$k=0,1,\dots$ Автокорреляционные функции исследуемых временных рядов приведены на рис. 7-9.

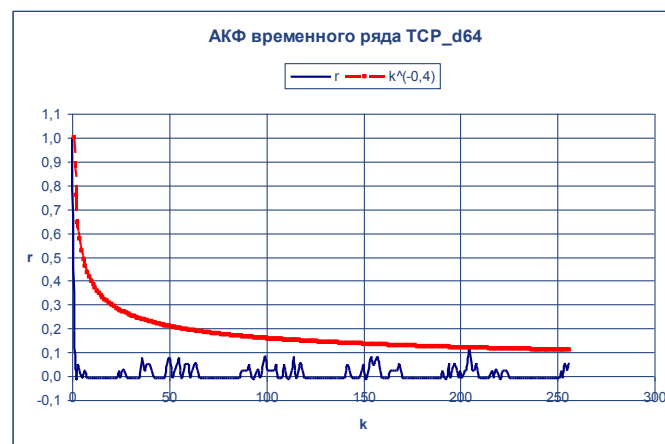


Рисунок 7 - АКФ временного ряда TSP_d64

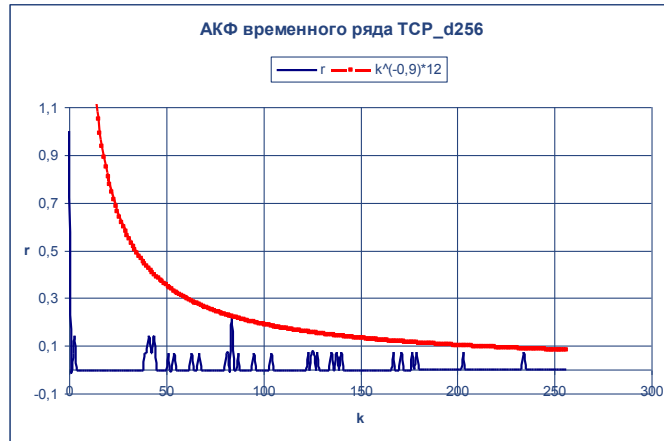


Рисунок 8 - АКФ временного ряда TCP_d256

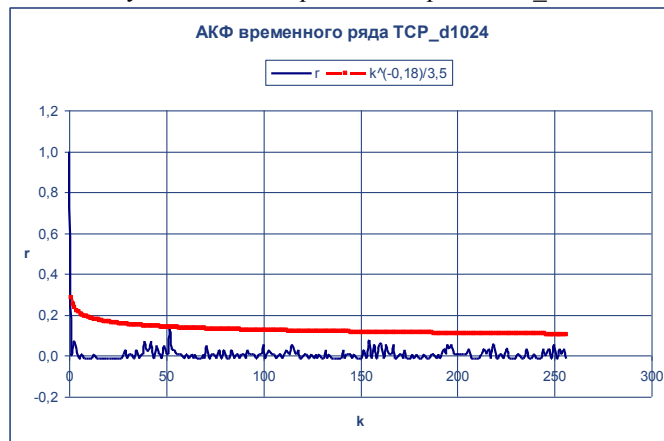


Рисунок 9 - АКФ временного ряда TCP_d1024

Спектральный анализ

В этом разделе выполнен расчет энергетических спектров исследуемых временных рядов. Известно [8,9], что в частотной области МУЗ приводит к степенному закону поведения спектральной плотности процесса. Процесс X обладает МУЗ, если для спектральной плотности $S(f) = \sum_k r(k) \cdot e^{ikf}$ выполняется условие

$S(f) \sim f^{-b} \cdot L_2(f)$, где $f \rightarrow 0$, $i = \sqrt{-1}$, $0 < b < 4$ и L_2 - медленно изменяющаяся в нуле функция.

Процесс с МУЗ обладает спектральной плотностью с особенностью в нуле: спектральная плотность стремится к бесконечности, по мере того как частота f стремится к нулю. Энергетические спектры реализаций представлены на рис. 10-12.

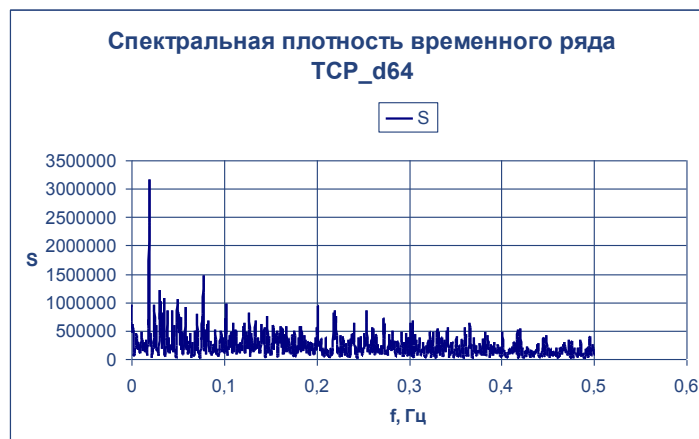


Рисунок 10 - Спектральная плотность временного ряда TCP_d64

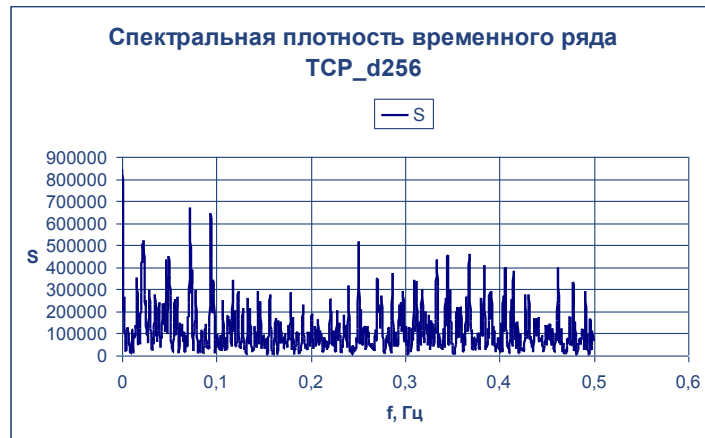


Рисунок 11 - Спектральная плотность временного ряда TCP_d256

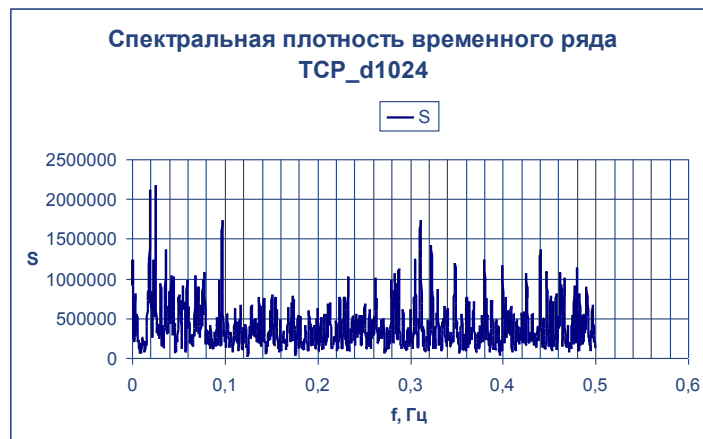


Рисунок 12 - Спектральная плотность временного ряда TCP_d1024

Визуальный анализ позволяет сделать вывод, что временные ряды RTT-задержки имеют на низких частотах самые мощные гармоники.

Исследование показателя Херста

В этом разделе оценивается показатель Херста H , который является мерой устойчивости статистического явления или мерой длительности долгосрочной зависимости процесса. Значение $H=0,5$ указывает на отсутствие долгосрочной зависимости. Корреляция между событиями отсутствует. Ряд является случайным, а не фрактальным. Чем ближе значение H к 1, тем выше степень устойчивости долгосрочной зависимости. При $0 \leq H < 0,5$ временной ряд является трендонеустойчивым (антиперсистентным). Он более изменчив, чем случайный ряд, поскольку состоит из частых реверсов спад-подъем. При $0,5 < H \leq 1$ ряд трендоустойчив. Тенденция его изменения может быть спрогнозирована. Показатель Херста связан с другими фрактальными параметрами формулами, показанными в табл. 2.

Таблица 2 Фрактальные параметры

Параметр	Расчетная формула
Фрактальная размерность D	$D = 2 - H$
Корреляционный параметр β	$\beta = 2(1 - H)$
Спектральный показатель b	$b = 2H - 1$

Для определения значения H в работе используется алгоритм R/S-анализа временных рядов, реализованный в программе Fractan 4.4 [10]. Алгоритм состоит в следующем. В исходной реализации Z последовательно выделяем отрезки $Z_\tau = z_1, z_2, \dots, z_\tau$, $\tau = 1000, 2000, \dots, 12000$, для каждого из которых

вычисляем текущее среднее $\bar{z}_\tau = \frac{\sum_{i=1}^{\tau} z_i}{\tau}$. Далее для каждого фиксированного Z_τ вычисляем накопленное отклонение для его отрезков длины t : $X_{\tau,t} = \sum_{i=1}^t (z_i - \bar{z}_\tau)$, $t=1,2,\dots,\tau$, разность (размах) между максимальным и минимальным накопленными отклонениями $R = R(\tau) = \max_{1 \leq t \leq \tau} (X_{\tau,t}) - \min_{1 \leq t \leq \tau} (X_{\tau,t})$ и стандартное отклонение $S = S(\tau)$. Показатель Херста вычисляем с помощью формулы $\frac{R}{S} = \left(\frac{\tau}{2}\right)^H$. Логарифмируя обе части этого равенства, получаем последовательность координат точек (x_τ, y_τ) , где $x_\tau = 1000, 2000, \dots, 12000$, $y_\tau = H(\tau) = \frac{\lg(R(\tau)/S(\tau))}{\lg(\tau/2)}$. Построив график зависимости $H(\tau)$ и линию тренда, определим аппроксимированное значение показателя Херста H .

На рис. 13 показан результат R/S-анализа для процесса задержки пакетов объемом 64 байт. Показатель Херста вычислен с погрешностью $\Delta = \pm 0,2846$ и лежит в диапазоне между 0,62 и 1,19. Для определения значения H по графику спектральной плотности предлагается следующий метод исследования спектральной плотности временного ряда. Как показано на рис. 14, строим линию L , которая огибает максимальные значения плотности S в направлении слева направо, и выполняем ее аппроксимацию степенной функцией. Показатель степени, равный 0,6, дает значение спектрального показателя b . Искомое значение $H=0,8$, вычисляется по формуле $H = (1 + b)/2$.

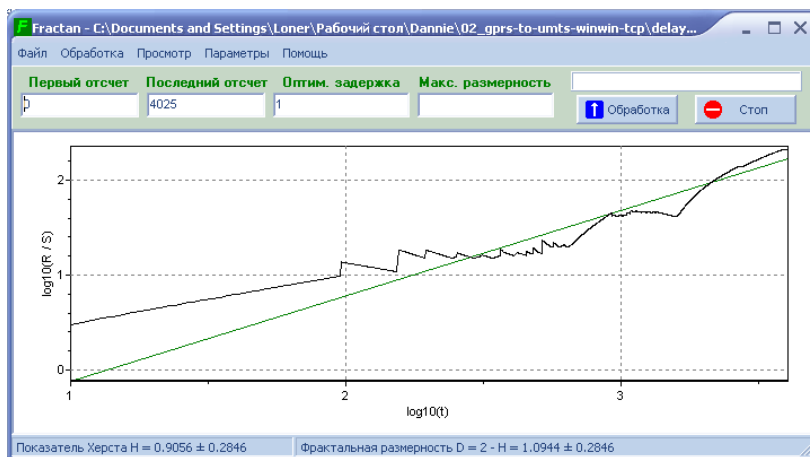


Рисунок 13 - Результат R/S-анализа временного ряда TCP_d64

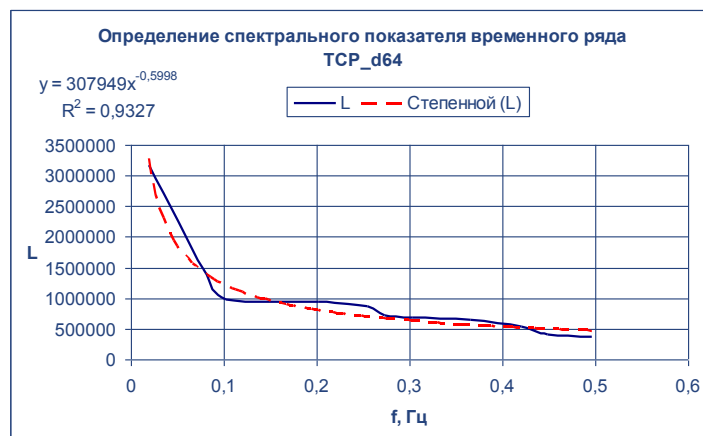


Рисунок 14 - Определение показателя b временного ряда TCP_d64

Поскольку значение H лежит в диапазоне $0,5 < H \leq 1$, то ряд TCP_d64 трендоустойчив. Для аппроксимации АКФ этого ряда вычислено значение $\beta = 0,4$ по формуле $\beta = 2(1 - H)$.

На рис. 15 показан результат R/S-анализа для процесса задержки пакетов объемом 256 байт. Показатель Херста лежит в диапазоне между 0,57 и 0,92. Определение спектрального показателя по графику спектральной плотности показано на рис. 16. Значение $b=0,09$, $H=0,55$, $\beta = 0,9$. Ряд TCP_d256 близок к случайному. На рис. 17 показан результат R/S-анализа для процесса задержки пакетов объемом 1024 байт. Показатель Херста лежит в диапазоне между 0,59 и 1,07. Определение спектрального показателя по графику спектральной плотности показано на рис. 18. Значение $b=0,18$, $H=0,59$. Аппроксимация автокорреляционной функции этого ряда выполнена при $\beta = 0,82$.

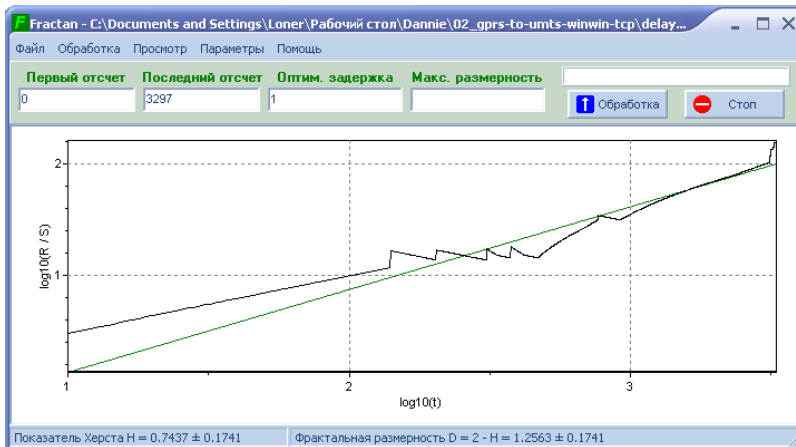


Рисунок 15 - Результат R/S-анализа временного ряда TCP_d256

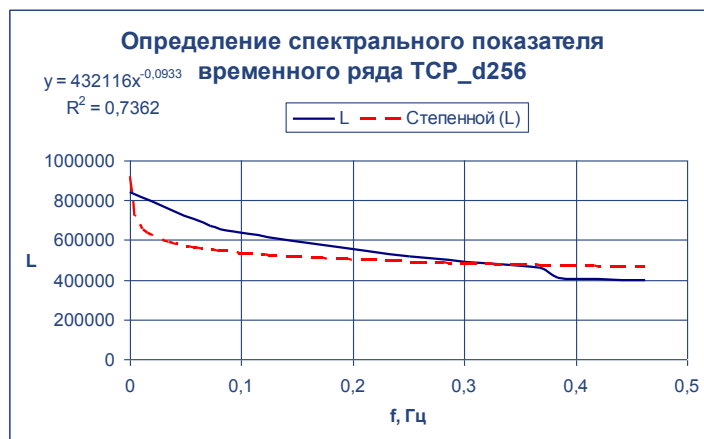


Рисунок 16 - Определение показателя b временного ряда TCP_d256



Рисунок 17 - Результат R/S-анализа временного ряда TCP_d1024

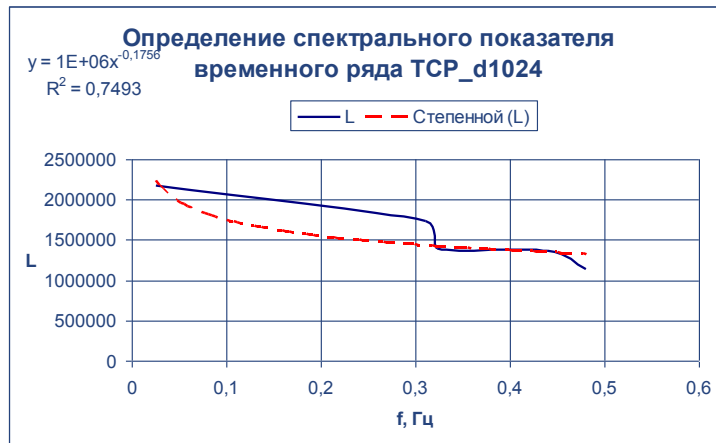


Рисунок 18 - Определение показателя ν временного ряда TCP_d1024

Выводы

В данной работе для реализаций [7] процесса задержки TCP-пакетов выполнен статистический анализ. Предложен метод определения спектрального показателя по графику спектральной функции. Получены следующие результаты: изучаемые временные ряды не подчиняются нормальному распределению, имеют медленно убывающую зависимость АКФ, спектральная плотность временных рядов убывает по степенному закону.

Литература

1. Городецкий А.Я., Заборовский В.С. Информатика. Фрактальные процессы в компьютерных сетях. – СПб.: СПбГТУ, 2000. – 102 с.
2. <http://pi.314159.ru/park1.pdf> - Park K. Self-Similar Network Traffic: An Overview.
3. <http://linkage.rockefeller.edu/wli/reading/taqu96.pdf> - Willinger W., Taqu M.S., Errimilli A. A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks.
4. Иванов А. В. Разработка и исследование алгоритмов прогнозирования и управления очередями в компьютерных сетях. Санкт – Петербург. – 2001. – 18 с.
5. Осин А.В. Влияние самоподобности речевого трафика на качество обслуживания в телекоммуникационных сетях. Автореферат диссертации. Москва. – 2005. – 20 с.
6. Ложковський А.Г. Аналіз і синтез систем розподілу інформації в умовах мультисервісного трафіка. Автореферат диссертации. – Одеса, 2010. – 38 с.
7. <http://www.grid.unina.it/Traffic/Traces/ttraces.php> - Network tools and traffic traces.
8. Петров В.В. Структура телетрафика и алгоритм обеспечения качества обслуживания при влиянии эффекта самоподобия. Автореферат диссертации. Москва. – 2004. – 20 с.
9. Шепухин О.И., Тенякшев А.М., Осин А.В. М. Фрактальные процессы в телекоммуникациях Москва: Радиотехника, - 2003.- 480с.
10. http://freesoft.ru/fractan_44 - Fractan 4.4. Фрактальный анализ временных рядов.

Бельков Д.В., Едемская Е.Н. Анализ трафика сети с беспроводным доступом. Недавние исследования трафика пакетов в сети показали, что это - самоподобный процесс и его Марковские (без последствия) модели приводят к некорректным решениям. Поэтому важной задачей является дальнейшее исследование современного трафика. Данная статья представляет результаты статистического анализа временных рядов задержки в сети с GPRS-доступом на стороне отправителя и UMTS-доступом со стороны получателя.

Ключевые слова: трафик, задержка, джиттер, функция автокорреляции, спектральная плотность, параметр Херста.

Belkov D.V., Edemskaya E.N. The analysis of the traffic traces with the wireless access to the network. Recent network packet traffic investigations have demonstrated, that it is self-similar process and more conventional models, such a simple Markovian (memory-less) models have fault. Analysis the real traffic is an important task. This paper presents the results of the statistical analysis of RTT-delay traces with GPRS-access to the network at sender side and UMTS-access to the network at receiver side.

Keywords: traffic, delay, jitter, autocorrelation function, spectral density, Hurst parameter

Применение логистической регрессии и ROC – анализа при построении скоринговой модели

И.Ю. Анохина

Донецкий национальный технический университет
ingatula@mailru

Анохина И.Ю. Применение логистической регрессии и ROC – анализа при построении скоринговой модели. Рассмотрены вопросы создания скоринговой системы для постановки медицинских диагнозов. При моделировании использовали методы логистической регрессии и ROC – анализа. Описаны этапы построения системы. Проведена проверка точности моделирования на основании статистических данных, полученных при клинических испытаниях.

Ключевые слова: скоринговые системы в медицине, логистическая регрессия, ROC – анализ, пакет Statistica.

Введение

Скоринг (от англ. scoring – подсчет очков в игре) - это модель классификации статистической базы данных по различным группам, если неизвестна характеристика, которая разделяет эти группы, но имеется набор параметров, оказывающих влияние на исследуемую характеристику.

Скоринговые модели и карты получили широкое распространение в различных областях жизнедеятельности человека. Например, в банковской сфере это кредитный скоринг. По социальным характеристикам клиента (пол, возраст, место проживания, должность и т.д.) можно, на основе его анкеты, отнести клиента к группе сильно или слабо соответствующих бизнесу. Исходя из рассчитанного уровня доверия, принимается решение о предоставлении кредита.

В фармацевтике скоринговые модели используются для определения класса препаратов, которые будут эффективны для определенных групп клиентов.

В молекулярной генетике – это молекулярный диагноз заболеваний с учетом того, что генетический код можно использовать для расчета вероятности заболевания.

В современной медицине существует ряд задач, для решения которых требуются количественные, оценочные методы. Поэтому существует необходимость в разработке математических моделей описания процессов, свойств человеческого организма и создания на их базе скоринговых карт или систем, описывающих состояние пациента, степень нарушения функций организма, степень тяжести его состояния.

Скоринговые системы позволяют определять вероятность летального исхода или выздоровления пациента, оценивать степень и стадии заболевания, причем в виде числовых характеристик, что невозможно даже для врача- клинициста с большим опытом. Такая система строится на основе модели, учитывающей симптомы заболевания и само заболевание [1].

Постановка задачи

На основании предоставленных данных анкетирования пациентов, проводимых в медицинских учреждениях г. Донецка, была поставлена задача определить степень риска возникновения заболевания микрохолелитиаза (далее МХЛ, в просторечии, образование желудочных камней или сладжа).

Как показали исследования, существует ряд факторов, влияющих на вероятность образования сладжа. Учитывая, что клинические испытания достаточно дороги, целесообразно на первом этапе исследований выявить группы лиц, предрасположенных к заболеванию, и только в случае высокой вероятности назначать клинические обследования.

Была обследована группа из 131 пациента с установленным заболеванием и контрольная группа обследуемых, у которых диагноз не подтвердился.

Первоначально исследовалась выборка, в которой количество анализируемых факторов было более двадцати. Проведение факторного анализа позволило сократить число исследуемых переменных до четырех. Как показал анализ факторных нагрузок, на увеличение вероятности риска заболевания существенное влияние оказывают: наличие травм, индекс массы тела (здесь и далее ИМТ), уровень физической активности пациента и изменения в режиме питания, включающие в себя соблюдение постов и голодание сроком более семи, а

особенно более четырнадцати дней.

Для решения поставленной задачи выбран метод логистической регрессии.

Основная часть

Логистическая регрессия – разновидность множественной регрессии, позволяющая установить зависимость между несколькими независимыми переменными и зависимой, причем как зависимая, так и/или независимые переменные могут принимать два значения (да/нет), в дальнейшем будем называть их бинарными.

Задачи, в которых используются бинарные коэффициенты, встречаются достаточно часто, например, в качестве бинарной переменной можно рассматривать переменную, характеризующую отсутствие или наличие заболевания у пациентов в медицинских исследованиях; наличие собственности, вкладов в банке при построении скоринговых систем для оценки возможности кредитования; ответы да/нет при обработке результатов анкетирования в социологических опросах и пр. [2].

Рассчитанные в результате логистического анализа величины переменных в большинстве случаев принимают не точные значения 0 или 1, а значения в этом интервале. В этом случае полученные в ходе анализа результаты интерпретируют не как конкретные значения, а как вероятность P того, что результат может быть отнесен к определенному классу. Например, вероятность близкая к единице, позволит сделать вывод, что пациент находится в зоне риска (вероятнее болен, чем здоров) и для уточнения диагноза необходимы клинические исследования [3].

В табл.1 представлена часть исходных данных, на основании которых строилась модель.

Таблица 1. Исходные данные для модели

№	А Сладж	В Травмы	С Физическая активность	Д Посты, голодание	Е ИМТ
1	да	да	да	да	2
2	да	нет	да	да	1
3	нет	нет	да	нет	0
4	нет	нет	да	нет	1
5	нет	нет	да	нет	0
6	да	да	нет	да	0
7	нет	нет	нет	нет	0
8	да	да	нет	да	2
9	да	да	нет	да	3
10	да	да	да	да	2
11	да	да	нет	нет	3
12	да	да	нет	да	1
13	нет	нет	да	да	1
14	нет	нет	нет	нет	0
15	да	нет	нет	да	1
16	да	нет	нет	да	1
17	да	да	нет	нет	1
18	да	нет	нет	да	1
19	нет	нет	да	нет	0
20	да	да	нет	нет	1

Показанные в табл.1 данные первоначально были размещены в Microsoft Excel. Поэтому первая строка таблицы отображает заголовки столбцов Excel, а во второй приводятся названия столбцов таблицы исходных данных. В столбце А указан установленный диагноз (да – подтверждено наличие заболевания, 0 – нет); в столбце В – наличие ранее травм; в столбцах С и Д – данные по физической активности и резким изменениям в режиме питания; ИМТ был классифицирован следующим образом: 0 – нормальное телосложение; 1 – первая стадия ожирения, 2 и 3 – соответственно вторые и третьи стадии.

Для обработки данных использовали пакет Statistica, модуль Nonlinear Estimate, в котором выбрана категория - **Logistic Regression**.

Для данных табл. 1 в качестве y рассматривали наличие заболевания ($y=1$ соответствует «сладж=да» и 0 в противном случае), в качестве независимых переменных значения в столбцах (x_1 характеризует наличие травм у пациента в анамнезе; x_2 наличие или отсутствие физической активности; x_3 – зафиксированные изменения в режимах питания; x_4 – ИМТ).

При построении логистической модели определяются параметры уравнения вида

$$y = \frac{e^{b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_n \cdot x_n}}{1 + e^{b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_n \cdot x_n}} \quad (1)$$

где b_0, b_1, \dots, b_n - коэффициенты логистической регрессии, n - количество независимых переменных.

Исходя из формулы (1) можно определить вероятность наступления некоторого события, в нашем случае это вероятность подтверждения диагноза, по формуле:

$$P = \frac{1}{1 + e^{-(b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_n \cdot x_n)}}$$

Для решения задач логистической регрессии в пакете Statistica используется метод максимального правдоподобия.

В результате расчетов получены параметры модели:

Таблица 2. Коэффициенты логистической регрессии

	Const.b ₀	b ₁	b ₂	b ₃	b ₄
Оценка	-1.068	3.361	-3.054	0.669	2.673

На рис. 1 представлено результирующее окно расчетов параметров. Кроме значений коэффициентов логистической регрессии, которые отображаются в строке **Estimate**, в окне отображения результатов указан p -уровень гипотезы. Если p -уровень менее 5%, модель считается значимой. В нашем случае $p=0.00018$, что позволило сделать вывод о значимости разработанной модели.

Оценивая величину хи-квадрата Пирсона ($\text{Chi}(4)=22.248$ на рис. 1) и сравнивая полученное значение критерия хи-квадрат с критическим, получаем $22.248 > 9.488$, откуда делаем вывод о наличии статистически значимой зависимости параметра Y от выбранных независимых переменных X . Уровень значимости данной взаимосвязи соответствует $p < 0.05$.

N=32	Const.B0	VAR2	VAR3	VAR4	VAR5
Estimate	-1.06822	3.361455	-3.05407	.668872	2.673268

Рисунок 1 – Параметры модели логистической регрессии

Еще одним важным шагом в проверке качества построенной модели является оценка параметра *Отношение несогласия*. На рис. 2 показана таблица с числом наблюдений, которые были правильно или неправильно классифицированы в соответствии с полученной моделью [4]. В данной статье рассматривается подвыборка, состоящая из 22 человек, которые классифицированы как больные, и десяти человек – условно здоровые.

Observed	Pred. 0	Pred. 1	Percent Correct
0	9	1	90.000
1	3	19	86.364

Рисунок 2 – Принадлежность наблюдений к классам

Все наблюдения с предсказанными значениями (вероятностью) меньше или равными 0.5 классифицируются в пакете Statistica как неудача – «Failure», остальные, с предсказываемыми значениями больше 0.5, классифицируются как успех – «Success». *Отношение несогласия* вычисляется как отношение произведения чисел правильно классифицированных наблюдений к произведению чисел неправильно классифицированных результатов [5].

В столбце *Percent Correct* приведены проценты совпадений исходных и прогнозируемых данных. Так, в строке, соответствующей $\text{Observed}=0$ (по нашей классификации сладж не обнаружен), рассматривается процент правильной классификации здоровых пациентов. Следовательно, в 90 случаях из 100 здоровый пациент

идентифицируется именно как здоровый, и в 10 случаях из 100 он будет считаться ошибочно больным. Это обусловлено тем, что даже у здорового человека существует определенный процент риска, обусловленный его возрастом, полом, массой тела, наследственной предрасположенностью, наличием серьезных травм, но далеко не все пациенты страдают этим заболеванием.

Во второй строке, соответствующей Observed=1 (подтверждено наличие сладжа), выводится процент правильно идентифицированных больных. Более, чем в 86 случаях из 100 модель позволяет предсказать наличие заболевания и менее, чем в 14 случаях больной пациент будет считаться здоровым (значение Percent Correct=86.384).

Проведенные исследования и сопоставление полученных статистических данных и прогноза позволили считать возможным использование разработанной модели для начальной, доклинической диагностики.

Подставляя полученные коэффициенты логистической регрессии (см. табл. 2) в уравнение (1) можно получить прогнозируемые значения величины $Y_{теор}$. Отметим, что если в исходном наборе данных зависимая переменная Y принимает значения 0 или 1 (да/нет), то результатом расчетов по полученной зависимости является массив данных в интервале от 0 до 1.

Например, для приводимого в табл.1. набора данных («нет» кодировалось как ноль, «да» – как единица), теоретические значения Y имели вид, показанный в табл. 3. В таблице приводятся первые десять данных выборки.

Таблица 3. Коэффициенты логистической регрессии

№	1	2	3	4	5	6	7	8	9	10
$Y_{теор}$	0.995	0.314	0.016	0.190	0.016	0.951	0.256	1.000	1.000	0.995
Y	да	да	нет	нет	нет	да	нет	да	да	да
	1	1	0	0	0	1	0	1	1	1

В табл. 3 в строке $Y_{теор}$ показана предсказанная вероятность того, что пациента можно отнести к классу больных, через Y обозначены реальные данные как в виде варианта да/нет, так и их бинарные значения.

Как видим, далеко не все значения во второй строке таблицы равны нулю или единице. В большинстве случаев значения колеблются в указанном интервале. В связи с этим возникает вопрос, следует ли считать больным, например, пациента под номером 7 с рассчитанной вероятностью 0.256 или пациента под номером 2 ($P=0.314$).

Таким образом, при расчете значения $Y_{теор}$ по уравнению логистической регрессии (1), возникает проблема, каким образом должны быть преобразованы полученных числовые значения $Y_{теор}$ в бинарные классификаторы. Следовательно, необходимо определить такое пороговое значение P , при котором выполняется система неравенств:

$$\begin{aligned} y &\geq P_{порог} \rightarrow \text{пациент болен} \\ y &\leq P_{порог} \rightarrow \text{пациент здоров} \end{aligned} \quad (2)$$

Т.е. возникает вопрос, как определить пороговое значение вероятности, которое разделяет исследуемое множество на два класса: здоров/болен; кредитоспособен/ не кредитоспособен, эффективно лекарство для группы пациентов или нет.

Варьируя значение порогового P , каждый раз при расчетах будут получаться новые значения бинарного классификатора. Например, если в качестве порогового значения выбрать величину 0.5, то при округлении $Y_{теор}$ для второго пациента (табл. 3) получим прогноз «пациент здоров», в то время, как известно обратное.

Для решения этой проблемы использовали аппарат ROC(Receiver Operating Characteristic) – анализа.

ROC – анализ долгое время использовали в теории обработки сигналов и радиолокации для описания события правильного обнаружения сигнала и исключения ложных срабатываний. В настоящее время ROC – анализ нашел широкое применение не только в задачах медицинской диагностики, но и в банковском, кредитном скорринге, теории принятия решений.

ROC-кривая показывает зависимость количества верно классифицированных наблюдений со значением параметра, равным нулю, от количества неверно классифицированных наблюдений со значением параметра, равным единице. При этом предполагается, что у классификатора имеется параметр, варьируя который, будет происходить разделение на два класса. Этот параметр часто называют порогом, или точкой отсечения (cut-off value). В зависимости от него будут получаться различные значения *ошибок I и II рода*.

В логистической регрессии порог отсечения изменяется от 0 до 1. В нашем случае, это то значение вероятности $P_{порог}$, которое следует использовать при разделении пациентов на больных и здоровых, формула (2).

На основании данных, приведенных на рис. 2, можно определить значения, необходимые для проведения ROC – анализа, а именно:

TP (*True Positives*) – верно классифицированные случаи с заболеванием, для рассматриваемого примера TP=19;

TN (*True Negatives*) – верно классифицированные случаи с отсутствием заболевания, для рассматриваемого примера TN=9;

FN (*False Negatives*) – случаи с заболеванием, но при прогнозировании оно выявлено не было (ошибка I рода), так называемый «ложный пропуск» – болезнь не была обнаружена, для рассматриваемого примера FN=3;

FP (*False Positives*) – случаи, когда было вынесено решение о наличии заболевания, хотя его не было (ошибка II рода). Это ложное обнаружение, т.к. при отсутствии события ошибочно выносится решение о его наличии, для рассматриваемого примера FP=1.

Для построения ROC-кривых необходимо вычислить два параметра: чувствительность (*Sensitivity*) и специфичность (*Specificity*).

Чувствительность определяется по формуле:

$$SE = \frac{TP}{TP + FN} \cdot 100\% \quad (3)$$

Специфичность считают, как:

$$SP = \frac{TN}{TN + FP} \cdot 100\% \quad (4)$$

Эти два параметра используются для оценки требуемого уровня точности решения поставленной задачи. С медицинской точки зрения существуют следующие правила:

– если ставится задача максимального предотвращения пропуска больных, то должен быть выбран высокий уровень чувствительности SE (гипердиагностика);

– если же лечение связано с серьезными побочными эффектами, то гипердиагностика не допустима и в этом случае следует использовать более высокие уровни специфичности SP [6].

В табл.4. показаны рассчитанные значения перечисленных выше параметров TP, TN, FN, FP и величин чувствительности и специфичности, которые будут использованы при построении ROC – кривой. Параметры определялись для каждого значения порога отсечения $P_{\text{порог}}$ в пределах от 0 до 1 с выбранным шагом 0.1.

Кроме этого определяли величину FPR – процент ложно обнаруженных случаев заболевания, где

$$FPR = \frac{FP}{TN + FP} \cdot 100\% . \quad (5)$$

При построении ROC-кривой по оси Y откладывают чувствительность SE, по оси X – значения параметра FPR.

Таблица 4. Параметры для построения ROC - кривой

№	$P_{\text{порог}}$	TP	TN	FN	FP	SE	SP	FPR
1	0.1	22	3	0	7	100%	30%	70%
2	0.2	22	4	0	6	100%	40%	60%
3	0.3	21	6	1	4	95%	60%	40%
4	0.4	20	8	2	2	91%	80%	20%
5	0.5	19	9	3	1	86%	90%	10%
6	0.6	19	9	3	1	86%	90%	10%
7	0.7	19	9	3	1	86%	90%	10%
8	0.8	19	9	3	1	86%	90%	10%
9	0.9	17	9	5	1	77%	90%	10%
10	1	0	10	22	0	0%	100%	0%

На рис. 3 показана построенная ROC – кривая.

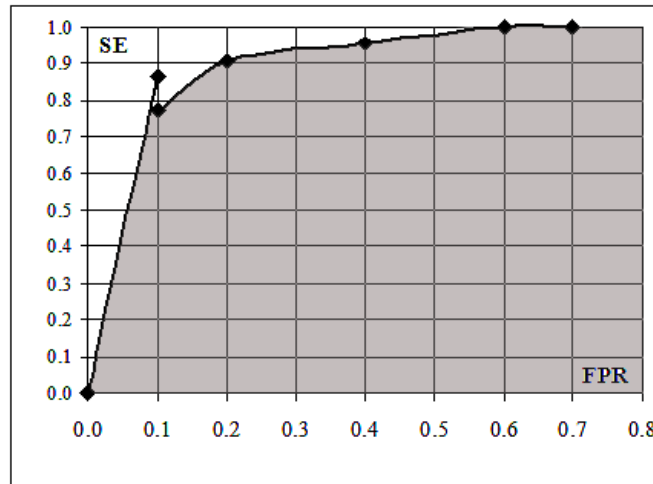


Рисунок 3 – ROC - кривая

Идеальная ROC-кривая должна проходить через верхний левый угол, в котором процент правильно идентифицированных случаев составляет 100% (идеальная чувствительность), а процент ошибки равен нулю. Отсюда следует, что чем ближе ROC - кривая к верхнему левому углу, тем выше уровень качества прогнозирования разработанной модели. И наоборот, чем меньше изгиб кривой и чем ближе она расположена к диагональной прямой (прямая линия под углом 45°, соединяющая левый нижний и правый верхний углы), тем менее эффективна модель.

Близость ROC-кривой к диагональной линии соответствует так называемую «бесполезному» классификатору, т.е. полной неразличимости классов [7,8].

Как видно из данных табл. 4 и рис. 3, значения рассматриваемых параметров SE, SP, FPR не меняются для порога, находящегося в пределах

$$0.5 \leq P_{\text{порог}} \leq 0.8. \quad (6)$$

Соответствующие строки выделены в табл. 4. Это означает, что пороговые значения могут быть любым числом в указанном интервале.

Неоднозначность полученного ответа обусловила необходимость в дополнительных исследованиях с целью уточнения порогового значения.

Визуальная оценка кривых ROC не всегда позволяет выявить наиболее целесообразное значение порога. Часто для оценки качества ROC-кривых используют оценку площади под кривыми (на рис. 3 закрашена серым цветом). Эта оценка может быть получена непосредственно вычислением площади под многогранником, ограниченным справа и снизу осями координат и слева сверху – экспериментально полученными точками (рис. 3). Численный показатель площади под кривой называется AUC (Area Under Curve).

В литературе приводится экспертная шкала для значений AUC, по которой можно судить о качестве разработанной модели [9,10], см. табл. 5.

Таблица 5. Интервалы площади под ROC-кривой и оценка качества модели

Интервал AUC	Качество модели
0.9÷1.0	Отличное
0.8÷0.9	Очень хорошее
0.7÷0.8	Хорошее
0.6÷0.7	Среднее
0.5÷0.6	Неудовлетворительное

Нами было вычислено значение AUC, величина составила приблизительно 83%, что позволяет сделать вывод об «очень хорошем» качестве модели.

В качестве дополнительной проверки качества модели и уточнения интервала порогового значения определяли отношение суммы правильно идентифицированных случаев (как при отсутствии болезни, так и в случае ее наличия) к общему числу имеющихся экспериментальных данных. Данный параметр определили, как переменную $F = (TP+TN)/N$.

Поставлена и решена задача подбора параметров аппроксимирующей функции, описывающей зависимость числа правильно идентифицированных случаев от величины порога. Для решения задачи использовали модуль Nonlinear Estimate пакета Statistica. Проведен подбор коэффициентов полиномиального

уравнения. Как показала оценка погрешности подбора, полином пятой степени описывает данные с наименьшей погрешностью [11].

На рис. 4 показано результирующее окно с приводимыми величинами параметров (отмечено цифрой 1 в левом углу рисунка) и окно расчетов параметров пакета Statistica (цифра 2 рисунка).

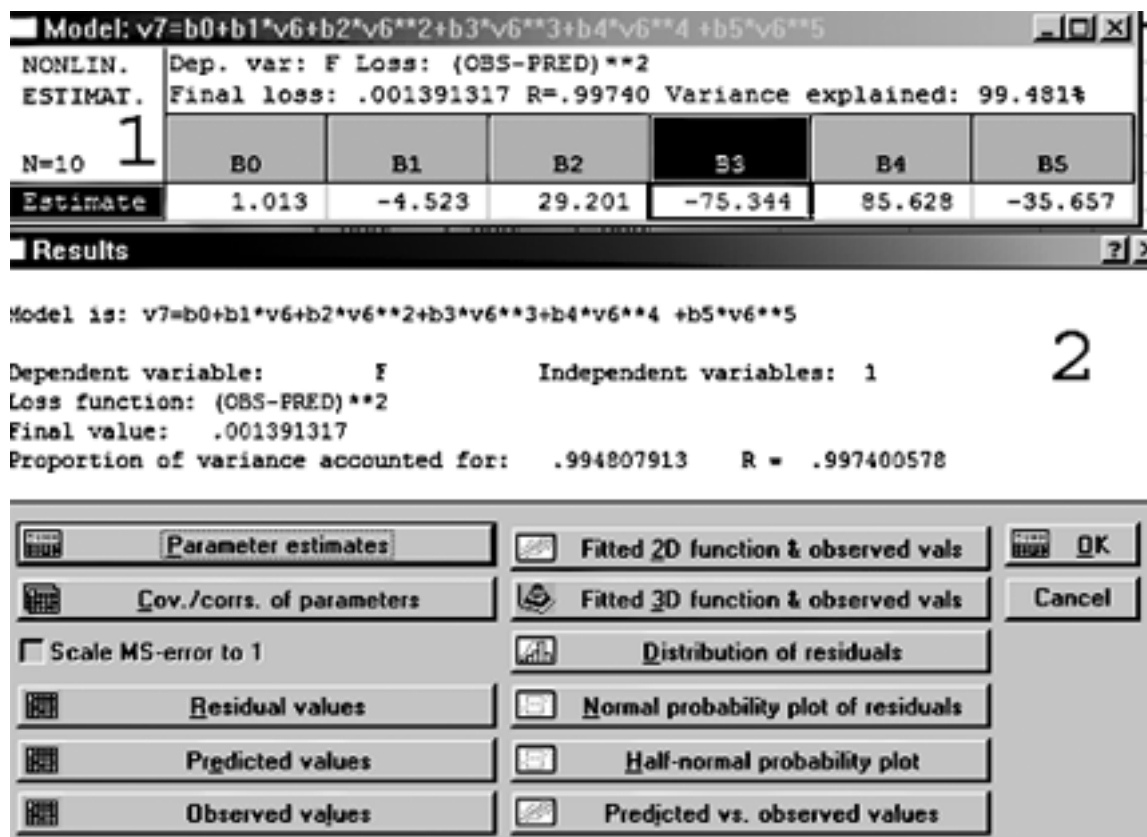


Рисунок 4 – Расчет параметров полиномиальной функции в пакете Statistica

Зависимость переменной F от величины порога можно описать уравнением вида:

$$F = 1.013 - 4.523 \cdot P_{\text{порог}} + 29.201 \cdot P_{\text{порог}}^2 - 75.344 \cdot P_{\text{порог}}^3 + 85.628 \cdot P_{\text{порог}}^4 - 35.657 \cdot P_{\text{порог}}^5 \quad (7)$$

Дифференцируя функцию F, находим точку экстремума. Как показали расчеты, при значении порога $P_{\text{порог}}=0.73$ имеет место максимальное количество правильно диагностированных случаев, равное 92.07%. Это значение является уточненным значением порога, см. формулу (6).

Таким образом, алгоритм идентификации пациентов свелся к нескольким элементарным шагам:

- 1 рассчитать по уравнению логистической регрессии (формула 1) значения $Y_{\text{теор}}$;
- 2 сравнить их с пороговым значением (в нашем примере 0.73);
- 3 если $Y_{\text{теор}} > 0.73$ можно сделать вывод о высокой вероятности заболевания и необходимости клинического диагностирования.

Выводы

В работе рассмотрены вопросы разработки скоринговой системы, позволяющей по предварительному анкетированию пациентов без дополнительной клинической диагностики оценивать степени возможного риска заболевания. Выявлены факторы, оказывающее влияние на вероятность появления заболевания. Проведена оценка качества модели. Точность проведенных расчетов более 92%, что позволяет рекомендовать предложенную методику для составления скоринг - карт пациентов.

Литература

1. Мильчаков К.С., Шебалков М.П. Скоринговые карты в медицине: обзор и анализ публикаций. // Врач и информационные технологии, №1, 2015. - с.71-79.
2. Методы статистической обработки медицинских данных: Методические рекомендации для ординаторов и аспирантов медицинских учебных заведений, научных работников. / сост.: Кочетов А.Г., Лянг

- О.В., Масенко В.П., Жиров И.В., Наконечников С.Н., Терещенко С.Н. – М.: РКНПК, 2012. – 42 с.
3. Леонов В. Логистическая регрессия в медицине и биологии / Биометрика. Журнал для медиков и биологов, сторонников доказательной биомедицины. - Режим доступа: http://www.biometrika.tomsk.ru/logit_1.htm.
 4. Бэстенс Д.-Э., Ван Ден Берг В.-М., Вуд Д. Нейронные сети и финансовые рынки. Принятие решений в торговых операциях. - М.: Научное издательство ТВП, 1997. - 235с.
 5. Логистическая регрессия / Портал Знаний StatSoft. - Режим доступа: <http://statistica.ru/theory/logisticheskaya-regressiya/>.
 6. Логистическая регрессия и ROC-анализ - математический аппарат / Base Group Lab. - Режим доступа: <https://basegroup.ru/community/articles/logistic>.
 7. Бююль А., Цёфель П. SPSS: Искусство обработки информации. Анализ статистических данных и восстановление скрытых закономерностей: - Пер. с нем. - СПб.: ДиаСофтЮП, 2005. - 608 с.
 8. Логистическая регрессия. / Центр статистического анализа. - Режим доступа: <http://statmethods.ru/konsalting/statistics-metody/116-logisticheskaya-regressiya.html>.
 9. Цыплаков А.А. Некоторые эконометрические методы. Метод максимального правдоподобия в эконометрии. Методическое пособие. - Новосибирск: НГУ, 1997. - 129 с.
 10. Сергиенко В.И., Бондарева И.Б. Математическая статистика в клинических исследованиях. - М.: ГЕОТАРДМЕД, 2001. - 256 с.
 11. Боровиков В.П., Боровиков И.П. Statistica. Статистический анализ и обработка данных в среде Windows. -М.: Информационно-издательский дом "Филинь", 1997.- 608с.

Анохина И.Ю. Применение логистической регрессии и ROC – анализа при построении скоринговой модели. Рассмотрены вопросы создания скоринговой системы для постановки медицинских диагнозов. При моделировании использовали методы логистической регрессии и ROC – анализа. Описаны этапы построения системы. Проведена проверка точности моделирования на основании статистических данных, полученных при клинических испытаниях.

Ключевые слова: скоринговые системы в медицине, логистическая регрессия, ROC – анализ, пакет Statistica.

Anokhina I.Y. The use of logistic regression and ROC - analysis of the construction of the scoring model. The problems of creating a scoring system for medical diagnosis. The simulation methods used logistic regression and ROC - analysis. Stages of construction of the system. Check of the accuracy of the simulation on the basis of statistical data obtained from clinical trials.

Keywords: Scoring systems in medicine, logistic regression, ROC - analysis, Statistica.

Математическое прогнозирование каскадных аварий электросетей

¹Кузнецов П.А., ¹Юдин А.В., ²Масло С.В., ¹Клюковкин В.Р.
¹Рыбинский государственный авиационный технический университет
им. П.А. Соловьева
Кафедра электротехники и промышленной энергетики
²Донецкий национальный технический университет
Кафедра прикладной математики
E-mail: kuznetsovpael@inbox.ru

Кузнецов П.А., Юдин А.В., Масло С.В., Клюковкин В.Р. Математическое прогнозирование каскадных аварий электросетей. В статье описана проблематика таких явлений как веерные отключения и каскадные аварии, происходящие в электрических сетях. Проведен анализ веерных отключений на двух магистралях по имеющимся данным и выбрана модель для углубленного анализа законов их распространения. Показана тенденция к увеличению их количества. Составлен алгоритм протекания стандартного веерного отключения, причиной которого является единичное инициирующее событие. Проведен анализ потерь мощности в двух магистралях за длительные временные рамки методом скользящего окна. Представлены зависимости и подчинение каскадной аварии степенным законам.

Ключевые слова: Веерное отключение, математическая модель, анализ аварий, Minitab, статистический анализ, авария в электросети.

Введение

Развитие современной промышленности и рост численности населения земли приводит к тому, что с каждым днём в мире потребляется все большее количество электроэнергии. Многие сети, которые используются на данный момент для снабжения потребителей различных категорий, уже не справляются с объемом нагрузки, который на них возлагается, и требуют ремонта и модернизации. Так по данным [1] износ основных производственных и технологических активов отечественных электросетей оценивается в два раза выше, чем за рубежом. В связи с этим, все чаще как мировые, так и отечественные электросети страдают от каскадных аварий (веерных отключений), которые приводят не только к браку и материальным убыткам, но, даже к человеческим жертвам. Разработка методов и устройств диагностирования, предсказания развития аварий и их нейтрализация становятся все более актуальными задачами [2]. Основными причинами аварий в электрических сетях могут быть природные катаклизмы, сильные порывы ветра, человеческие или технические факторы, трудовые конфликты, терроризм, военные действия и т.д [3, 4]. А очень многие аварии могут быть вызваны комбинацией электрических, компьютерных и человеческих ошибок и развиваться далее в лавинообразное веерное отключение, которые имеют огромное влияние для потребителей [5].

Цели и задачи работы

Основной задачей данной работы является математический анализ веерных отключений, происходящих в электрических сетях, на основе собранной статистики по потерям мощности. Помимо этого, часть исследования посвящена поиску первоисточника каскадных аварий и возможным методам их устранения.

Анализ последних публикаций и исследований

Технические условия, которые могут приводить к возникновению аварийных ситуаций - это перегрузки, превышения крайних пределов уровня напряжений или частоты, нестабильность системы, отключение подстанций или источников генерации [5]. В трудах по изучению свойств электрических систем структурная сложность и высокая взаимная зависимость между компонентами системы усложняют изучение их поведения. Взаимная зависимость – это главная причина уязвимости электрических сетей и относится к риску возникновения огромного веерного отключения.

Уязвимость системы объясняется чувствительностью системы к угрозам и опасностям [5]. Очевидно, что из-за наличия внутренних взаимосвязей, локальные перегрузки линии или выход локальных компонентов из

стройка могут привести к каскадному нарушению функционирования текущих узлов всей системы, что в дальнейшем приведет к её отключению. Терминологически этот процесс называется веерным отключением. Веерные отключения могут быть вызваны неспособностью электрической системы к адаптации изменению нагрузки или вследствие чрезмерного потребления электрического тока потребителем в размере, превышающем емкость сети.

Многие исследователи руководствовались поведением электрических сетей и возникновением ошибок в них в процессе передачи энергии. Сети состоят из генераторов, трансформаторов, линий электропередачи и электростанций [6]. Авария в этих системах означает, что, когда случается неисправность, по крайней мере, с одним из этих компонентов или внутри определенного элемента. Неисправность в компоненте системы может привести к серьезному веерному отключению, порождающему серьезный кризис в обществе. Однако, большинство крупнейших веерных отключений и лавинообразных аварий не были внезапными. Каскадные аварии – это главный механизм, согласно которому развиваются крупные веерные отключения. Таким образом, это – цепочка начальных явлений, которыми электрическая сеть вяло сигнализирует о реальном состоянии всех соседних звеньев [7, 8 - 10].

Каскадная неисправность определяется как последовательность зависимых неисправностей отдельных компонентов, которая постепенно ослабляет электрическую систему. Эта неисправность охватывает физические элементы системы, программное обеспечение, исполняющие функции, обслуживающий персонал и организации, которые разрабатывают, управляют и регулируют электрическую систему [11]. Каскадные отключения могут являться следствием состояния элементов системы, таких как вышедших из строя, или зависеть от характеристик передачи мощности, а также последствием ручных или автоматических команд [12]. Благодаря осложнениям полный перебор всех вариаций каскадных отключений не представляется возможным.

Выбор математической модели

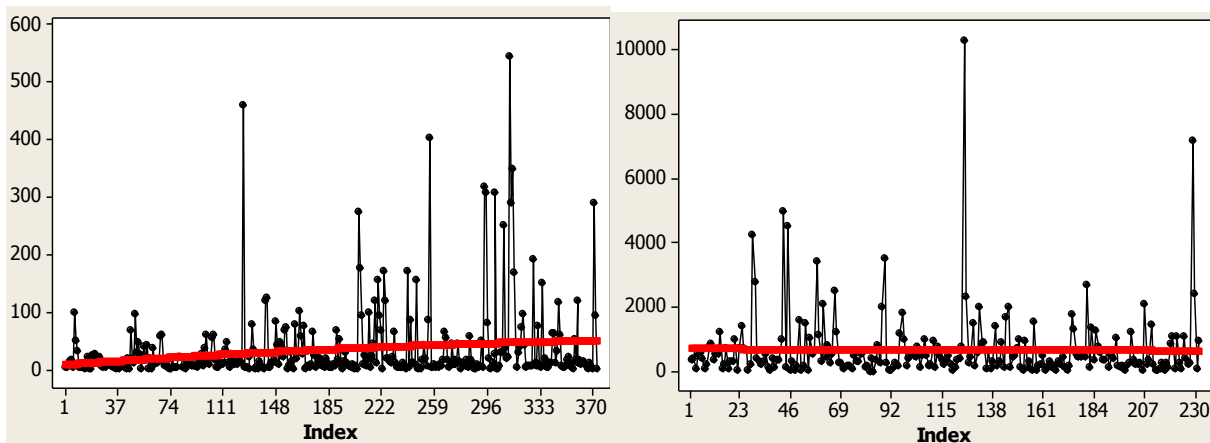
Веерные отключения могут быть вызваны неспособностью электрической системы к адаптации изменению нагрузки или вследствие чрезмерного потребления электрического тока потребителем в размере, превышающем емкость сети. В настоящее время существуют три основные концепции изучения веерного отключения в технической и научной литературе [8]. Первое учение описывает сеть и взаимодействие между её компонентами во времени, где взаимодействие выражает равенство и неравенство алгебраических систем с ограничениями в нелинейных дифференциальных уравнениях. Второе учение рассматривает поведение сети в устойчивом состоянии введением случайных флуктуаций при потреблении нагрузки, а третья концепция полагает, что электрическая сеть – это пример сложной системы, чье поведение глубоко зависит от её топологических свойств.

Модель сети на основе дифференциальных уравнений дает наиболее точное представление о физических процессах, происходящих в ней. Однако, решение, а зачастую даже и составление системы очень трудоемко, а, иногда даже и не представляется возможным. Третья модель не учитывает коммутационные процессы, постоянно влияющие на сеть. Поэтому для проведения расчетов была выбрана именно вторая теория.

Вычислительный эксперимент

Т.к. каждая авария сопровождается потерями мощности, то для возможности анализа веерных отключений по имеющимся данным, было принято следующее упрощение. На основании второй теории о функционировании электрической сети она всегда находится в равновесном состоянии. Если в один момент времени мощность, отданная генератором (P_r) и полученная потребителем (P_n) равны, то аварии не происходит. Если же данные о генерации и потреблении мощности разнятся, то мы получаем потери (ΔP), которые представляют собой аварию.

Поскольку выбранная модель сети представляет исследуемый объект всегда находящимся в статическом состоянии и выходящим из равновесия только при коммутационных процессах, то это отразится и на описании модели аварии. Веерные отключения и каскадные аварии становятся все более частыми и разрушающими. Для подтверждения этой теории были отобраны данные [7] по двум магистралям – №1 (370 случаев) и №2 (230 случаев). Данные первой магистрали соответствуют периоду с 1990 по 2003 гг, а данные второй - с 1980 по 2005 гг. Анализ проводился в программе Minitab. На рис. 1 представлены результаты расчетов по данным потерь мощности. Ось абсцисс – это порядковый номер аварии. Они задокументированы и внесены в анализ в хронологической последовательности. По оси ординат расположены потери мощности, которые случились в результате каскадной аварии. Для прогнозирования дальнейших вероятностей возникновения аварий были аппроксимированы линии тренда. Для данных из первой области кривая тренда соответствует степенному закону, описываемому формулой $8,2 + 0,175 \cdot x - 0,000169 \cdot x^2$. Из графика четко видно, что вероятность возникновения аварии в сети растет во времени по степенному закону и в будущем отключения будут происходить все чаще.



а)

б)

Рисунок 1 – Анализ тенденции протекания веерных отключений: а) для данных по магистрали №1; б) для данных по магистрали №2

Анализ американских данных показывает несколько иную картину. Линия тренда также подчиняется степенному закону, описываемому формулой $701 - 0,34 \cdot x + 0,0004 \cdot x^2$. Примечательно, что согласно данной кривой вероятность аварий будет снижаться. Это противоречие объясняется тем, что после масштабных аварий 1997-2001 годов, были приняты меры по модернизации сети и реконструкции ее мощностей.

Составление алгоритма веерного отключения

Не смотря на значимость этого события и разрушающие последствия его протекания, исследованию каскадных аварий и веерных отключений не уделяется много внимания. До недавнего времени вообще считалось, что основной причиной масштабных поломок электросетей являются только природные бедствия. Оказывается, что это не так. Многие исследователи ссылаются на то, что практически все масштабные аварии, которые происходили в последние десятилетия, начинались с какой-либо незначительной причины – так называемого инициирующего события. Оно сопоставимо с первым камнем, который тянет за собой лавину.

Анализ [5-7] показывает, что большинство веерных отключений, как локального, так и национального характера были следствием какой-либо единичной поломки. К примеру, веерное отключение 2003 года между США и Канадой, которое вывело из строя практически 11% нагрузки североамериканской сети и оставило без электричества более 50 млн человек, было вызвано изначально небольшой рассинхронизацией генераторов модуля 5 электростанции Eastlake. Она была вызвана наличием большого количества реактивной мощности в сети. Персонал не отреагировал вовремя и должным образом на показания системы автоматики, что и привело к каскадному отключению электроснабжения [8].

Исходя из этого, модель веерного отключения можно представить в виде алгоритма, изображенного на рис. 2. Как видно из диаграммы, в любом веерном отключении есть точка невозврата, после которого сеть будет неконтролируемо затухать, а есть и место для своевременной реакции и восстановления системы в номинальное состояние.

Очень часто эти первые сигналы не замечаются обслуживающим персоналом и даже игнорируются автоматикой. Однако, не каждое такое инициирующее событие может стать причиной веерного отключения. Возможно, что оно произойдет на локальном уровне и не вызовет цепную реакцию. Предсказание влияния таких аварий на целостность сети является важной задачей.

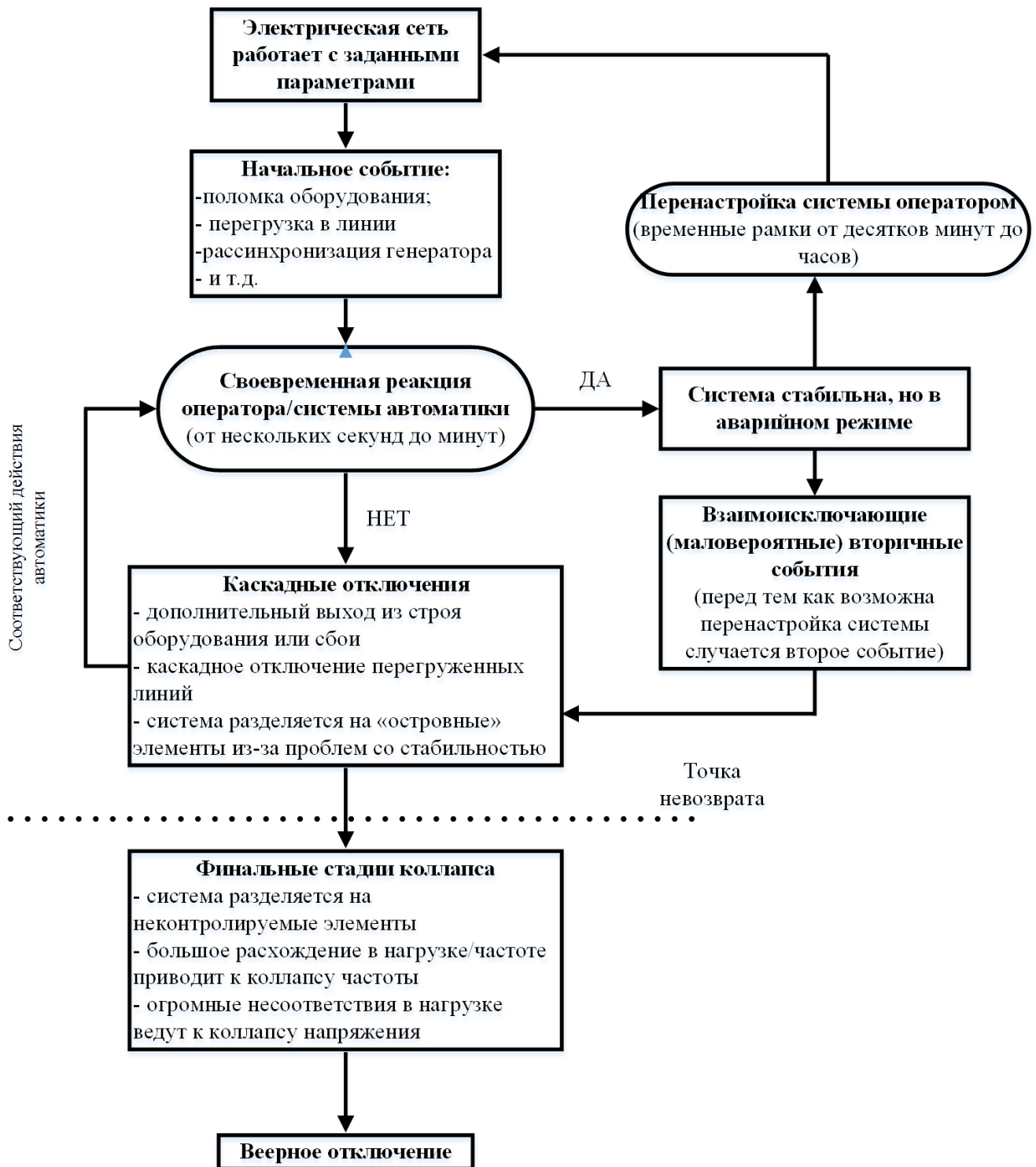


Рисунок 2 – Алгоритм протекания веерного отключения

Если течение каскадных аварий подчиняется степенным законам, то это должно подтверждаться реальными данными. Проверим это, проанализировав данные методом скользящего окна (МСО). Каждый набор данных был рассмотрен как группа временных отрезков X_t и детальное теоретическое статистическое отклонение было получено из [7].

$$X = \{X_t \ t = 1, 2, 3, \dots, n\} \quad (1)$$

Броуновское движение может быть получено из (1):

$$Y = \{Y_t \ t = 1, 2, 3, \dots, n\} \quad (2)$$

Где Y_t – это первообразная интеграла по времени X_t и записана в виде

$$Y_t = \sum_{i=1}^t X_i \quad (3)$$

Новый ряд $Y^{(m)}$ потом получается для ряда Y и $m=1, 2, 3, \dots, n$.
Отсюда

$$Y^{(m)} = \{Y_u^m, u = 1, 2, 3, \dots, n/m\} \quad (4)$$

Элементы этого ряда – это блоки, содержащие m элементов начального ряда Y . Это значит, что

$$Y_u^{(m)} = \{Y_{mu-m+1}, \dots, Y_{um}\} \quad (5)$$

Стандартное отклонение $\sigma_u^{(m)}$ в пределах каждого n/m блока и содержащее m элементов определяется как

$$\sigma_m^{(u)} = \sqrt{\sum_{m=1}^n \sum_{u=1}^{n/m} \frac{((Y_{mu-m+1}, \dots, Y_{um}) - \mu_m^{(u)})^2}{m}} \quad (6)$$

Где $\mu_m^{(u)}$ – это значение Броуновского движения для ряда из n/m блоков и m элементов. После всего этого среднее значение стандартного отклонения n/m блоков рассчитывается по формуле

$$\sigma_m = \frac{\sum_{u=1}^{n/m} \sigma_m^{(u)}}{n/m} \quad (7)$$

Таким образом, временной ряд X с определенной функцией автокорреляции может быть представлен как функция σ_m , имеющая фазу и шкалу степенного закона. Это - $\sigma_m \propto m^H$, где H – это экспонента Хёрста, определенная ранее. Или

$$\log(\sigma_m) = H \log(m) \quad (8)$$

Согласно [7], для $0,5 < H < 1$, ряд X_t имеет длительную временную зависимость. А для $0 < H < 0,5$ ряд имеет длительную временную антикорреляцию. Если же $H=1,0$, то ряд X_t детерминистский или выход результата предсказуем в то время как он некоррелирован, если $H=0,5$. Точный график расчета этой таблицы показан на рис. 3, 4 используя метод скользящего окна. Для второго случая результат показывает, что $H \approx 1,0$, что резко отражает наличие сильной корреляции или предсказуемой зависимости между последовательными верными отклонениями. Для первой сети точно так же прослеживается наличие длительной временной корреляции.

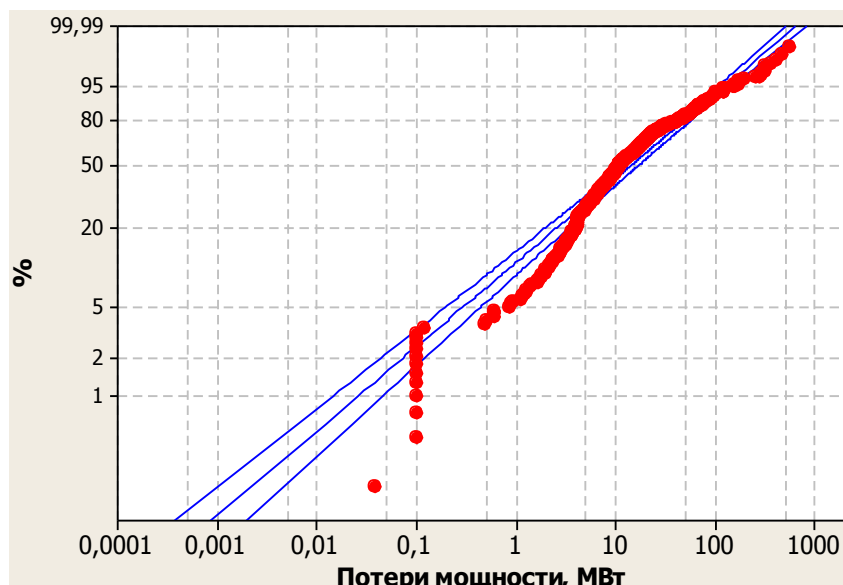


Рисунок 3 - Анализ веерных отключений первой магистрали МСО

На рис. 4 показан точечный график анализа потерь мощности во второй магистрали методом скользящего окна. Кривая представляет собой стандартное отклонение от временной зависимости для последовательных временных рядов веерных отключений. Было обнаружено, что стандартное отклонение может быть нормировано как степенной закон $\sigma_m \propto m^{0,93}$, где m представляет временное запаздывание последовательных временных рядов веерных отключений. Число $0,93$ – это приблизительное значение H . И, т.к. H меньше единицы, но более половины, то это значит, что существует длительная временная корреляция или зависимость между временными рядами веерных отключений.

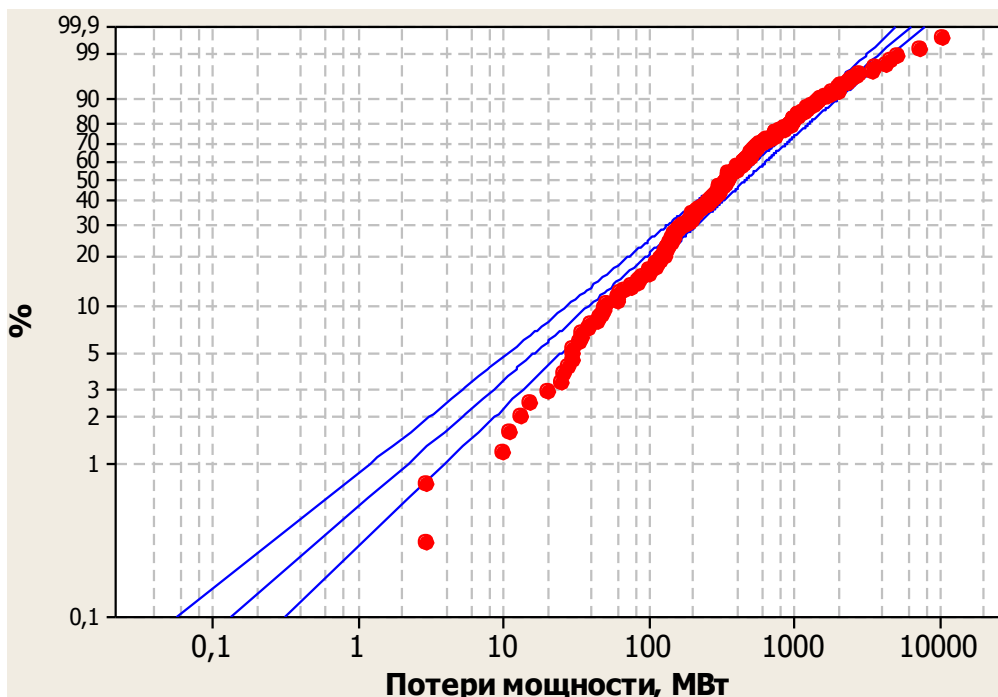


Рисунок 4 Анализ веерных отключений второй магистрали МСО

Точечный график на рис. 4 отражает анализ потерь мощности для второй сети МСО. Похоже, как и для первого случая, вторая система так же доказывает степенной закон по формуле $\sigma_m \propto m^{1,00}$ и масштабирующая экспонента $1,00$ – это аппроксимированная величина для H . Однако, в отличие от первого случая, здесь $H \approx 1,0$, что отражает детерминистскую природу временного ряда веерного отключения или причину предшествующего события во временном ряде веерного отключения, делая его совершенно неизбежным.

Выводы

После анализа собранной статистики веерных отключений можно сделать следующие заключения.

1. Учитывая текущие состояния магистральных сетей и растущую на них нагрузку, количество веерных отключений будет только увеличиваться. Для их предотвращения необходимы защитные меры: введение прогнозирующей автоматики. Увеличение пропускной способности, эффективный менеджмент.

2. Большинство каскадных аварий происходят вследствие начального единичного инициирующего события, которое часто остается незамеченным и не анализируется. 85% крупных аварий могли бы быть предотвращены своевременной реакцией персонала на подобные единичные события.

3. Статистика каскадных аварий была обработана методом скользящего окна. Обе сети показали степенную зависимость.

Литература

1. Кобец Б. Б., Волкова И. О. Инновационное развитие электроэнергетики на базе концепции Smart Grid. — М.: ИАЦ Энергия, 2010. — 208 с.
2. Department of Energy USA. <http://www.energy.gov/statistics>.
3. World Energy Outlook 2015. International Energy Agency (IEA), Paris. — 2015. — 15 pp.
4. British Petroleum Energy Outlook 2016 Edition. British Petroleum, London – 2016. – 80pp.
5. Pouyan Pourbeik, Prabha Kundur The anatomy of a power grid blackout – Root causes and dynamics of recent major blackouts. – Article in IEEE Power and Energy Magazine, Piscataway, New Jersey – 2016.
6. J. Blumschein, Y. Yelgin, M. Kereit Blackout Prevention by Power Swing Detection and Out-of Step Protection. – Journal of Power and Energy Engineering (JPEE), Irvine CA – 2014.
7. Friew Gebremedhin Abraha Statistics of Electric Power Blackouts: Data Analysis and Data Modelling. – Norwegian Univecity of Science and Technology, Trondheim – 2013, 71 с.
8. North American Electric Reliability Council Technical Analysis of the August 14, 2003, Blackout: What Happened, Why, and What Did We Learn? – Report to the NERC Board of Trustees by the NERC Steering Group, New Jersey - 2004.
9. A. Farina, A. Graziano, F. Mariani, and F. Zirilli. Probabilistic analysis of failures in power transmission networks and phase transitions: Study case of a high-voltage power transmission network. Journal of Optimization Theory and Applications, 139(1), 2008.
10. M. Vaiman, K. Bell, Y. Chen, B. Chowdhury, I. Dobson, P. Hines, M. Papic, S. Miller, and P. Zhang. Risk assessment of cascading outages: Methodologies and challenges. Power Systems, IEEE Transactions on, 27(2), 2012.
11. David L. Pepyne. Topology and cascading line outages in power grids. Journal of Systems Science and Systems Engineering, 16(2), 2007.
12. Bei Gou, WeibiaoWu. Is the prediction of power system blackouts possible? In Power and Energy Society General Meeting – Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE, 2008.
13. Кузнецов П.А., Солёный С.В. Борьба с веерными отключениями в системах электроснабжения. – Завалишинские чтения: молодежная секция. Сб. докл. / СПб.: ГУАП., 2016, 210с.
14. Кузнецов П.А., Степанов О.А., Юдин А.В. Структура автоматической системы компенсации реактивной мощности промышленных предприятий. – Проблемы недропользования: сборник трудов международного форума-конкурса молодых ученых. Часть II/ Санкт-Петербургский горный университет. СПб, 2016. 247с.

P. Kuznetsov, A. Judin, S. Maslo, V. Kliukovkin. Mathematical forecast of blackouts in power networks. This article describes the problem of blackouts or cascade malfunctions which appears in electrical networks. There were analyzed blackout's data on the two power lines according to the collected information and there was chosen a model for the next profound analysis of their expansion laws. There is shown a tendency for their amount increase. There was developed an algorithm for standard blackout duration which reason can be a single initiating event. There was also held an analysis of power losses in two power grids for the long term time interval using scaled window variance method. There are shown dependencies which reveal the obedience of blackout's behavior according to the power law.

Keywords: *blackout, mathematical model, outage analysis, Minitab, statistical analysis, emergency mode in power grid.*

Особенности решения задачи определения координат летательного аппарата на основе дельта-преобразований

П. П. Кравченко, Л.В. Пирская
Южный федеральный университет
kravchenkopp@sfedu.ru, lyubov.pirskaya@gmail.com

Кравченко П.П., Пирская Л.В. Особенности решения задачи определения координат летательного аппарата на основе дельта-преобразований. В работе рассматриваются возможности решения задачи локальной навигации бортовым специализированным вычислительным устройством. В постановке задачи локальной навигации: определение координат летательного аппарата на основе координат, расположенных на относительно близком расстоянии от начала локальной системы координат групп четверок маяков и дальности от летательного аппарата до маяков. На основании имеющихся данных для каждой группы формируется в стандартном для дальномерной навигации система четырех уравнений, которая преобразовывается к СЛАУ третьего порядка. Решение данной СЛАУ с непрерывными переменными свободными членами определяет координаты летательного аппарата. В работе рассмотрены особенности алгоритмизации решения задачи локальной навигации при наземной подготовке предложенного задания и решении задачи на борту летательного аппарата в режиме реального времени. Проведено исследование возможности использования дельта-преобразований второго порядка для решения СЛАУ в поставленной задаче на различных примерах размещения маяков. Полученные результаты экспериментов с использованием компьютерного моделирования подтверждают возможность получения решения СЛАУ за одну итерацию в установившемся процессе с временным шагом, представляющим практический интерес, а начало установившегося процесса обеспечивается на достаточно большой удаленности от начала координат.

Ключевые слова: *определение координат летательного аппарата, решение СЛАУ, дельта-преобразование второго порядка, бортовое специализированное вычислительное устройство.*

Введение

В настоящее время актуальным являются решение вопросов управления и навигации беспилотным летательным аппаратам (БПЛА). Решение бортовой задачи локальной навигации, в частности, определение координат БПЛА, осуществляется в режиме реального времени в условиях одновременного выполнения других бортовых задач при определенном временном шаге дискретизации процессов управления [1,2]. В связи с этим весьма актуальной является проблема минимизации временных и аппаратных затрат для решения поставленной задачи в пределах временного шага, а также обеспечение минимальной задержки решения относительно начала временного шага, что связано с ограничением влияния задержки на ошибку управления. Для решения задач с ограничениями отмеченного характера возможности применения высокопроизводительных вычислителей общего назначения ограничены.

Таким образом, возникает необходимость в создании специализированного вычислительного устройства, которое должно обеспечивать высокопроизводительное и экономичное по аппаратным затратам решение задачи локальной навигации, и в частности, определение координат летательного аппарата на основе решения СЛАУ в реальном масштабе времени. В рамках данной работы рассматривается возможность построения специализированного вычислительного устройства на основе использования дельта-преобразований, позволяющих при использовании итерационного метода решения СЛАУ организовывать вычислительный процесс с исключением операций многократного умножения и с получением результата за одну итерацию установившегося процесса [3-8].

С точки зрения эффективного использования ресурсов бортовых вычислительных средств представляет интерес такая организация вычислительного процесса, когда обработка информации на уровне одной итерации выполняется с достаточной точностью, высоким быстродействием и с предельно большим временным шагом. В таких условиях возможно предъявление наиболее низких требований по частоте формирования данных, характеризующих изменяющиеся свободные члены СЛАУ (например, частоте формирования расстояний от

летательного аппарата до маяков локальной системы навигации [1,2]), а также к производительности вычислительных средств с учетом возможности одновременной реализации других алгоритмов и программ. Актуальность эффективного решения СЛАУ даже невысоких порядков в рассматриваемых условиях резко возрастает, когда необходимо одновременно решать большое количество СЛАУ, что имеет место при использовании маяковой системы локальной навигации с одновременным решением задачи обеспечения целостности [9].

В настоящей работе рассматриваются вопросы построения высокопроизводительных экономичных по аппаратным ресурсам специализированных вычислителей для одновременного решения большого количества СЛАУ на основе дельта-преобразований второго порядка, соответствующих задаче локальной навигации на четыре маяка.

Постановка задачи локальной навигации

Одна из задач определения местоположения БПЛА базируется на использовании нескольких разнесенных в пространстве маяков, расположенных на относительно близком расстоянии от начала локальной системы координат. Бортовой вычислитель получает значения дальностей от БПЛА до маяков. На основании имеющихся координат маяков и полученных дальностей можно сформировать навигационные определения в стандартном для дальномерных навигационных систем вида [10]:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = D_i^2, \quad (1)$$

где x, y, z – координаты БПЛА; (x_i, y_i, z_i) – координаты i -го маяка в выбранной системе координат; D_i – дальность до i -го маяка. Решение системы уравнений для всех маяков позволяет определить местоположения БПЛА (x, y, z) в пространстве.

Решение данной задачи локальной навигации на борту БПЛА включает 2 этапа.

Первым этапом является определение текущих координат БПЛА путем решения систем уравнений, каждое из которых базируется на данных размещения соответствующих этой системе четырех маяков и дальностей БПЛА до этих маяков.

Вторым этапом является решение задачи целостности, обеспечивающей достоверность получаемых результатов с учетом возможных нарушений в работе маяков. Исходными данными для задачи целостности являются формируемые множественные значения координаты БПЛА, полученные на первом этапе. Для успешного решения данной задачи необходимо использование большого количества групп маяков, позволяющих независимо формировать по каждой группе координаты БПЛА. В работе, собственно, решение задачи целостности не рассматривалось.

Задача локальной навигации на четыре маяка

В постановке задачи рассматриваются:

- координаты четырех маяков в декартовой системе координат: (X_1, Y_1, Z_1) , (X_2, Y_2, Z_2) , (X_3, Y_3, Z_3) (X_4, Y_4, Z_4) ; данные координаты формируются при установке маяков путем выполнения измерений на местности;

- расстояния от каждого маяка до БПЛА: D_1, D_2, D_3, D_4 ; данные координаты формируются в полете с использованием специальной бортовой аппаратуры.

Необходимо определить координаты X_p, Y_p, Z_p БПЛА.

В соответствии с (1) квадраты расстояний от 4-х маяков до БПЛА имеют вид:

$$\begin{cases} (X_p - X_1)^2 + (Y_p - Y_1)^2 + (Z_p - Z_1)^2 = D_1^2 \\ (X_p - X_2)^2 + (Y_p - Y_2)^2 + (Z_p - Z_2)^2 = D_2^2 \\ (X_p - X_3)^2 + (Y_p - Y_3)^2 + (Z_p - Z_3)^2 = D_3^2 \\ (X_p - X_4)^2 + (Y_p - Y_4)^2 + (Z_p - Z_4)^2 = D_4^2 \end{cases}.$$

Раскрываем скобки и получаем:

$$\begin{cases} X_p^2 - 2X_p X_1 + X_1^2 + Y_p^2 - 2Y_p Y_1 + Y_1^2 + Z_p^2 - 2Z_p Z_1 + Z_1^2 = D_1^2 & (2.1) \\ X_p^2 - 2X_p X_2 + X_2^2 + Y_p^2 - 2Y_p Y_2 + Y_2^2 + Z_p^2 - 2Z_p Z_2 + Z_2^2 = D_2^2 & (2.2) \\ X_p^2 - 2X_p X_3 + X_3^2 + Y_p^2 - 2Y_p Y_3 + Y_3^2 + Z_p^2 - 2Z_p Z_3 + Z_3^2 = D_3^2 & (2.3) \\ X_p^2 - 2X_p X_4 + X_4^2 + Y_p^2 - 2Y_p Y_4 + Y_4^2 + Z_p^2 - 2Z_p Z_4 + Z_4^2 = D_4^2 & (2.4) \end{cases}$$

В рамках данной работы с целью создания более благоприятных условий для формирования нормы достаточной ограниченности переход к СЛАУ третьего порядка предлагается осуществлять путем следующей последовательности вычитаний: (2.1)-(2.2), (2.2)-(2.3), (2.3)-(2.4). В результате получаем:

$$\begin{cases} 2X_p(X_2 - X_1) + 2Y_p(Y_2 - Y_1) + 2Z_p(Z_2 - Z_1) = D_1^2 - D_2^2 + X_2^2 + Y_2^2 + Z_2^2 - X_1^2 - Y_1^2 - Z_1^2 \\ 2X_p(X_3 - X_2) + 2Y_p(Y_3 - Y_2) + 2Z_p(Z_3 - Z_2) = D_2^2 - D_3^2 + X_3^2 + Y_3^2 + Z_3^2 - X_2^2 - Y_2^2 - Z_2^2 \\ 2X_p(X_4 - X_3) + 2Y_p(Y_4 - Y_3) + 2Z_p(Z_4 - Z_3) = D_3^2 - D_4^2 + X_4^2 + Y_4^2 + Z_4^2 - X_3^2 - Y_3^2 - Z_3^2 \end{cases} \quad (3)$$

Сходимость итерационного решения СЛАУ обеспечивается в соответствии с условиями для метода простой итерации [11-13], которые формируются на основе данных о размещении маяков.

Преобразуем (3) к виду:

$$\begin{cases} (X_2 - X_1)X_p + (Y_2 - Y_1)Y_p + (Z_2 - Z_1)Z_p = \frac{1}{2}(D_1^2 - D_2^2 + X_2^2 + Y_2^2 + Z_2^2 - X_1^2 - Y_1^2 - Z_1^2) \\ (X_3 - X_2)X_p + (Y_3 - Y_2)Y_p + (Z_3 - Z_2)Z_p = \frac{1}{2}(D_2^2 - D_3^2 + X_3^2 + Y_3^2 + Z_3^2 - X_2^2 - Y_2^2 - Z_2^2) \\ (X_4 - X_3)X_p + (Y_4 - Y_3)Y_p + (Z_4 - Z_3)Z_p = \frac{1}{2}(D_3^2 - D_4^2 + X_4^2 + Y_4^2 + Z_4^2 - X_3^2 - Y_3^2 - Z_3^2) \end{cases} \quad (4)$$

Далее для определения координат (X_p, Y_p, Z_p) БПЛА необходимо решать полученную систему уравнений (4) относительно X_p, Y_p, Z_p .

Особенности алгоритмизации решения задачи локальной навигации

В решении задачи локальной навигации имеет место два этапа:

- наземная подготовка предложенного задания: при этом выполняются все математические преобразования, которые могут минимизировать алгоритм по вычислительной трудоемкости при решении задачи на борту БПЛА, а также определяются те группы маяков по четыре, которые соответствуют условиям обеспечения сходимости [11-13];

- решение на борту подготовленной на земле задачи локальной навигации в обстановке реального времени.

В условиях наземной подготовки целесообразно устанавливать $n \gg 4$ маяков, что как было отмечено в разделе 1, связано в первую очередь с обеспечением эффективного решения задачи целостности.

Число сочетаний из n маяков по m вычисляется по формуле:

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

Предположим, что установлено $n=8$ маяков, тогда число сочетаний данных маяков применительно к данному алгоритму будет составлять $C_8^4 = 70$, но использование всех комбинации не обязательно.

В рамках данной работы рассматриваются возможности одновременного (параллельного) решения множества СЛАУ для 4-х маяков на основе специализированного вычислителя с алгоритмизацией решения на базе дельта-преобразований второго порядка [7-8].

На рисунке 1 представлена укрупненная структурная схема специализированного бортового вычислителя, отражающая особенности возможной реализации решения задачи локальной навигации, где в виде заштрихованного блока представлена группа специализированных вычислителей для решения СЛАУ на основе дельта-преобразований второго порядка и переменного кванта, D_n - расстояния от каждого n -ого маяка до БПЛА.

В бортовой вычислитель при наземной подготовке загружаются разности координат маяков и значения известных сумм квадратов свободных членов системы (4). В полете в режиме реального времени на универсальный бортовой вычислитель (процессор на рисунке 1) поступают значения дальностей $D_1^2, D_2^2, D_3^2, D_4^2$, и далее в представленном на рисунке 1 заштрихованном блоке осуществляется решение СЛАУ на основе алгоритма дельта-преобразований второго порядка [7-8].

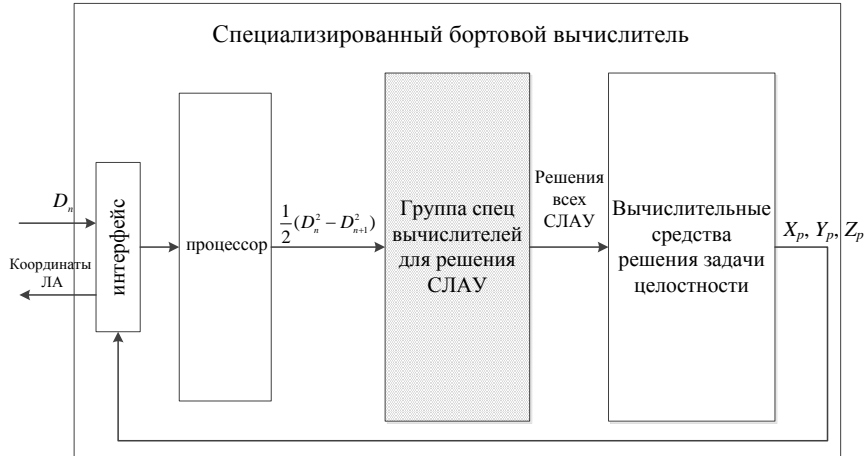


Рисунок 1 - Укрупненная структурная схема специализированного бортового вычислителя

Алгоритм параллельного решения СЛАУ с использованием дельта-преобразований второго порядка и переменного кванта

Рассмотрим решение СЛАУ, содержащую матрицу постоянных коэффициентов и в общем случае переменные свободные члены, удовлетворяющую условиям сходимости для метода простой итерации [11-13], и имеющую вид:

$$BY^*(t) = G(t). \quad (5)$$

Преобразуем систему:

$$Y^*(t) = AY^*(t) + D(t).$$

Переходим к форме записи с введением невязки $z(t)$ и использованием итерационного метода:

$$z(t) = Y(t) - AY(t) - D(t). \quad (6)$$

В приведенных системах $B = [b_{ij}]$, $A = [b_{ij}/b_{rr}]$ - матрицы коэффициентов размерности $n \times n$; $G(t)$, $D(t)$ - вектор-столбцы свободных членов системы (в частном случае для системы с постоянными свободными членами $G(t) = G = [g_r]$, $D(t) = D = [g_r/b_{rr}]$); $Y^*(t)$ - вектор-столбец неизвестных системы; $z(t)$, $Y(t)$ - вектор-столбцы невязок и приближенных значений неизвестных; t - независимая переменная; $\det A \neq 0$.

Алгоритм (7) параллельного решения СЛАУ (5) с использованием дельта-преобразования второго порядка и переменного кванта представим в следующей разностной форме для i -го шага при начальных условиях $Y_{r01} = 0$, $\nabla Y_{r01} = 0$, $z_{r01} = -D_{r01}$, $\nabla z_{r01} = -\nabla D_{r01}$, $r = \overline{1, n}$ [3, 7-8]:

- демодуляция:

$$\nabla^2 Y_{ril} = c_l A_{ril}; \quad (7.1)$$

$$\nabla Y_{ril} = \nabla Y_{r(i-1)l} + \nabla^2 Y_{ril}; \quad (7.2)$$

$$Y_{ril} = Y_{r(i-1)l} + \nabla Y_{ril}; \quad (7.3)$$

$$r = \overline{1, n}, \quad i = 1, 2, \dots, R_l, \quad l = 1, 2, \dots, P;$$

- формирование второй разности преобразуемой переменной:

$$\nabla^2 y_{ril} = \sum_{\substack{j=1 \\ (j \neq r)}}^n a_{rj} c_l^* A_{jil} + \nabla^2 D_{ril}; \quad (7.4)$$

- формирование значений невязок:

$$\nabla^2 z_{ril} = \nabla^2 Y_{ril} - \nabla^2 y_{ril}; \quad (7.5)$$

$$\nabla z_{ril} = \nabla z_{r(i-1)l} + \nabla^2 z_{ril}; \quad (7.6)$$

$$z_{ril} = z_{r(i-1)l} + \nabla z_{ril}; \quad (7.7)$$

- формирование значений переключающих функций и знаков квантов вторых разностей:

$$F_{ril} = z_{ril} + 1.5 \nabla z_{ril} + (0.5 \nabla z_{ril}^2 / c_l - 0.125 c_l) \text{sign}(\nabla z_{ril}); \quad (7.8)$$

$$A_{r(i+1)l} = -\text{sign} F_{ril}; A_{ril} \in \{+1, -1\}, c_l = 0,75c_l^*. \quad (7.9)$$

В алгоритме (7) c_l^* - вес модуля кванта преобразования на l -ом итерационном цикле ($c_l^* > 0$), P - количество циклов, выполняемых при постоянных по модулю значениях квантов, R_l - количество итераций в цикле. Кроме того, для стыков участков соседних циклов при решении СЛАУ с переменными свободными членами используются соотношения: $Y_{r0l} = Y_{rR(l-1)}$; $z_{r0l} = z_{rR(l-1)}$.

Сущность процесса решения системы (5) на основе алгоритма (7) состоит в том, что задаются начальные условия $Y_{r0}, \nabla Y_{r0}, D_{r0}, \nabla D_{r0}, r = \overline{1, n}$ (в частном случае, например, $Y_{r0} = 0, \nabla Y_{r0} = 0$, тогда, соответственно, $z_{r0} = -D_{r0}, \nabla z_{r0} = -\nabla D_{r0}, r = \overline{1, n}$) и организуется итерационный (переходный) процесс решения до вхождения в установившийся процесс, когда $|z_{rip}| \leq z_{steady}, z_{steady} > 0, r = \overline{1, n}$, где z_{steady} - достаточно малые, соответствующие обеспечению заданной точности решения системы величины. Дискретные значения $D_{ri}, r = \overline{1, n}, i = 1, 2, \dots$ в алгоритме (7) предполагаются численно определенными на каждом шаге [3].

В работах [7-8] разработаны целочисленные оценки параметров алгоритма (7), определяющие способ задания последовательности значений переменных квантов в циклах. Предложены следующие соотношения для квантов соседних циклов (S - степень минимального кванта, $s \in N$):

$$\text{для } R_{\text{int},1} = 4 \quad c_{P_{\text{int},1}-l}^* = 2^{2(l-1)-s}, l = \overline{P_{\text{int},1}, 1};$$

$$\text{для } R_{\text{int},2} = 8 \quad c_{P_{\text{int},2}-l}^* = 2^{3(l-1)-s}, l = \overline{P_{\text{int},2}, 1}.$$

В работах [7-8] введены и теоретически обоснованы эффективные условия окончания итерационного процесса в текущем цикле алгоритма (7):

$$\text{sign}\left(\frac{z_{r(i+1)l}}{c_l^*}\right) = -\text{sign}\left(\frac{z_{ril}}{c_l^*}\right),$$

$$\text{sign}\left(\frac{|z_{ril}|}{c_l^*}\right) = -\text{sign}\left(\frac{|z_{ril}|}{c_l^*} - \text{sign}\left(\frac{|z_{ril}|}{c_l^*}\right) \cdot \frac{|z_{ril}|^*}{c_l^*}\right);$$

$$r = \overline{1, n}, i = 1, 2, \dots, R_l; l = 1, 2, \dots, P.$$

Исследование возможности использования дельта-преобразований второго порядка для решения СЛАУ в задаче локальной навигации

Проводилось исследование использования дельта-преобразований второго порядка для решения СЛАУ в задаче локальной маяковой навигации на различных примерах размещения маяков. Особое внимание уделялось высоте расположения маяков, так как с уменьшением этих высот в большей мере проявляется влияние возмущений на ошибку итерационного процесса на каждом шаге.

Кроме того, исследование проводилось с учетом двух возможных способов организации вычислительного процесса:

1. Значения текущих дальностей поступают на каждом текущем шаге как переходного, так и установившегося процессов; длительность всех шагов (итераций) фиксирована.

2. Первый временной шаг решения используется для реализации переходного процесса при фиксированных, соответствующих этому шагу времени значениях дальностей (данный режим реализуется при наличии возможности передачи для данной задачи необходимых вычислительных ресурсов на шаге). После завершения данного переходного процесса продолжается решение на последующих временных шагах с реализацией одной итерации на каждом из этих шагов. Предполагается автоматическое выявление момента завершения переходного процесса и формирование соответствующего признака в решении задачи управления БПЛА по установлению наименьшего значения кванта преобразования.

Следует обратить внимание на то, что в основе организации вычислительного процесса по второму способу реализуются переходные процессы, которые по критерию сходимости соответствуют рассмотренным в [11-13] положениям. В случае использования данного способа организации переходного процесса при работе с переменными свободными членами представляются возможности сокращения количества временных шагов решения (но не количества итераций).

Исследование использования дельта-преобразований второго порядка для решения СЛАУ в задаче локальной маяковой навигации будем рассматривать на отдельных группах маяков по четыре на двух типах задач. Первая задача – определение координат при большой начальной высоте БПЛА и большой скорости, вторая задача – при посадке БПЛА.

В первом типе задач рассмотрены три варианта расположения 4-х маяков:

1. $(-200, -400, 20), (450, -100, 120), (120, 400, 170), (-100, 450, 420)$;
(8)
2. $(-200, -400, 5), (450, -100, 25), (120, 400, 0), (-100, 450, 25)$;
(9)
3. $(100, -40, 1), (150, -100, 0), (10, 400, 0), (-15, 450, 10)$.
(10)

Вариант (8) отличается от варианта (9) значительным снижением высоты расположения маяков. Вариант (10) имеет отличный вариант расположения маяков в плоскости XOY , а также имеет предельно низкие высоты маяков.

Начальная высота БПЛА составляет $H_0 = 17000$ м., скорость $V_1 = 700$ м/с и $V_2 = 300$, угол к поверхности земли 60° .

Решение всех СЛАУ, соответствующих представленным выше координатам маяков и составленным по системе (4), осуществлялось при обеспечении одинаковой точности 2^{-14} , характеризующейся значением 0,25 м.

В таблицах 1, 2 представлены результаты, характеризующие временные шаги ∇t_2 и высоту H_2 в метрах при использовании алгоритма на базе дельта-преобразований второго порядка [7-8]. Параметр H_2 отражает высоту, начиная с которой обеспечивается решение СЛАУ за одну итерацию в установившемся процессе с заданной одинаковой максимальной точностью решения для трех вариантов расположения маяков (8), (9), (10). В таблице 1 организация вычислительного процесса осуществлялась на основе первого способа, в таблице 2 – на основе второго способа. В таблицах 1, 2 в скобках обозначены номера описаний координат расположения маяков (8), (9), (10).

Таблица 1. Результаты экспериментов решения СЛАУ задачи локальной маяковой навигации на основе первого способа организации вычислительного процесса.

∇t_2	Высота ЛА для дельта-преобразований второго порядка, H_2					
	$V_1 = 700$ м/с			$V_1 = 300$ м/с		
	(8)	(9)	(10)	(8)	(9)	(10)
0,005	16357	15877	15507	16845	16277	16510
0,01	15707	13559	14764	16600	16010	15628
0,025	14363	11018	9652	15397	13847	14342
0,05	10308	3249	-	14868	12218	11722
0,075	7723	-	-	13610	6730	10322
0,1	-	-	-	12105	6084	4088

Таблица 2. Результаты экспериментов решения СЛАУ задачи локальной маяковой навигации на основе второго способа организации вычислительного процесса.

∇t_2	Высота ЛА для дельта-преобразований второго порядка, H_2					
	$V_1 = 700$ м/с			$V_1 = 300$ м/с		
	(8)	(9)	(10)	(8)	(9)	(10)
0,005	16705	16358	16632	16938	16843	16951
0,01	15868	12887	15644	16738	16615	16692
0,025	9914	6630	9494	15667	15446	15397
0,05	-	-	-	11790	9830	11332
0,075	-	-	-	5424	5705	4433

Анализ полученных в таблицах 1, 2 данных показывает, что при решении СЛАУ на базе дельта-преобразований второго порядка обеспечивается установившийся процесс и решение СЛАУ за одну итерацию уже на достаточно большой высоте по отношению к начальной с представляющим интерес для практического использования временным шагом ∇t_2 . При этом предполагается, что получение результата для всех уравнений системы осуществляется в пределах ~ 10 тактов при реализации алгоритма в специализированном вычислителе на базе ПЛИС. Организация вычислительного процесса как первым, так и вторым способом показывает в

значительной мере схожие результаты, однако наблюдается преимущество использования второго способа при расположении маяков в большей близости к поверхности земли.

Для управления БПЛА могут быть взяты вычисляемые координаты после начала установившегося процесса, то есть, начиная с высоты H_2 .

Во втором типе задач (при посадке БПЛА) рассмотрены следующие варианты расположения 4-х маяков:

1. $(100, -40, 1), (150, -100, 0), (10, 400, 0), (-15, 450, 10)$. (11)
2. $(100, -40, 5), (150, -100, 25), (10, 400, 0), (-15, 450, 25)$. (12)

Принимаем также: начальная высота БПЛА $H_0 = 200$, скорость $V_1 = 20$ м/с, угол к поверхности земли 4° .

В таблице 3 представлены результаты, характеризующие временные шаги ∇t_2 и высоту H_2 в метрах при использовании алгоритма на базе дельта-преобразований второго порядка [7-8] для первого и второго способов организации вычислительного процесса. Параметр H_2 отражает высоту, начиная с которой обеспечивается решение СЛАУ за одну итерацию в установившемся процессе с заданной одинаковой максимальной точностью решения 2^{-13} (характеризующейся значением 0,12 м.) для двух вариантов размещения маяков (11), (12).

Таблица 3. Результаты экспериментов решения СЛАУ задачи локальной маяковой навигации (при посадке БПЛА).

∇t_2	Высота БПЛА для дельта-преобразований второго порядка, H_2			
	на основе первого способа организации вычислительного процесса		на основе второго способа организации вычислительного процесса	
	(11)	(12)	(11)	(12)
0,005	155	169	192	196
0,01	115	124	183	194
0,025	30	17	148	191
0,05			103	159
0,75				111
0,1				20

Анализ полученных результатов, представленных в таблице 3, показывает, что при решении задачи локальной маяковой навигации, в частности, определения координат БПЛА при посадке, вычислительный процесс целесообразно организовать на основе второго способа. При этом первый временной шаг решения реализуется при фиксированных значениях дальностей. После завершения переходного процесса осуществляется решение на очередном временном шаге с реализацией одной итерации. Установившийся процесс начинается с момента использования наименьшего значения кванта преобразования. Для управления БПЛА могут быть использованы вычисляемые координаты после начала установившегося процесса, начиная с высоты H_2 .

В исследованиях рассматривалось решение СЛАУ без учета ошибок в оценке координат расположения маяков и измерения дальностей, вопросы целостности не рассматривались.

Заключение

В работе проведено исследование использования дельта-преобразований второго порядка для решения СЛАУ в задаче локальной навигации: определения координат ЛА, которое показало, что использование оптимизированных дельта-преобразований второго порядка представляет возможность решения СЛАУ с переменными свободными членами на каждом временном шаге установившегося процесса за одну итерацию. Полученные результаты экспериментов с использованием компьютерного моделирования подтверждают возможности оперирования с практически значимыми временными шагами работы системы при достаточно большой удаленности начала установившегося процесса от начала координат.

Работа выполнена в рамках выполнения базовой части государственного задания в сфере научной деятельности (проект № 3442 "Информационно-алгоритмическое обеспечение систем цифрового управления, автономной высокоточной навигации и технического зрения для перспективных летательных аппаратов: разработка теоретических основ проектирования, алгоритмов, способов эффективной и надежной программной реализации, использование высокопроизводительной вычислительной инфраструктуры для экспериментального моделирования").

Литература

1. Барабанов О.О., Барабанова Л.П. Математические задачи дальномерной навигации. – М.:

ФИЗМАТЛИТ, 2007. – 272 с.

2. Скрыпник, О.Н. Радионавигационные системы воздушных судов: Учебник/ О.Н. Скрыпник. – М.: ИНФРА-М, 2014. – 348 с.

3. Кравченко, П.П. Оптимизированные дельта-преобразования второго порядка. Теория и применение. Монография. – М: Радиотехника, 2010. – 288 с.

4. Кравченко, П.П. Итерационный метод решения систем линейных алгебраических уравнений, исключая операцию многоразрядного умножения [Текст] / П.П. Кравченко, Л.В. Пирская // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 214-224.

5. Pirskaaya, L.V. Iterative Algorithm for Solving of Linear Algebraic Equations Systems without Multi-bit Multiplication Operation // Engineering and Telecommunication (EnT), 2014 International Conference on. – IEEE, 2014. – P. 87 – 91.

6. Kravchenko P.P., Pirskaaya L.V. The method of organizing the iterative process of the system of the linear algebraic equations solution excluding the multidigit multiplication operation// Biosciences Biotechnology Research Asia December. – 2014. – Vol. 11(3). – P.1831-1839.

7. Кравченко П.П., Пирская Л.В. Метод организации итерационного решения систем линейных алгебраических уравнений с использованием дельта-преобразований второго порядка // Известия ЮФУ. Технические науки. – 2015. – № 6 (167). – С. 57-71.

8. Kravchenko P.P., Pirskaaya L.V., Khusainov N. Sh. Algorithm of iterative solution of linear algebraic equations systems based on the second order delta-transformation for specialized computers of real-time systems// Biosciences Biotechnology Research Asia December. – 2015. – Vol. 11(Spl. Edn. 2). – P. 279-289.

9. Н.Ш. Хусаинов, П.П. Кравченко, В.Н. Лутай, С.А. Тарасов, В.В. Щербинин Системы радионавигации современных и перспективных летательных аппаратов. Ч. 1. Методы определения местоположения и автономный контроль целостности: Монография. – Таганрог: Изд-во ЮФУ, 2015. – 118 с.

10. Шебшаевич, В.С., Дмитриев П.П., Иванцевич Н.В. и др Сетевые спутниковые радионавигационные системы. – М.: Радио и связь, 1993. – 408 с.

11. Березин, И.С. Методы вычислений, том 2. - М.: Наука, 1966. – 632 с.

12. Самарский, А.А. Численные методы. – М.: Наука, 1989. – 432 с.

13. Фаддеев, Д. К. Фаддеева. В.Н. Вычислительные методы линейной алгебры, 4-е изд., стереотип. - СПб.: Лань, 2009. – 734 с.

Kravchenko P.P., Pirskaaya L.V. Zvyagintseva A.V. Features of solving the problem of the aircraft coordinates determining on the basis of delta-transformations. In this paper it is considered the possibility of solving the task of local navigation by dint of a specialized onboard computing device. The problem statement of the local navigation (determining the coordinates of the aircraft) based on the coordinates of groups of four beacons located at a relatively close distance from the local coordinate system and range of the aircraft to the beacons. Based on available data, a system of four equations is formed in the standard representation for ranging navigation systems and then converted to a third-order linear systems. The solution of this linear systems with continuous variables constant terms define the coordinates of the aircraft. In this paper it is discussed the features of the solution algorithmization the task of local navigation while preparing the terrestrial part of the proposed tasks and solving problems on the aircraft board in real time. It is carried out the research of the possibility of using the second-order delta-transformations for solving linear systems in the task at different examples of beacons placement. The results of experiments using computer simulations confirm the possibility of solving the linear system at one iteration in a steady process with time-step having practical interest ($\sim 0,005 \div 0,025$) and the beginning of the established process is provided at a sufficiently large distance from the origin of coordinates.

Keywords: Determining the coordinates of the aircraft, the solution of leaner systems, the second order delta transformation, the specialized onboard computing device.

Численное моделирование тепловых процессов в теплообменных аппаратах с противоточными теплоносителями

В. К. Толстых, К. А. Пшеничный
Донецкий национальный университет
mail@tolstykh.com, kirillpch@mail.ru

Толстых В.К., Пшеничный К.А. Численное моделирование тепловых процессов в теплообменных аппаратах с противоточными теплоносителями. Предложена упрощённая модель описания конвективных, нестационарных, одномерных процессов теплопереноса для теплообменников с противоточными теплоносителями, содержащая модельный коэффициент теплопередачи между теплоносителями. Описана конечно-разностная схема расчёта тепловых процессов, приведены расчёты с данными промышленного теплообменника (кислородная станция Енакиевского металлургического завода). Сформулирована задача идентификации модельного коэффициента теплопередачи.

Ключевые слова: теплообменный аппарат, теплообменник, противоточные среды, идентификация, коэффициент теплопередачи, численное моделирование.

Введение

Теплообменные аппараты используются во многих отраслях промышленности, таких как металлургия, энергетика, химическое производство. От качественной работы теплообменников в значительной степени зависит работа других объектов-потребителей различных ресурсов с заданными температурными характеристиками. Поэтому для управления тепловыми процессами на производстве важно уметь с достаточной точностью моделировать и управлять работой теплообменных аппаратов. В данной работе рассматриваются конвективные, нестационарные, одномерные процессы теплопереноса для теплообменников с противоточными теплоносителями. Одна среда теплоносителя – это охладитель жидкий кислород, вторая – охлаждаемый воздух. Рассматриваемая общая математическая модель данных процессов для различных теплообменных аппаратов предполагает, что конструктивные и технологические особенности конкретных аппаратов должны учитываться адекватным модельным коэффициентом теплопередачи между противоточными средами. Такой модельный коэффициент может существенно отличаться от физического коэффициента теплопередачи. Очевидно, что для достаточно точного численного моделирования работы теплообменника по предлагаемой модели необходимо решать задачу идентификации коэффициента теплопередачи.

В первом разделе настоящей работы описывается математическая модель процессов теплопереноса между двумя противоточными средами с учётом гидродинамических особенностей потоков. Второй раздел посвящён численному моделированию конвективных тепловых процессов по конечно-разностным схемам типа «уголок» для обеих сред. Далее обсуждаются результаты расчётов численного моделирования потоков обеих сред и влияние на тепловые процессы модельного коэффициента теплопередачи. Предпоследние разделы посвящены обсуждению постановки и возможного алгоритма идентификации коэффициента теплопередачи.

Математическая модель процессов

На рис. 1 показана принципиальная схема работы противоточного теплообменника, в котором прямым потоком является воздух, а обратным – кислород. В одномерной постановке конвективные нестационарные тепловые процессы для указанного теплообменника можно описать следующей системой дифференциальных уравнений [1, 2]:

$$\begin{cases} \frac{\partial T_1}{\partial \tau} + v_1 \frac{\partial T_1}{\partial x} = \frac{1}{c_{p1}\rho_1} K(T_2 - T_1), \\ \frac{\partial T_2}{\partial \tau} - v_2 \frac{\partial T_2}{\partial x} = \frac{1}{c_{p2}\rho_2} K(T_1 - T_2), \end{cases} \quad (1)$$

где $T_1(\tau, x)$ и $T_2(\tau, x)$ температуры первого и второго теплоносителя (соответственно – воздух и жидкий кислород), движущихся со скоростями $v_1(\tau, x)$ и $v_2(\tau, x)$, K – коэффициент теплопередачи между теплоносителями, $\tau \in [0, \theta]$ – время, $x \in [x_a, x_b]$ – пространственная координата, c_p – удельная теплоемкость при постоянном давлении, $\rho(T)$ – плотность теплоносителей. Система (1) предполагает, что K одинаковый для обеих сред. Именно этот коэффициент требует дальнейшей идентификации.

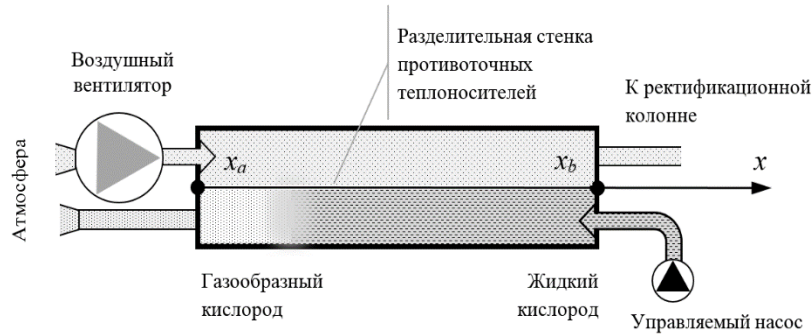


Рисунок 1. Принципиальная схема противоточного теплообменного аппарата

Граничные условия для системы (1) это:

$$\begin{aligned} T_1(\tau, x_a) &= T_{1a}(\tau), \\ T_2(\tau, x_b) &= T_{2b}(\tau). \end{aligned} \quad (2)$$

Как правило начальная температура теплоносителей вдоль всего теплообменника неизвестна. Зададим в качестве начального приближения линейное изменение температуры вдоль теплообменника при заданных граничных значениях:

$$\begin{aligned} T_1(0, x) &= T_1(0, x_a) - [T_1(0, x_a) - T_1(0, x_b)] \frac{x - x_a}{x_b - x_a}, \\ T_2(0, x) &= T_2(0, x_b) - [T_2(0, x_b) - T_2(0, x_a)] \frac{x - x_b}{x_a - x_b}. \end{aligned} \quad (3)$$

Такое приближение будет приводить к искусственным переходным тепловым процессам в начале расчётов до некоторого момента $\tau_0 \in (0, \theta]$. После τ_0 начальные условия уже не будут влиять на дальнейшие расчёты, и полученное распределение температур можно будет считать новым начальным условием для дальнейших расчётов. Для определения τ_0 будем задавать стационарные граничные условия как тепловые, так и гидродинамические. Момент τ_0 будет определяться как момент стабилизации искусственных переходных тепловых процессов.

Поскольку плотность теплоносителей существенно зависит от температуры [3], то и скорость течений теплоносителей будет различной по длине теплообменника. Волновыми процессами в промышленных теплообменниках пренебрегают из-за их быстротечности. Поэтому гидродинамические течения теплоносителей будем принимать стационарными для каждого расчётного временного слоя $\tau^j \in [0, \theta]$. Тогда из уравнения неразрывности $div(\rho v) = 0$ мы получаем уравнения для расчёта скоростей течений теплоносителей:

$$\begin{aligned} \rho_1^j(T) v_1^j(x) &= C_1 \\ \rho_2^j(T) v_2^j(x) &= C_2 \end{aligned} \quad (4)$$

где C_1^j , C_2^j – набор констант, которые определяются через известную скорость на границах потоков: $C_1^j = \rho_1^j v_{1a}^j$, $C_2^j = \rho_2^j v_{2b}^j$. Плотность и температура вещества связаны уравнением состояния. В нашей

постановке мы использовали уравнение Менделеева-Клапейрона для моля вещества: $PV_\mu = RT$, $\rho = \frac{m_\mu}{V_\mu}$, где m_μ – молярная масса, V_μ – молярный объем.

Таким образом, уравнения (1) –(4) образуют математическую модель для расчёта тепловых процессов рассматриваемой принципиальной схемы противоточного теплообменника.

Численное моделирование процессов теплопереноса

Для численного решения системы двух уравнений (1) с (2), (3) проводилась аппроксимация каждого уравнения по конечно-разностным схемам типа «уголок», которые показаны на рис. 2. Причем для прямого хода теплоносителя – это будет левый уголок с x_{i-1}, x_i , а для обратного потока теплоносителя – эта же схема приобретает вид правого уголка с x_i, x_{i+1} .

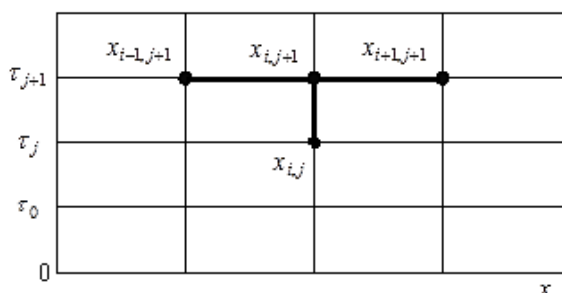


Рисунок 2. Конечно-разностная схема с двумя уголками для противоточных сред

Аппроксимация дифференциальных уравнений (1) по приведенной разностной схеме приводит к следующим алгебраическим уравнениям [4]:

$$\begin{cases} \frac{T_{1,i}^{j+1} - T_{1,i}^j}{\Delta \tau} + v_{1,i}^j \frac{T_{1,i}^{j+1} - T_{1,i-1}^{j+1}}{\Delta x} = \frac{1}{c_{p1}\rho_{1,i}^j} K(T_{2,i}^j - T_{1,i}^j), \\ \frac{T_{2,i}^{j+1} - T_{2,i}^j}{\Delta \tau} - v_{2,i}^j \frac{T_{2,i}^{j+1} - T_{2,i-1}^{j+1}}{\Delta x} = \frac{1}{c_{p2}\rho_{2,i}^j} K(T_{1,i}^j - T_{2,i}^j). \end{cases}$$

Откуда можно найти

$$\begin{cases} T_{1,i}^{j+1} = \frac{\gamma}{\gamma+1} T_{1,i-1}^{j+1} + \frac{1}{\gamma+1} T_{1,i}^j + \frac{\Delta \tau}{1+\gamma} \frac{1}{c_{p1}\rho_{1,i}^j} K(T_{2,i}^j - T_{1,i}^j), \\ T_{2,i}^{j+1} = \frac{\gamma}{\gamma+1} T_{2,i+1}^{j+1} + \frac{1}{\gamma+1} T_{2,i}^j + \frac{\Delta \tau}{1+\gamma} \frac{1}{c_{p2}\rho_{2,i}^j} K(T_{1,i}^j - T_{2,i}^j) \end{cases} \quad (5)$$

при условии:

$$\gamma = \frac{v_{\max} \tau}{h} \leq 1.$$

Расчёт решения для первой среды находится слева-наперво, а для второй – справа-налево на каждом временном слое.

Результаты расчётов и оценка влияния коэффициента теплопередачи

Модель (1)-(4) можно будет считать приемлемой для какого-нибудь теплообменника, только если коэффициент K можно будет задать таковым, что модельная температура будет мало отличаться от экспериментально наблюдаемой температуры для этого же теплообменника. Оценим значение и влияние

модельного коэффициента K на тепловые процессы в условиях промышленного теплообменника (кислородный цех Енакиевского металлургического завода).

На рис. 3 представлены результаты расчётов распределения температуры в обоих теплоносителях для двух различных коэффициентов теплопередачи $K_1 = 10^4$ и $K_2 = 10^5$. Расчёты проводились при следующих данных. Длина теплообменного аппарата $L = 10$ м. Приближения для начальных условий температуры рассчитывались по (3) при $T_1(0, x_b) = 315,15$ К, $T_2(0, x_a) = 303$ К. Температура на границах теплообменника задавалась постоянной и равной $T_1(\tau, x_a) = 315,15$ К, $T_2(\tau, x_b) = 90,2$ К. Давления в потоках принимались $p_1 = 5 \cdot 10^6$ Па, $p_2 = 2,4 \cdot 10^6$ Па. Расчётная сетка задавалась с ячейками по пространству $\Delta x = 1$ м и шагами по времени $\Delta \tau = 0,5$ с.

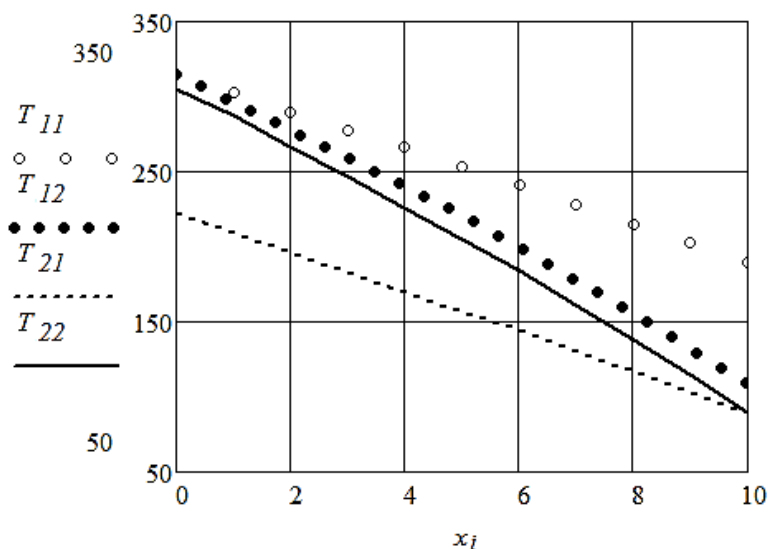


Рисунок 3. Влияние коэффициента теплопередачи на распределение температур в воздухе T_{11}, T_{12} и в кислороде T_{21}, T_{22} при K_1 и K_2 соответственно

Как показали расчеты, переходные процессы стабилизировались к моменту $\tau_0 \approx 20$ с. После этого времени нестационарность процессов, вносимая приближительными начальными условиями (3), исчезала. Система переходила в установившееся состояние. Других источников нестационарности в рассматриваемом примере не было. Полученное время стабилизации совпадает со временем переходных процессов теплообменного аппарата Енакиевского металлургического завода.

Из приведенных двух расчётов с K_1 и K_2 видно, что коэффициент K существенно влияет на тепловые процессы теплообменника. При большем коэффициенте теплопередачи K_2 температуры обеих сред в большей степени сближаются, что иллюстрируют кривые T_{12} и T_{22} . При меньшей теплопередаче (коэффициент K_1) температуры противоточных сред слабо приближаются друг к другу, что видно по кривым T_{11} и T_{21} .

Распределение температур T_{12} и T_{22} близко к реальным рабочим температурам теплообменника Енакиевского металлургического завода. Это свидетельствует о корректности приведенного численного моделирования противоточных процессов и, что коэффициент теплопередачи K_1 был выбран близким к правильному модельному значению.

Задача идентификации

Математическая модель (1)-(4) содержит в уравнении (1) неизвестный коэффициент K . В нашем случае – это модельный, а не физический коэффициент, включающий в себя погрешности всех допущений модели: неопределённость точной геометрии теплообменника, значительность толщины разделительной стенки и отсутствие учёта в ней тепловых процессов, приближённое гидродинамическое моделирование и т.д. Очевидно, что коэффициент K для каждой конкретной установки может быть найден только с использованием оптимизационных методов идентификации [5, 6, 7] на основе экспериментальных данных, например, по результатам измерений температуры потоков.

Очевидно, что удобнее (проще) проводить идентификацию при установившейся работе теплообменника. В этом случае задача идентификации будет формулироваться следующим образом. Необходимо найти коэффициент K модели (1)-(4) доставляющий минимум функционалу:

$$J(K) = \int_{x_a}^{x_b} [T_1(\tau_0, x) - T_e(\tau_0, x)]^2 dx \rightarrow \min \quad (6)$$

где $J(K)$ – критерий качества идентификации, который в нашем случае представляет собой невязную одномерную функцию модельного коэффициента теплопередачи K , T_e – экспериментально наблюдаемая температура воздуха в теплообменнике. Коэффициент K , найденный из условия минимума J , будет содержать в себе все погрешности математической модели (1)-(4) таким образом, чтобы температура, рассчитываемая в дальнейшем по этой модели, была максимально приближена к реальной рабочей температуре теплообменника.

Функционал (6) может быть «облегчён» с учётом того, что мы рассматриваем установившиеся процессы, а сам коэффициент K одинаков по всей длине теплообменника. Здесь расхождение модельной и реальной температур можно контролировать только на выходе x_b охлаждённого воздуха из теплообменника. В этом случае критерий качества идентификации принимает вид:

$$J(K) = [T_{1b}(\tau_0) - T_e(\tau_0)]^2 \rightarrow \min \quad (7)$$

Для решения задач идентификации (6), (7) могут применяться различные методы одномерной минимизации, например, метод золотого сечения, как метод поиска экстремума функции одной переменной на заданном отрезке. Мы имеем $J(K): [K_a, K_b] \rightarrow R$. Отрезок $[K_a, K_b]$ делится в пропорции золотого сечения, то есть выбираются две точки K_1, K_2 из условия:

$$K_1 \approx K_a + 0,4(K_b - K_a),$$

$$K_2 \approx K_a + 0,6(K_b - K_a).$$

Далее в этих точках оцениваются значения критерия качества. Если $J(K_1) \leq J(K_2)$, тогда принимают $K_b \leftarrow K_2$ иначе $K_a \leftarrow K_1$ и все расчёты повторяют заново до уменьшения длины отрезка $[K_a, K_b]$ до желаемых минимальных размеров. После чего определяется решение задачи (7) как $K = (K_b - K_a) / 2$.

Выводы

Полученные результаты расчётов течений противоточных сред (рис. 3) для общей упрощённой модели теплообменника (1) -(4) с конечно-разностной схемой (5) показали физически обоснованные распределения температур в теплообменнике. Для использования полученной модели на практике необходимо решать задачу идентификации коэффициента теплопередачи K приведенной модели, что можно делать по упрощённому критерию (7) методом золотого сечения.

Литература

1. Девятов, Б. Н. Теория переходных процессов в технологических аппаратах с точки зрения задач управления. – Новосибирск: Наука, 1964. – 324 с.
2. Девятов, Б. Н. Теория и методы анализа управляемых распределенных процессов / Б.Н. Девятов, Н.Д. Демиденко. – Новосибирск: Наука, 1983. – 310 с.
3. Ландау, Л.Д., Лифшиц, Е.М. Теоретическая физика гидродинамика. 3-е изд., испр. – М.: Наука. Гл. ред. физ.-мат. лит., 1986. – 736 с.
4. Самарский, А. А. Теория разностных схем. М.: Наука, 1977. – 656 с.
5. Толстых В.К., Недопекин Ф.В., Бодряга В.Е. Идентификация теплофизических параметров в виде полиномов, зависящих от температуры. – Технічна теплофізика та промислова теплоенергетика, N1. – 2009. – С. 193-199.
6. Tolstykh V.K. Optimality Conditions and Algorithms for Direct Optimizing the Partial Differential Equations, Engineering. Scientific Research Publishing. Vol. 4, N7. – 2012, pp. 390-393.
7. Tolstykh V.K. Optimization for Systems Governed by Partial Differential Equations, The Electronic International Journal "Advanced Modeling and Optimization", Vol. 14, N 3. – 2012, pp. 703-716.

Толстых В.К., Пшеничный К.А. Численное моделирование тепловых процессов в теплообменных аппаратах с противоточными теплоносителями. Предложена упрощённая модель описания конвективных, нестационарных, одномерных процессов теплопереноса для теплообменников с противоточными теплоносителями, содержащая модельный коэффициент теплопередачи между теплоносителями. Описана конечно-разностная схема расчёта тепловых процессов, приведены расчёты с данными промышленного теплообменника (кислородная станция Енакиевского металлургического завода). Сформулирована задача идентификации модельного коэффициента теплопередачи.

Ключевые слова: теплообменный аппарат, теплообменник, противоточные среды, идентификация, коэффициент теплопередачи, численное моделирование.

Tolstykh V.K., Pshenichny K.A. Numerical modeling of thermal processes in heat exchangers with countercurrent heat transfer fluids. A simplified model for the description of convective, nonstationary, one-dimensional processes heat transfer for heat exchangers with countercurrent heat transfer fluids, comprising coefficient of heat transfer between the heat carriers was proposed. Described finite-difference scheme for calculating thermal processes, are presented with industrial heat exchanger data (oxygen plant Yenakiyevo Metallurgical Plant). Was formulated a task of identification of the model of heat transfer coefficient.

Keywords: heat exchanger, countercurrent environments, identification, heat transfer coefficient, numerical simulation.

Выбор модели для компьютеризированной системы прогнозирования поведения групп лиц

А.В. Сложеницын, С.Ю. Землянская, Н.К. Андриевская
Донецкий национальный технический университет
SlogenitsinAV@gmail.com, zsaa@ya.ru, nataandr@yandex.ru

Сложеницын А.В., Землянская С.Ю., Андриевская Н.К. Выбор модели для компьютеризированной системы прогнозирования поведения групп лиц. Рассмотрено общее состояние направления моделирования поведения человека и поведения толпы. Выявлены достоинства и недостатки существующих моделей и готовых программных комплексов, с точки зрения прогнозирования преступного поведения. Определены основные критерии, необходимые для построения новой модели поведения человека, лишенной недостатков проанализированных продуктов. Выбраны методы и подходы для создания новой модели.

Ключевые слова: моделирование поведения, прогнозирование деятельности, искусственный интеллект, агентное моделирование, нейронная сеть.

Введение

На сегодняшний день человечество достаточно полно изучило поведение индивидуума и поведение толпы, выявило множество факторов, определяющих их тип поведения и причинно-следственные связи, приводящие к различным событиям. Однако к настоящему времени программных реализаций моделирования поведения человека немного.

Для решения задач моделирования или прогнозирования поведения применяются различные методы: соцопросы, сбор информации о предпочтениях человека при его обращении к поисковым системам в интернете, статистический анализ прошлых реакций человеческих масс и прочие. Но все эти методы зачастую оказываются недостаточными для качественного прогнозирования поведения индивидуума.

Многие компании, страны, области науки заинтересованы в получении ответов о реакции и действиях различных людей в ответ на всевозможные события: выпуск новой продукции, проведение митинга, чрезвычайное происшествие и прочее.

Весьма важным направлением моделирования человеческой деятельности является моделирование преступной деятельности и прогнозирование криминогенной обстановки в городе. При успешных результатах построения модели станет возможным прогнозирование широкого спектра человеческой деятельности.

В данной статье основной целью исследования является обзор текущих разработок для моделирования поведения человека, а также выделение на их основе методов и подходов, наиболее подходящих для программной реализации системы прогнозирования преступной деятельности групп лиц.

Моделирование поведения человека

Так что же такое моделирование поведения человека? В самом общем смысле – это возможность предсказать человеческие реакции и действия в ответ на любые события. Но не стоит забывать, что поведение человека состоит из множества факторов: психология субъекта, его социальные связи, его прошлый опыт и т.д. Учитывая количество внешних и внутренних факторов, которые влияют на поведение человека, а также то, что сам человек не может точно сказать, почему он поступил так, а не иначе, создание модели поведения – отнюдь нетривиальная задача.

Сегодня моделирование человека – это многоотраслевое междисциплинарное направление. В основном, моделирование человека и групп лиц разделяют по задачам моделирования, например, моделирование поведения толпы, моделирование поведения отдельного индивида в самоорганизующейся толпе, моделирование деятельности единого субъекта, моделирование влияния поведения отдельных индивидов на изменение общего состояния группы лиц и т.д.

В первую очередь, различные задачи подразумевают разные подходы и средства для моделирования. Так, для моделирования поведения групп лиц чаще используют методы роевого интеллекта, элементы теории игр, аппарат искусственных нейронных сетей и прочие. В то же время для моделирования поведения индивида в группе используются методы агентного моделирования, математический аппарат нечеткой логики, некоторые

концепции модели жизнеспособной системы.

Примером использования перечисленных технологий, может выступать модель движения толпы на основе роевого поведения насекомых, исследуемая Карповым В.Э. в статье «Коллективное поведение роботов. Желаемое и действительное» [1]. Другим примером может служить эволюция агентов и их поведение в различных средах, рассматриваемые Петрушаниным М.В. и Самариным А.М. в работе «Эволюционная модель поведения агентов в среде, побуждающей к коллективным действиям» [2].

Естественно, наиболее прогрессивные результаты получаются при соединении нескольких подходов, но хотя совмещение моделирования поведения толпы и поведения отдельного индивида довольно перспективно, оно сопряжено с рядом проблем. В частности, большой объем обрабатываемых системой данных; проблема создания качественного математического аппарата, способного корректно учитывать «сложного» индивида в толпе; ресурсоемкий процесс создания (в том числе, не всегда экономически обоснованный), по сравнению с отдельным моделированием поведения толпы и индивида, для более узких, конкретных задач.

Разработки в области моделирования поведения

Многие исследовательские центры и отдельные фирмы уже создали ряд продуктов, позволяющих в определенной степени прогнозировать поведение групп лиц. Рассмотрим некоторые из них.

Программный продукт DI Guy – модуль искусственного интеллекта, моделирование поведения толпы с учетом каждого индивида [3].

Программа разработана и представлена компанией VT MAK. Основное ядро системы для моделирования поведения человека представлено в базовом пакете DI Guy. Дополнительными элементами являются DI Scenario, DI SDK, DI Motion editor. С их помощью возможно разрабатывать новые сценарии поведения и ситуативные среды, добиваться более реалистичного поведения отдельного человека (вроде движения ногами или хождения по кругу в ожидании); добавлять отображение лицевой анимации в формате 3D.

Внедренные в программу искусственный интеллект и графические возможности позволяют создавать реалистичные уличные драки, митинги, военные операции, допрос подозреваемых и многие другие ситуации. В связи с хорошо проработанными возможностями для моделирования поведения людей во время военных действий, большой интерес к данному проекту проявляют американские военные. Причем здесь важно заметить: при моделировании столкновения в мирном районе города с вражескими солдатами можно моделировать поведение не только этих солдат, но и поведение мирных жителей (каждого индивидуально), а также различных животных (кур, коров, лошадей). При качественном моделировании открывается возможность обучения солдат определять вероятных противников по жестам и мимике, даже с учетом поведения людей другой культуры [4].

Поскольку это коммерческий продукт, компания неохотно раскрывает использованные для моделирования технологии. Известно, что проект опирается на агентное моделирование, в котором отдельный человек – это агент с набором характеристик, реагирующий на деятельность других агентов (людей). Однако детали реализации агентов и их взаимодействия не раскрываются.

Плюсами данной системы является возможность моделирования большого количества сцен с сильной детализацией, а также возможность моделирования в этих сценах различных построек, транспортных средств, животных. Минусом является заданная и мало изменяемая модель поведения человека – основные реакции заданы как «реакция на агрессию» в различных проявлениях. Также имеется ряд характеров для разных людей, но в них не учитываются особенности прошлого человека или его социальные связи.

Институт MOVES при военно-морской высшей школе США в Монтерее, штат Калифорния, занимается моделированием человеческого поведения [5]. Так как это военная школа, то основное направление моделирования – военные операции и реакция людей на них. В данной области, разработчиками были достигнуты хорошие результаты. Однако, дополнительная разработка направлена еще и на социальное и культурное моделирование. Из таких разработок данного института хотелось бы упомянуть проект FOCUS под руководством Стива Холла и Джеффа Апплегета, а также когнитивное и перцептронное моделирование. Первый проект позволяет строить модель социальной сети населения, которая адекватно реагирует на работу СМИ и помогает определить тип телерадиовещания, при котором население будет настроено максимально лояльно к армии. Эта разработка поможет создать программу профилактики как для снижения числа терактов в оккупированном городе, так и для снижения уровня преступности. Дополнительно разработчики планируют создать функционал для прогнозирования наиболее неблагоденных (опасных) мест. Второй проект создается для улучшения понимания, как информация, доступная отдельному лицу, может быть интегрирована в подобие человеческого сознания для прогнозирования последствий различных направлений деятельности. То есть программа сможет давать ответ на вопрос, как человек воспринимает определенную деятельность и о возможных действиях человека, в ответ на какие-либо действия. [6]

В данном продукте в качестве основы для моделирования боя используется модель искусственного интеллекта, описывающая поведение объектов с точки зрения иерархических сетей. В звеньях этих сетей расположены вероятностные варианты поведения для различных ситуаций.

В процессе моделирования работы СМИ используется принцип географического положения, с помощью

которого определяется, какие слои населения имеют доступ к системам телерадиовещания. Затем определяется зона распространения информации «от человека к человеку», после чего моделируется реакция на вещание заданных объектов из заранее сформированного перечня.

Процесс создания проекта «Когнитивное и перцептронное моделирование» все еще находится на стадии разработки. В связи с этим информации о построенной модели немного. Единственное, что достоверно известно – текущие исследования основаны на многослойной нейронной сети, обучающейся без учителя.

Работа данного института позволяет понять, насколько различны технологии, применяемые для разных задач моделирования поведения. Это и является основным минусом каждого из представленных институтом проектов. Но, несмотря на это, каждый из проектов дает положительные результаты моделирования в своей области.

Наиболее подходящий для прогнозирования преступной деятельности проект – это моделирование боя на основе иерархических сетей. Однако использование данной модели становится практически невозможным, если преступность составляет менее 90-95% от общего числа исследуемых объектов. Это связано с тем, что на поле боя практически все лица являются прямыми участниками событий и обязаны реагировать в соответствии с общей тенденцией поведения. Данное обстоятельство практически исключает этот подход в области прогнозирования преступлений.

Еще одним интересным проектом стал проект Марко Страно и Роберта Брузоне [7]. Они собрали команду из психологов, медицинских экспертов и детективов. Целью этой команды стала разработка статистической модели, которая связывает профиль преступника с его/ее действиями, решениями, а также относительными последствиями на месте преступления.

Марко Страно и Роберта Брузоне определили переменные, которые играют существенную роль при характеристике поведения преступника во время насильственного преступления. Сто пятьдесят из этих переменных были выбраны путем анализа литературы и путем опроса выборки из нескольких сотен детективов и медицинских экспертов из разных стран, которые обычно участвуют в расследовании убийства. Эти переменные они используют для построения так называемых сетей Байеса. Байесовские сети являются графическими моделями, которые извлекают знание данной системы из эмпирических данных и отображают причинно-следственные связи между всеми соответствующими переменными. Используя условные вероятности, они могут определять степень, в которой переменные могут влиять друг на друга, даже если основные механизмы деятельности неизвестны. Данная работа направлена на определение профиля преступника и моделирование его возможного поведения. [7]

Плюсом этой системы можно считать хорошие показатели моделирования. Это объясняется тем, что переменные для системы собирались много лет из статистически верных данных. Минусами же являются: сложность разработки модели Байесовской сети такого масштаба и большие вычислительные мощности, требуемые для однократного моделирования. Так, для каждой переменной существует от 2 до 30 возможных значений, и многие переменные связаны с пятью и более другими переменными, при том, что переменных уже сейчас насчитывается 150 [7, с 5].

Инструмент имитационного моделирования AnyLogic был разработан российской компанией The AnyLogic Company. Данная программная среда создана с использованием многоподходного моделирования. Объединение системной динамики, агентного и дискретно-событийного моделирования позволяет использовать пакет программ AnyLogic в разнообразных областях знаний. [8]

Проект включает в себя различные библиотеки, написанные на языке Java. Эти библиотеки реализуют отдельные возможности системы, такие как моделирование физической окружающей среды, имитацию операций сортировочной станции, интерактивную анимацию 2D и 3D моделей, моделирование движения автомобилей и многое другое.

Применение агентного моделирования позволяет использовать AnyLogic для моделирования поведения человеческих масс. На текущий момент уже есть некоторые наработки в этой области, например, «Моделирование поведения толпы» [9].

Минусами этого продукта является отсутствие возможности добавления большого количества характеристик агенту, что не позволяет в полной мере имитировать деятельность человека, а также строго заданные отношения между агентами, что не дает возможности организовать социальную связь, близкую к человеческой.

Некоторые перспективные разработки все еще не готовы, но их разработчики ожидают получить хороший результат. Примером такой разработки является проект «Модель поведения человека» – это неформальное сообщество исследователей, занимающихся в той или иной форме компьютерным моделированием человеческого поведения [10].

Проект представляет собой модульную программу с возможностью включения или отключения основных компонентов. Компонентами являются «Эмоциональный модуль», «Модуль опыта», «Модуль внешней среды», «Модуль энергетического баланса», «Модуль привыкания и автоматизмов» и прочие. Центральный элемент модели – человек. Вторым по важности элементом модели является деятельность человека, то есть его конкретные действия. На текущий момент реализовано немного действий, однако уже

сейчас для каждого действия есть определенный набор альтернатив. Таким образом, возникла проблема принятия решений «модульным человеком». Для решения этой проблемы были введены факторы (временные, энергетические), цели, эмоции, интересы, результаты действий и прочее.

Стоит отметить, что помимо моделирования отдельного человека, модель предполагает моделирование групп людей, взаимодействующих друг с другом. Это введено для более корректного описания человека, ведь человек социален. Также для социализации человека в модели введены фирмы, на которых человек может работать.

К сегодняшнему дню выпущена демонстрационная версия «Модели поведения человека». Подходы, использованные в модели, тяжело классифицировать. Можно сказать, что некоторые методы являются концептуально новыми. Это отрицательно сказывается на их понимании и возможности внедрения, так как их формальное описание, а также публикации о достоинствах и недостатках их использования, не представлены. В результате можно сделать вывод, что разработки активно ведутся во многих направлениях, применяются различные подходы для реализации необходимого функционала. Однако многие разработки применимы исключительно в их предметной области. Также большинство программных реализаций опирается лишь на текущее состояние индивида, что является минусом, по сравнению с системами, учитывающими его прошлое.

Выбор модели для реализации

Анализируя вышеуказанные исследования и разработки, можно утверждать, что моделирование поведения человека применимо к различным областям знаний. Однако данные исследования зачастую направлены на довольно узкий круг задач и применение изложенных методов практически невозможно вне этого круга. Также следует заметить, что рассмотренные модели поведения человека крайне слабо учитывают его прошлое, а также делают слабый акцент на текущем психо-, социо- и физиологическом состоянии.

Это заключение позволяет говорить о необходимости создания нового проекта для моделирования поведения, в котором будут отсутствовать основные недостатки приведенных выше систем. Все еще остаются нерешенными вопросы моделирования адаптивной к разным задачам системы, способной прогнозировать деятельность человека.

Для реализации этой идеи необходимо создание модели, способной адекватно учитывать любые факторы, влияющие на поведение человека. В частности, необходимо учитывать события, произошедшие с человеком до моделируемого события, текущие психологическое и физиологическое состояние, социальное положение и социальные связи, и прочее. Рассмотрим идею более детально.

Поведение – способность человека и животных изменять свои действия, реагировать на воздействие внутренних и внешних факторов [11]. Данное множество факторов, в комплексе, и определяют вероятность конкретных действий:

$$\text{Действие} = F(\text{фактор1}, \text{фактор2}, \dots, \text{фактор}N)$$

Под фактором в данном случае может пониматься характеристика человека, событие, социальная связь и прочее. Множество действий, в свою очередь, определяют общее поведение человека:

$$\text{Поведение} = \{\text{действие1}, \text{действие2}, \dots, \text{действие}N\}$$

Соответственно, чем больше факторов и вариантов действий можно представить в модели поведения, тем точнее будет прогноз деятельности человека (при условии корректного учета влияния каждого фактора на шанс совершения поступка).

При успешном моделировании деятельности одного человека получаем возможность моделирования поведения групп лиц как результат деятельности множества людей. При этом не следует забывать о важности моделирования связей между людьми.

Таким образом, для построения модели необходимо:

1. Установить способ представления человека как множества персонализированных характеристик.
2. Выделить факторы, влияющие на деятельность\поведение человека (в частности, с точки зрения правонарушений).
3. Определить зависимости (степень влияния) отдельных факторов на каждый вариант действий.
4. Определить метод выбора действий на основании факторов.
5. Учесть в модели взаимное влияние индивидов друг на друга.

В этом случае человека необходимо представить, как обособленную единицу с индивидуальным набором характеристик и возможностью связи с другими единицами. Оптимальным подходом для этого является использование агентного моделирования с представлением человека в качестве агента.

Агентное моделирование – это раздел имитационного моделирования, где для создания поведения целой системы, разрабатываются элементы системы – взаимодействующие, но независимые агенты. Поведение системы в этом случае возникает как результат деятельности агентов. В качестве агентов допускается использование практически любых объектов. Однако они должны соответствовать необходимым нормам

«интеллектуальности». Например, на рисунке 1, изображен пример общей внутренней организации агента.

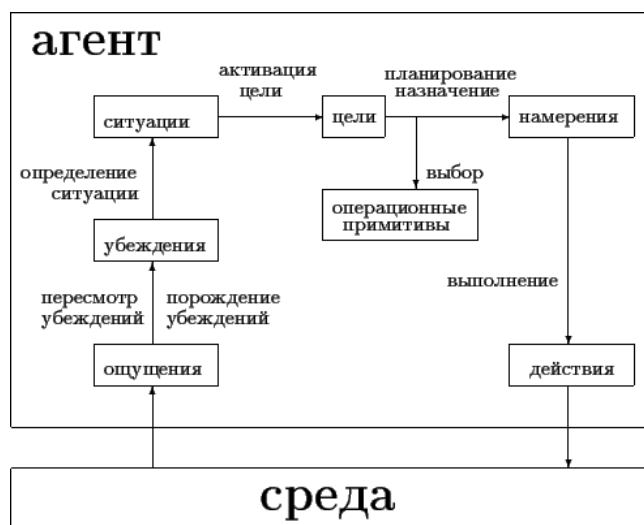


Рисунок 1 – Пример схемы реализации агента

Предположительно, количество характеристик человека весьма велико и прямо пропорционально достоверности результата моделирования. Это означает, что для хороших результатов может понадобиться большое количество данных о каждом человеке. Чтобы корректно учитывать различные и разноразмерные характеристики, а также получить оптимальную деятельность агента, возможно использование нейронной сети. Нейронная сеть – это система соединенных и взаимодействующих между собой «простых процессоров». Каждый из «процессоров» может выполнять лишь малые задачи, но при взаимодействии большого количества процессоров и корректных входных данных система может решать громоздкие и нетривиальные задачи. На вход сети, в случае моделирования поведения, необходимо подавать характеристики человека, приведенные к общей размерности. Необходимо обеспечить учет характеристик вместе с их влиянием на индивида (весами каждой характеристики). На выходе сети будет сформировано множество допустимых действий человека.

Комбинирование агентного подхода и нейронных сетей для получения состояния агента, при условии наличия качественных данных о характеристиках человека и корректной модели социальных связей, может дать хороший результат. Особенно интересным здесь является то, что чем больше характеристик мы внесем, тем точнее будет прогноз поведения, в то время, когда сами характеристики могут относиться к любой сфере жизнедеятельности человека. Это значит, что модель будет применима для моделирования поведения групп лиц практически в любой ситуации.

Заключение

Исследуемое в статье направление имеет огромный потенциал и, соответственно, хорошие перспективы. Любые разработки, направленные на исследование и прогнозирование человеческих реакций и действий будут востребованы во многих областях знаний, а также многими фирмами и корпорациями.

Сегодня программные реализации моделирования поведения человека уже довольно обширны. Но, все же, их недостаточно для реализации многих задач и учета полного спектра характеристик индивидуума.

Анализируя предметную область, можно говорить о возможности проведения новой разработки прогнозирования поведения групп лиц на основе агентного моделирования и нейронных сетей. В этом случае часть нерешенных вопросов общей проблемы моделирования поведения человека сможет быть решена.

Литература:

1. Карпов В.Э. Коллективное поведение роботов. Желанное и действительное // Современная мехатроника. Сб. научн. трудов Всероссийской научной школы (г.Орехово-Зуево, 22-23 сентября 2011), Орехово-Зуево, 2011, с.35-51, -132с.
2. Петрушан М.В., Самарин А.И. Эволюционная модель поведения агентов в среде, побуждающей к коллективным действиям // Научная сессия МИФИ-2006. Нейроинформатика - 2006. Часть 1 Теория нейронных сетей. Нейробиология. Нейронные сети в задачах обработки изображений. Нейробиология. Нейронные сети в задачах обработки изображений. Модели адаптивного поведения, стр. 196-200
3. VT MAK – Project Humans – DI-Guy SDK [Электронный ресурс]. – Режим доступа: <http://www.mak.com/products/humans>.

4. Макаров О. Игра в людей: Цифровой аналог человечества // Популярная механика. №3 (89), 2010 – с. 78 – 80.
5. The Naval Postgraduate School's MOVES Institute [Электронный ресурс]. – Режим доступа: <https://www.movesinstitute.org/>.
6. Chris Darken: Human Behavior Simulation [Электронный ресурс]. – Режим доступа: <https://www.movesinstitute.org/research/human-behavior-simulation/>.
7. Marco Strano and Roberta Bruzzone, Modeling of Human Behavior in Violent Crimes [Электронный ресурс]. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=A4A285B0DEB8C9E33391781795309D1E?doi=10.1.1.125.4662&rep=rep1&type=pdf>.
8. Инструмент имитационного моделирования AnyLogic: обзор [Электронный ресурс]. – Режим доступа: <http://www.anylogic.ru/overview>.
9. Бекларян А. Л., Акопов А. С. Моделирование поведения толпы на основе интеллектуальной динамики взаимодействующих агентов / «Бизнес-информатика» / № 1 (31) / 2015 / с 69-77
10. Текущее состояние модели поведения человека [Электронный ресурс]. – Режим доступа: <http://www.humanmodel.ru/model/47-current-model?showall=&start=1>.
11. Биологический энциклопедический словарь. Определение «Поведение» [Электронный ресурс]. – Режим доступа: http://dic.academic.ru/dic.nsf/dic_biology/4378.

Сложеницын А.В., Землянская С.Ю., Андриевская Н.К. Выбор структуры модели для компьютеризированной системы прогнозирования поведения групп лиц. Рассмотрено общее состояние направления моделирования поведения человека и поведения толпы. Выявлены достоинства и недостатки существующих моделей и готовых программных комплексов, с точки зрения прогнозирования преступного поведения. Определены основные критерии, необходимые для построения новой модели поведения человека, лишенной недостатков проанализированных продуктов. Выбраны методы и подходы для создания новой модели.

Ключевые слова: моделирование поведения, прогнозирование деятельности, искусственный интеллект, агентное моделирование, нейронная сеть.

Slozhenitsyn A.V., Zemliansky S.Y., Andrievskaya N.K. Model structure choice for the people group behavior forecasting computerized system. General state of human behavior and crowd behavior modeling is reviewed. Advantages and disadvantages of existing models and softwares on their bases are discovered considering behavior forecasting. The basic criteria of human behavior new model building are revealed without shortcomings of previous models. The methods and approaches for new model constructing are chosen.

Keywords: behavior modeling, activities forecasting, artificial intelligence, agent-based modeling, neural network.

Интервальные оценки параметров хаотических чипов

А.И. Андрюхин
Донецкий национальный технический университет
alexandruckin@ramber.ru

Андрюхин А.И. Интервальные оценки параметров хаотических чипов. Эта работа относится к глобальной проблеме построения компьютеров нового типа, которые способны самоорганизовываться и адаптироваться во внешней среде. В работе рассматриваются теоретические обоснования для создания всех основных логических элементов на основе одной хаотической нелинейной системы. Решаемая в статье задача состоит в построении интервальных оценок пороговых параметров использования и управления хаотической временной структуры, генерирующей нелинейные временные ряды с целью получить гибкие и реконфигурируемые вычислительные устройства. Приведены результаты компьютерных расчетов. В расчетах использовался пакет Mathematica.

Ключевые слова: хаотический чип, интервальная оценка, самоорганизация

Введение

Современные компьютеры представляют собой детерминированные системы с очень большим, но конечным числом состояний. Живой организм для существования в изменчивой среде должен обладать элементами хаотического поведения. Искусственные системы, способные эффективно взаимодействовать с меняющейся внешней средой, должны быть в какой-то мере хаотичными.

Создание компьютеров нового типа - открытых систем с точки зрения термодинамики, способных самоорганизовываться и адаптироваться во внешней среде-это задача уже сегодняшнего дня (госпрограмма США BRAIN-2013).

Сейчас на основе динамического хаоса создают аппаратно стохастические чипы. Подчеркнем, что существует большая область построения новых типов транзисторов с использованием современных физических направлений.

Основными из этих направлений являются:

- 1) Транзисторы, управляемые одним электроном Single-electron transistors SET.
- 2) Квантовые клеточные автоматы (QCA-quantum cellular automata) базирующиеся на основе квантовой точки (Quantum dot).
- 3) Туннелированные диоды (RTD-)Resonant tunneling diode
- 4) Устройства на быстрой одноквантовой логике (Rapid single-flux quantum (RSFQ) device)
- 5) Транзистор на основе карбоновых трубок (Carbon nanotube FET)
- 6) ДНК-вычисления (DNA computing)

Некоторые их недостатки и перспективы развития представлены в [1]

Известным направлением является реализация всех основных логических элементов на основе одной нелинейной хаотической системы [2-5]. Известно, что краеугольным камнем современной компьютерной архитектуры является бинарная цифровая логика, логика истинного и ложного. Булева логика отличается своей концептуальной простотой.

Строго показано, что любая логическая операция может быть реализована путем соответствующих соединений адекватного подключения NOR и/или NAND вентилей, т.е. построение универсальной вычислительной техники связано с реализацией фундаментальных NOR и NAND вентилей.

В работе рассматриваются теоретические обоснования для получения всех основных логических элементов на основе одной нелинейной системы.

Решаемая в статье задача состоит в построении интервальных оценок пороговых параметров использования и управления хаотической временной структуры, генерирующей нелинейные временные ряды с целью получить гибкие и реконфигурируемые вычислительные устройства.

Задача создания модели хаотического процессора

Рассмотрим хаотическую чип или процессор, основой которого может быть одномерная система, состояние которой представлена переменной x и динамика которой задается нелинейным отображением $F(X)$.

Постановка подобной задачи предполагает:

- выбор типов аттракторов, пригодных для работы с информацией;
- выбор динамических явлений для реализации базовых операций обработки информации с использованием хаоса;
- разработку принципов, позволяющих ставить во взаимно-однозначное соответствие информацию и траектории динамической системы;
- разработку конкретных математических моделей, позволяющих работать с информацией как с траекториями отображений и
- управлять динамическими явлениями для реализации базовых операций хаотического процессора;
- создание программных комплексов для симуляции хаотических процессоров на компьютерах;
- исследование модели хаотического процессора;
- решение с помощью хаотического процессора сложных задач, плохо решаемых при традиционных подходах.

Решение задачи

Отображение функционирования всех 2-входовых вентилях NAND, NOR, XOR, AND, OR и XNOR включает в себя следующие шаги:

Логические входы I_1 и I_2 для логических операций 2-входовых вентилях кодируют начальное состояние системы X_0 следующим образом:

1) $X_0 \rightarrow x_{gate} + X_1 + X_2$, где физическая величина X_1, X_2 , которой обычно соответствует напряжение, имеет значение 0, когда логические значения входов I_1, I_2 равно 0, и имеет значение L_{in} когда логические значения входов I_1, I_2 равны 1, где L_{in} – ранее оговоренная положительная константа (нижний порог входных сигналов). Аналогично X_1 и X_2 , x_{gate} является пороговым напряжением рассматриваемого вентиля, которое мы можем изменять.

2. Динамическая эволюция в n временных шагов (n -я итерация начального состояния), в результате чего в обновленном состоянии $X \rightarrow F_n(X_0)$, Конкретно мы возьмем $n = 1$, то есть $X = F_1(X_0)$.

3. Определение выхода: логическое значение выхода равен 0, если $F(X_0) < H_{out}$ и значение выхода равен 1, если $F(X_0) > H_{out}$ где H_{out} - ранее оговоренная положительная константа (верхний порог).

Так как F нелинейна, чтобы точно задать начальное X_0 , нужен механизм управления порогом для установки начального X_0 , т.е. определить L_{in} , H_{out} , x_{gate} для каждого вентиля (желательно для упрощения управления L_{in} определять единым).

Для того, чтобы получить все желаемые значения откликов различных вентилях, мы должны с помощью порогового механизма управления удовлетворять условиям, перечисленным в таблице 1 одновременно.

Таблица 1. Логические значения выходов двухвходовых вентилях

I_1	I_2	And	Nand	Or	Nor	Xor	NXor
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

Основные результаты представлены в табл.2 во 2 и 3 строчках, показывающие точные решения для H_{out} , x_{gate} при $L_{in}=1/4$, которые удовлетворяют условиям в таблице 1 и динамическая эволюция определяется логистическим уравнением $F(x) = gx(1-x)$.

На рис. представлены графики отображения $F_n(X)$, где $F_n(X)=F(F_{n-1}(X))$ и здесь параметр нелинейности $g = 4$.

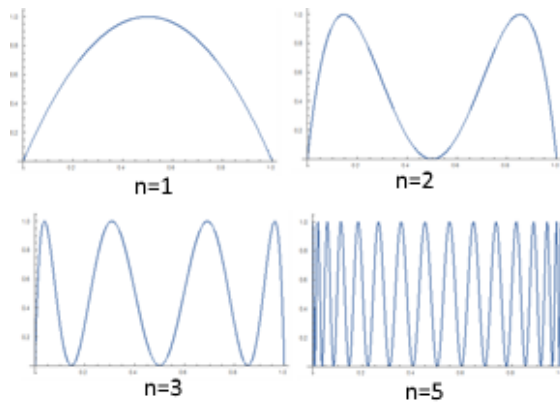


Рисунок 1 – Графики отображения $F_n(X)$.

Возможно использование иных отображений, к примеру $F(X) = 6.5x^2(1-x)$, $x_{n+1} = 2x_n / (1+x_n^{10})$ [5].

Свойства отображения $F(X) = rx(1-x)$.

При интерпретации отображения $F(X) = rx(1-x)$ для эволюции популяции (Ферхольст) имеем, что при изменении значения параметра

- 1). $0 < r < 1$, популяция вымрет, независимо от начальных условий.
- 2). $1 < r < 2$, численность популяции быстро выйдет на стационарное значение $(r-1)/r$, независимо от начальных условий.
- 3). $2 < r < 3$, численность популяции придёт к стационарному значению $(r-1)/r$, но вначале будет колебаться вокруг него. Скорость сходимости линейна везде, кроме значения $r=3$, при котором она крайне мала.
- 4). $3 < r < 1 + \sqrt{6} \approx 3.45$, численность популяции будет бесконечно колебаться между 2 значениями.
- 5). $3.45 < r < \approx 3.54$, численность популяции будет бесконечно колебаться между 4 значениями.
- 6). $r > 3.54$, численность популяции будет колебаться между 8 значениями, потом 16, 32 и т.д.

Длина интервала изменения параметра, при котором наблюдаются колебания между одинаковым количеством значений, уменьшается по мере увеличения r .

Отношение между двумя длинами смежных интервалов стремится к числу Фейгенбаума $\delta \approx 4.669...$

Эта ситуация является типичным примером каскада бифуркаций удвоения периода, который представлен на рис.2.

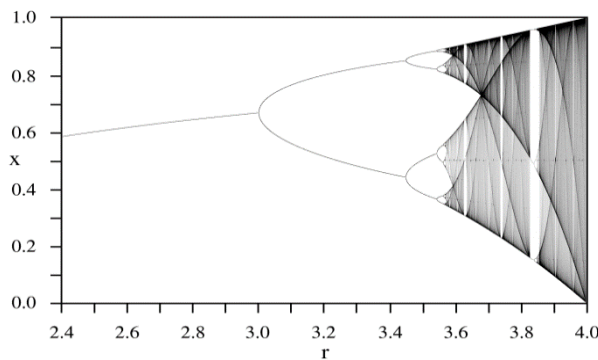


Рисунок 2 – Бифуркационная диаграмма.

При $r \approx 3.57$, начинается хаотическое поведение, а каскад удвоений заканчивается.

Колебания больше не наблюдаются.

Небольшие изменения в начальных условиях приводят к большим отличиям дальнейшего поведения системы во времени.

Это является главным признаком хаотического поведения.

При $r > 3.57$ обычно имеем хаотическое поведение, однако существуют небольшие изолированные интервалы значений r , при которых система ведет себя регулярно,

Их называют «окнами периодичности».

К примеру, начиная со значения $1+\sqrt{8} \approx 3.83$, существует интервал параметров r , при котором наблюдаются колебания между 3 значениями.

Для больших значений r - между 6 значениями, потом 12 и т. д.

В системе можно найти периодические колебания с любым количеством значений.

Последовательность смены количества значений удовлетворяет порядку Шарковского.

При $r > 4$, значения отображения покидают интервал $[0,1]$ и расходятся при любых начальных условиях.

Основные понятия при построении хаотических компьютеров

В табл.2 во 2 и 3 строчках представлены нижний и верхний пороги для соответствующих элементов(операций) и порог $L_{in} = 1/4$, общий для всех логических вентилях. Итеративное применение хаотического логического отображения позволяет использовать эти логические элементы с иной логической функцией, т.е. выполнять их реконфигурацию без дополнительного аппаратного оборудования. Подчеркнем, что начальный нижний порог равен 0.316.

Таблица 2. Значения пороговых параметров двухвходовых вентилях

Операция	Nand	And	Nor	Xor	Or
$L_{op}(x_{gate})$	3/8	0	*	1/4	1/8
$H_{op}(H_{out})$	11/16	3/4	*	3/4	11/16
Итерация	1	2	3	4	5
Состояние чипа	X_1	X_2	X_3	X_4	X_5
Входы (0,0) $X_0=0.316$	0.864	0.468	0.996	0.016	0.063
Логическое значение	1	0	1	0	0
Входы (1,0)/(0,1) $X_0=0.566$	0.982	0.068	0.255	0.760	0.729
Логическое значение	1	0	0	1	1
Входы (1,1) $X_0=0.816$	0.601	0.959	0.155	0.525	0.997
Логическое значение	0	1	0	0	1

Иной подход представлен на рис.3, где реконфигурация выполняется с помощью динамического логического порога C так, что при одном значении C элемент выполняет функцию Nand, а при другом значению функцию Nor и т.п.[5]

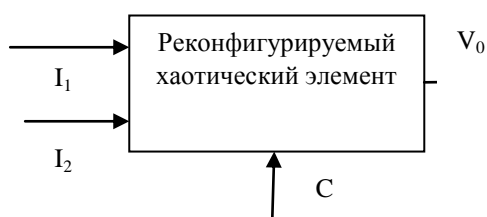


Рисунок 3 –Реконфигурация хаотического элемента сигналом C .

В работе построены интервальные оценки пороговых параметров L_{op} , H_{op} и L_{in} всех двухвходовых вентилях и на рис.4-7 представлены результаты для вентилях NAND и AND [6,7].

Элемент NAND

Функционирование элемента должно удовлетворять следующим условиям

- 1) $F(L_{nand}) - H_{op} > L_{in}$,
- 2) $F(L_{nand} + L_{in}) - H_{op} > L_{in}$,
- 3) $F(L_{nand} + 2L_{in}) < L_{nand}$.
- 4) $H_{op} > L_{nand} + 2L_{in}$

Здесь $L_{nand} = L_{op} = x_{gate}$ и $H_{nand} = H_{op}$.

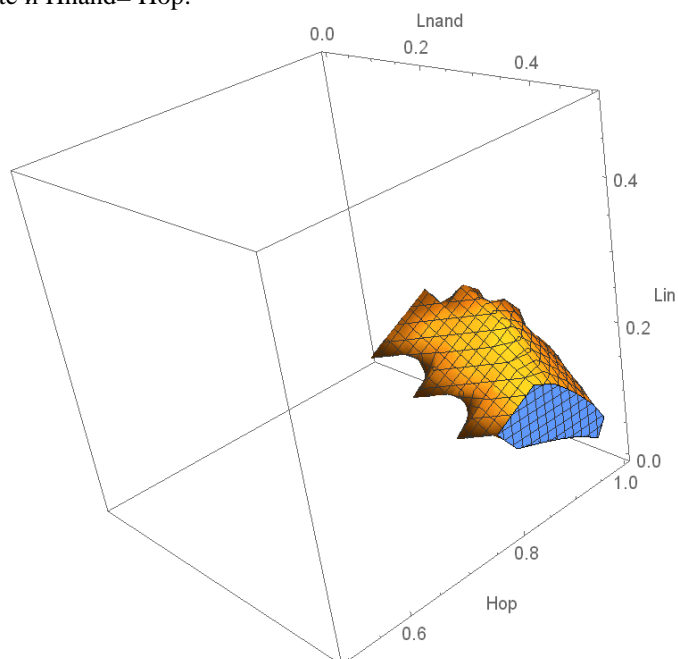


Рисунок 4 – 3-D область возможных решений для элемента NAND на 1 итерации.

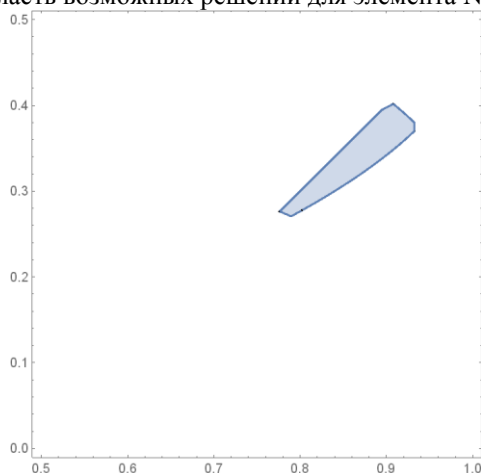


Рисунок 5 – 2-D область L_{op}, H_{op} решений для элемента NAND при $L_{in}=0.25$.

Элемент AND

Функционирование элемента должно удовлетворять следующим условиям

- 1) $F(L_{and}) < H_{op}$,
- 2) $F(L_{and} + L_{in}) < H_{op}$,
- 3) $F(L_{and} + 2L_{in}) - H_{op} > L_{in}$.
- 4) $H_{op} > L_{and} + 2L_{in}$

Аналогично предыдущему случаю, $L_{and} = L_{op} = x_{gate}$ и $H_{and} = H_{op}$.

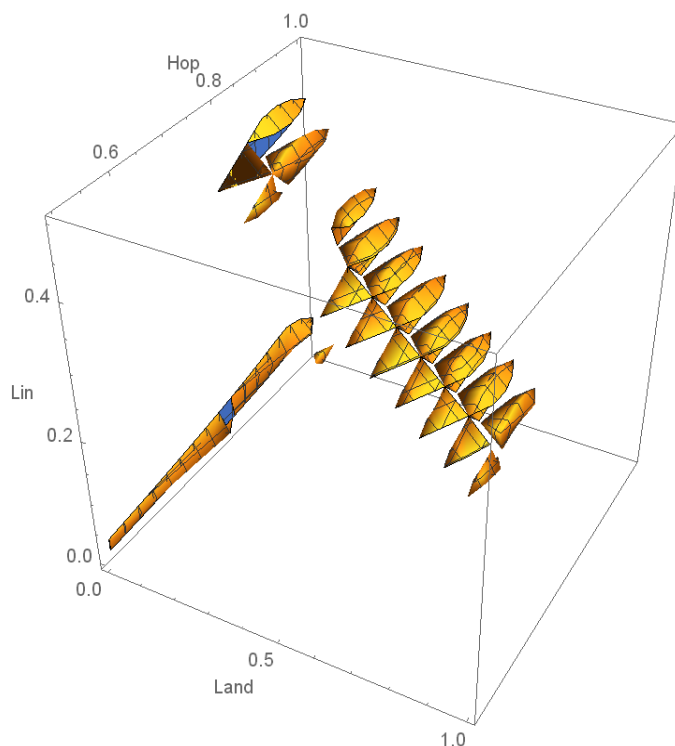


Рисунок 6 – 3-D область возможных решений для элемента AND на 2 итерации.

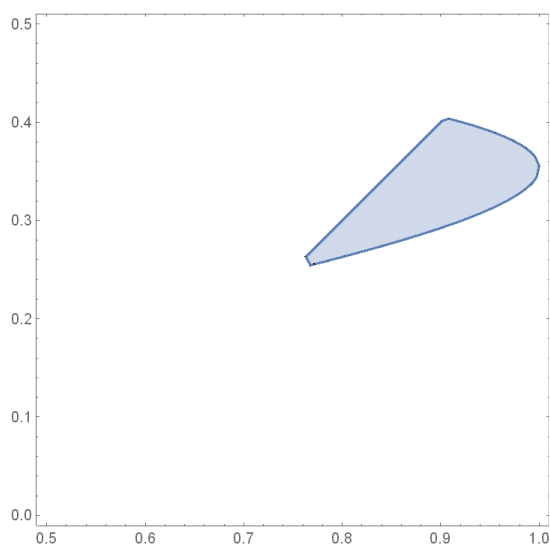


Рисунок 7 – 2-D область решений L_{op}, H_{op} для элемента AND при $L_{in}=0.25$ на 2 итерации.

Аппаратная реализация порогового механизма

Общая схема аппаратной реализации порогового механизма представлена на рис.8 , где V соответствует пороговому значению[2,3].

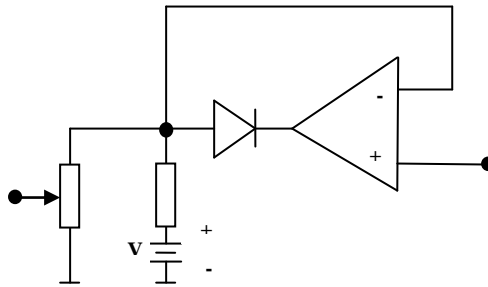


Рисунок 8 - Аппаратная реализация порогового механизма

Выводы

Недавно фирмой ChaoLogix Inc. разработан и изготовлен чип, который демонстрирует возможность построения реконфигурируемых хаотических логических вентилях (стандартная CMOS-технология, 0,18 мкм, процесс TSMC, частота 30 МГц, размер матрицы 3,1 × 3,1 мм, напряжение 1,8 в).

Основной блок ChaoGate схематически показан в [2-3].

ChaoGates включены в массив ChaoGate в чипе СБИС, чтобы продемонстрировать их использование и более высокую функциональность за время, менее чем за один такт:

А) в небольшом АЛУ, (множитель и сумматор).

Б) в протоколах связи (CP), в частности в последовательном периферийном интерфейсе (SPI, синхронный последовательный канал передачи данных) и реализациям шина Inter Integrated Circuit Control (I2C).

В настоящее время усилия сосредоточены на оптимизации конструкции одного ChaoGate до уровней, где они сопоставимы с одним вентилях с точки зрения мощности и размеров и способны менять свое поведение в границах одного тактового цикла.

Предварительные разработки показывают, что эта цель достижима, т.е. все вентилях, используемые для разработки компьютеров, могут быть заменены ChaoGates для обеспечения дополнительной гибкости и производительности.

Литература

1. Wong H.-S. P. Beyond the conventional transistor/ IBM Journal RES. & DEV. Vol. 46 no. 2/3 March/May 2002 .pp.133-168.
2. William L. Ditto, 1 A. Miliotis, 2 K. Murali, 3 Sudeshna Sinha, 4, Mark L. Spano. Chaogates: Morphing logic gates that exploit dynamical patterns // CHAOS 20, 037107 (2010).
3. William L. Ditto, K. Murali and Sudeshna Sinha. Construction of a Chaotic Computer Chip.
4. Ю.В. Андреев, А.С. Дмитриев, Д.А. Куминов. Хаотические процессоры // Успехи современной Радиоэлектроники, No10, 1997
5. Miliotis, K. Murali, S. Sinha, W. L. Ditto, and M. L. Spano, Chaos, Solitons Fractals 42, 809 2009.6.
6. Андрюхин А.И. Оценка вариаций параметров нейроморфных сетей // Наукові праці Донецького національного технічного університету, серія: «Проблеми моделювання та автоматизації проектування» № 1 (10)-2(11), 2012, с.122-131.
7. С.П. Шарый. Конечномерный интервальный анализ, 2013. – 613 с. – <http://www.nsc.ru/interval>

Andruckin A.I. Interval estimates of parameters of chaotic chips. *This work refers to the global problem of building a new type of computers that are able to self-organize and adapt in the environment. The paper discusses the theoretical foundations for the creation of all basic logic elements on the basis of a chaotic nonlinear system. Solved in the article the problem is to construct a threshold interval estimates of parameters of use and management of chaotic time structure generating non-linear time series in order to obtain a flexible and reconfigurable computing devices. The results of computer simulations are presented. The calculations used Mathematica package.*

Keywords: chaotic chip interval estimate, self-organization

Компьютерные расчёты и визуализации характеристик устойчивости систем: проблема центра-фокуса и гармонические функции

Филер З.Е.¹, Андрюхин А.И.²

¹Кировоградский государственный педагогический университет,

²Донецкий национальный технический университет

filier@ramber.ru, alexandruckin@ramber.ru

Филер З.Е., Андрюхин А.И. Компьютерные расчёты и визуализации характеристик устойчивости систем: проблема центра-фокуса и гармонические функции. Эта работа относится к известной проблеме определения параметров устойчивости систем. Теоретические обоснования решения этой задачи рассматриваются для случая центра. Обсуждаются решения для линейных и гармонических функций, а также однородных многочленов высших степеней. Приведены результаты компьютерных расчётов. В расчётах использовался пакет Mathematica.

Ключевые слова: проблема центра-фокуса, устойчивость системы, визуализация, компьютер.

Введение

Общеизвестно влияние результатов А. Пуанкаре по качественному исследованию систем. В частности, они послужили толчком для работ Ляпунова по устойчивости систем и, следовательно, на развитие общей теории управления, теории автоматического управления, теории систем.

С 1880 г. начался цикл работ А. Пуанкаре «О кривых, определяемых дифференциальными уравнениями». В своём «Аналитическом резюме» [1] в 1-ом разделе «Дифференциальные уравнения» в подразделе Y он даёт анализ своих работ в этом направлении. На с. 597-598 он пишет: «... я начал свои исследования ... с изучения кривых, определяемых уравнениями вида

$$\frac{dx}{X} = \frac{dy}{Y},$$

где X, Y - ... многочлены от x и y ... [оно] привело к различению 4 видов особых точек ... Я дал [им] ... названия: 1) седла... 2) узлы ... 3) фокусы ... наподобие логарифмической спирали...4) центры, где все кривые, ... замкнуты, наподобие кривых уровня топографического плана лишь в крайне редких случаях). Далее, на с. 599, он пишет: «Имеется частный случай.... ... необходимо бесконечное число условий ... это чаще всего ... доказывається, что все условия удовлетворены, ... когда ...

$$\frac{\partial X}{\partial x} + \frac{\partial Y}{\partial y} = 0 \text{»}.$$

Здесь нет ни одной «картинки» - в докомпьютерную эру построение графиков решений было весьма трудоёмкой работой.

В нашей статье мы попытаемся обобщить эти результаты и использовать современные средства, предоставляемые ЭВМ.

1. Случай линейных функций X, Y

Он рассматривается практически во всех пособиях по дифференциальным уравнениям [2-5]. Иногда это изложение занимает много страниц. Заменяя уравнение $\frac{dx}{X} = \frac{dy}{Y}$ системой уравнений $\dot{x} = X, \dot{y} = Y$,

получим для $X = ax + by, Y = cx + ey$ характеристическое уравнение $\det(A - \lambda E) = 0$. Оно является

квадратным относительно λ . След матрицы $A = \begin{pmatrix} a & b \\ c & e \end{pmatrix}$ $\text{Sp}A = a + e$ и её определитель $\det(A) = ae - bc$

определяют тип особой точки $O(0, 0)$. **Центр** будет при чисто мнимых корнях характеристического уравнения, когда $\text{Sp}A = a + e = 0$, а $\det(A) = ae - bc = \omega^2 > 0$. Тогда x и y будут периодическими функциями от t частоты ω (с периодом $2\pi/\omega$). При *комплексных* корнях этого уравнения, когда $\text{Sp}A \neq 0$ и $\det(A) = ae - bc > 0$ налицо **узел**. Все кривые будут подобными ввиду однородности уравнения. В других случаях будут узлы и седла. Если записать это уравнение в дифференциалах $Ydx - Xdy = 0$, то сказанное Пуанкаре условие $X_y + Y_x = 0$ является условием полного дифференциала левой части уравнения. Таким образом, уравнение имеет полный интеграл $cx^2 + 2axy + by^2 = f$ при постоянных действительных коэффициентах. Если $bc > a^2, b > 0, f > 0$, то оно описывает эллипс. Точка O является **центром**.

Переход к комплексной форме для функции $\dot{z} = (a + bi)z + iAz$ при $\sqrt{a^2 + b^2} < A$ даёт центр при любых a, b, A . У этой формы есть простой механический смысл. Вектор касательной к траектории при $a = b = 0$ перпендикулярен к радиусу - вектору каждой её точки. Такой линией является окружность. При $a \neq 0, b \neq 0$ траекторией является эллипс.

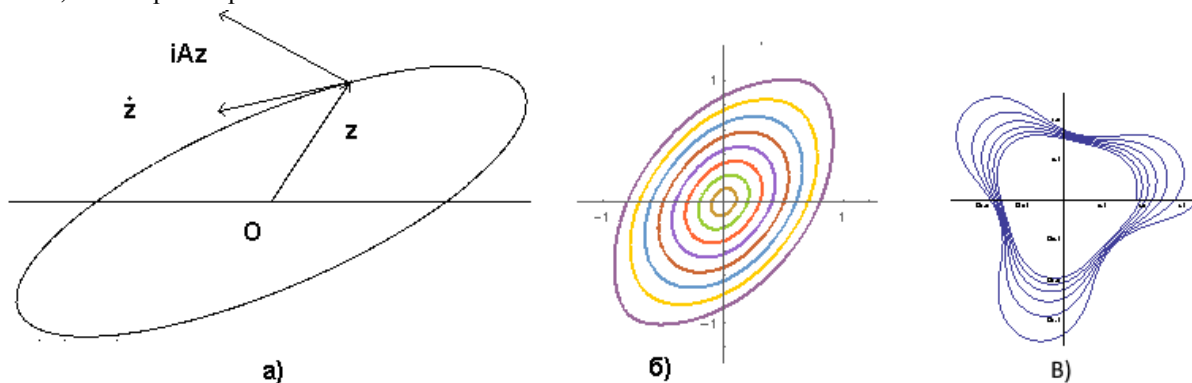


Рисунок - 1 а) Механическое истолкование центра - скорость движения точки z под действием сил iAz и az ; б) *Центр* системы концентрических эллипсов в) фокус при $n = 4$ и $a=1, b=0.8, A=1.68, r_0=0.9, B=0.01$.

В [5, с.128-129] по поводу центра читаем: «... случай центра: при малом изменении элементов матрицы ... центр перейдет в ... фокус. Включение этого вырожденного случая ... в основной текст ... объясняется его важностью». Периодические решения уравнения 2-го порядка изображаются замкнутыми линиями на фазовой плоскости.

2. Случай однородных многочленов высших степеней

Для системы $\dot{x} = X, \dot{y} = Y$ удобнее в этом случае перейти к полярным координатам $x = r \cos \varphi, y = r \sin \varphi$. Если многочлены X, Y **гармоничны**, то есть лапласиан от них равен нулю, правые части являются линейными комбинациями косинусов и синусов аргумента $n\varphi$, умноженных на r^n . Умножая уравнения на x и y соответственно и складывая, получим систему $\dot{r} = (xX + yY)/r, \dot{\varphi} = (xY - yX)/r^2$. Она равносильна уравнению

$$\frac{dr}{d\varphi} = r \frac{Y \sin \varphi + X \cos \varphi}{Y \cos \varphi - X \sin \varphi}, X = r^n (a \cdot \cos(n\varphi) + b \cdot \sin(n\varphi)), Y = r^n (c \cdot \cos(n\varphi) + e \cdot \sin(n\varphi))$$

Правая часть уравнения имеет точки бесконечного разрыва, в которых тригонометрическое уравнение $Y \cos \varphi - X \sin \varphi = 0$ имеет корни. Добавив в знаменатель число A , что эквивалентно добавлению в правые части соответственно членов $-Ay$ и Ax , получим при выборе постоянной $A > \max[\sqrt{a_k^2 + b_k^2}], k = 1, 2$, где a_k, b_k - коэффициенты указанных линейных комбинаций.

В этом случае при использовании комплексной переменной $z = x + iy$ получим уравнение $\dot{z} = \alpha_n z^n + iAz$. Комплексный коэффициент $\alpha_n = a_n + ib_n$ выбирается так, чтобы выполнялись условия центра: $|\alpha_k| < A$. Соответствующее уравнение будет

$$\frac{dr}{d\varphi} = r \frac{Y \sin \varphi + X \cos \varphi}{Y \cos \varphi - X \sin \varphi + A}, X = r^n (a \cdot \cos(n\varphi) - b \cdot \sin(n\varphi)), Y = r^n (b \cdot \cos(n\varphi) + a \cdot \sin(n\varphi)). \quad (2)$$

Оно содержит лишь 3 произвольных постоянных a, b, A , которые могут изменяться и не мало, но согласованно.

В декартовых координатах в левой и правой точках с вертикальными касательными приходится переходить от уравнения $\frac{dy}{dx} = \frac{Y}{X}$ к уравнению $\frac{dx}{dy} = \frac{X}{Y}$. Поэтому удобно для построения траекторий переходить к *полярным* координатам.

На рис. 2-4 представлены центры с различными n . Нижние рисунки показывают изменение радиуса $r(\varphi) - r(0)$. На рис. 1в) показан неустойчивый фокус.

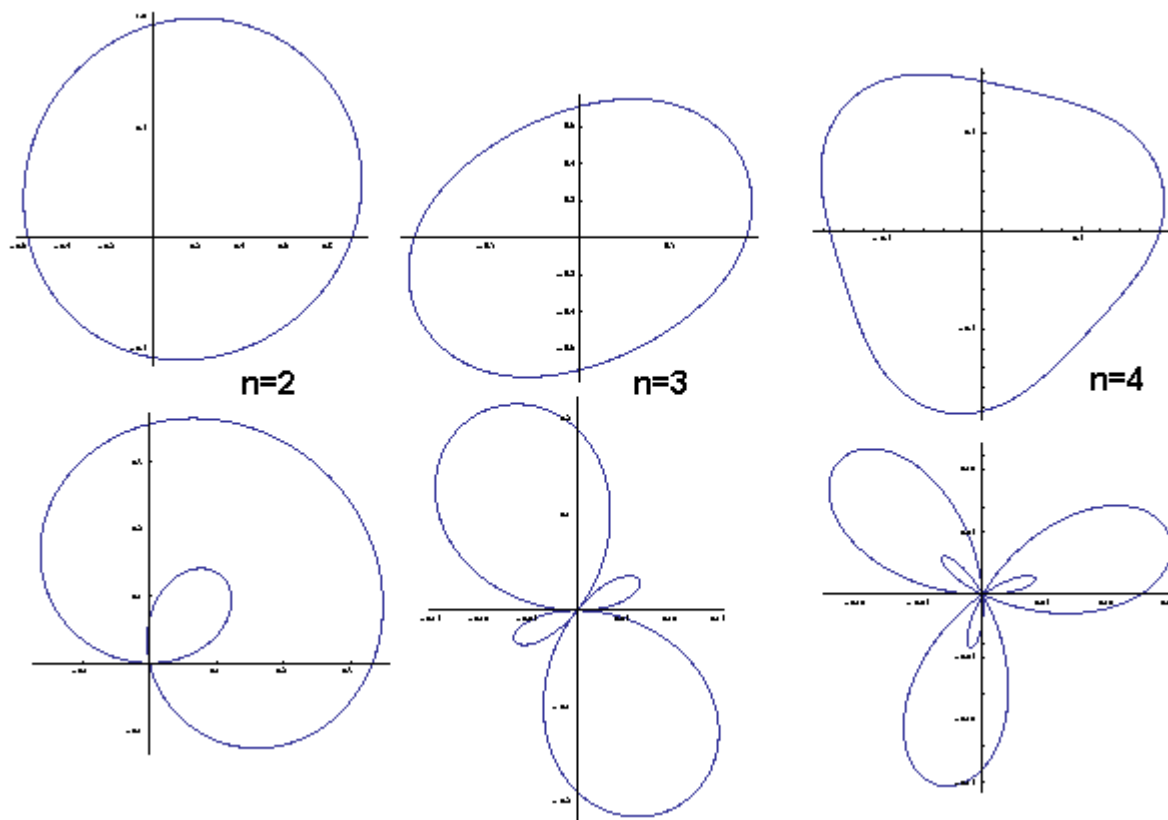


Рисунок 2 - Коэффициенты имеют значения: $a = 1, b = 8, A = 1.68$.

«Атомарные» гармонические многочлены являются «кирпичиками», из которых строятся функции, обеспечивающие появление центра. Примерами их при $n = 3$ являются 2 конкретных многочлена 3-ей степени: $X_3 = x^3 - 3xy^2, Y_3 = 3x^2y - y^3$; при $n = 4$ имеем многочлены $X_4 = x^4 - 6x^2y^2 + y^4, Y_4 = 4x^3y - 4xy^3$. Они являются действительными и мнимыми частями степени $z^n = (x + iy)^n$ при n , соответственно равном 3 и 4.

3. Случай гармонических функций

Для комплексной аналитической функции $f(z) = \sum_{k=0}^{\infty} c_k z^k$ центр находится в точке О при $c_0=0$; член $c_1 z$ должен иметь вид $(a+bi)z + iAz$. Коэффициенты $a, b, A, a_k, b_k, k \geq 2$ произвольны, но сумма модулей коэффициентов сходящегося ряда должна быть меньше А.

Для известных гармонических функций можно пользоваться привычными формулами, например, для функции $f(z) = e^z - 1 - z$ будем иметь центр в точке О, если взять $(a+ib)f(z) + iAz$ с достаточно большим $A \gg \sqrt{a^2 + b^2}$. Действительной частью этой функции является $X = e^x \cos(y) - 1 - x$, мнимой – функция $Y = e^x \sin(y) - y$. Добавление в состав одного из слагаемых, которые не входят в ряд гармонических членов, делает траекторию отличной от центра, ибо в числителе появляется постоянная составляющая, ведущая к росту модуля производной.

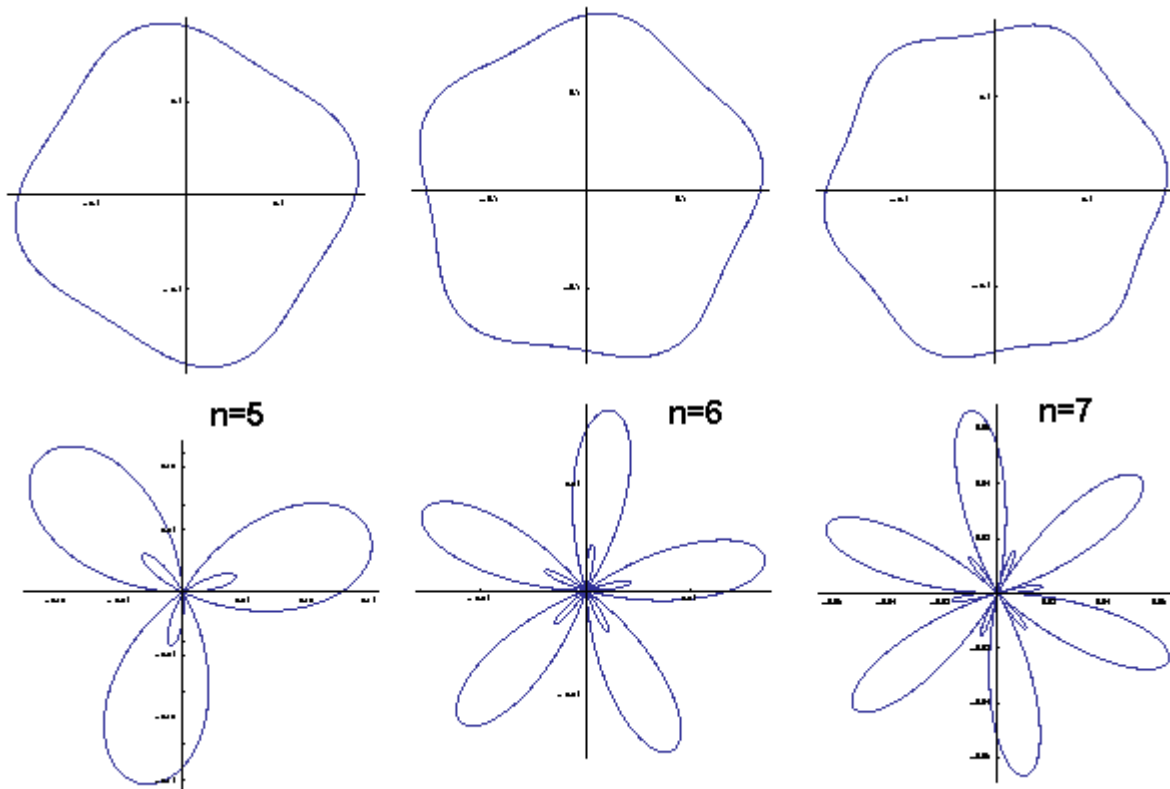


Рисунок 3 - Центры при $n = 5, 6, 7$. Количество «лепестков» и «сторон» многоугольника равно $n-1$.

В уравнении (2) необходимо взять вместо x, y выражения $r \cdot \cos \varphi, r \cdot \sin \varphi$ соответственно, а член A заменить на Ar .

На рис. 5а построены траектории центров для системы уравнений $\dot{x} = e^x \cos(y) - x - 1, \dot{y} = e^x \sin(y) - y$, а на рис. 5б – для системы $\dot{x} = \sin(x)ch(y) - x, \dot{y} = \cos(x)sh(y) - y$.

В этих системах правые части являются действительными и мнимыми частями комплексных функций $e^z - z - 1$ и $\sin(z) - z$ соответственно. Вычитаемые x и y необходимы, чтобы члены соответствующих рядов начинались с $n = 2$. Без них особая точка О будет фокусом.

Выводы

1. В нашей работе берутся члены ряда из однородных многочленов, как и в других работах [3 - 5], но эти члены гармоничны. Может этим и объясняются сложные необходимые условия в форме связей между коэффициентами в статьях Альмухамедова и И.С. Куклеса [3, С. 87]. Между тем, в работах А.Пуанкаре есть

условие центра $\frac{\partial X}{\partial x} - \frac{\partial Y}{\partial y} = 0$, вытекающее из условий полного дифференциала, совпадающее с одним из условий Коши – Римана для гармонических функций.

2. В работе используются современные программные комплексы, позволяющие проверять и иллюстрировать полученные аналитически результаты.

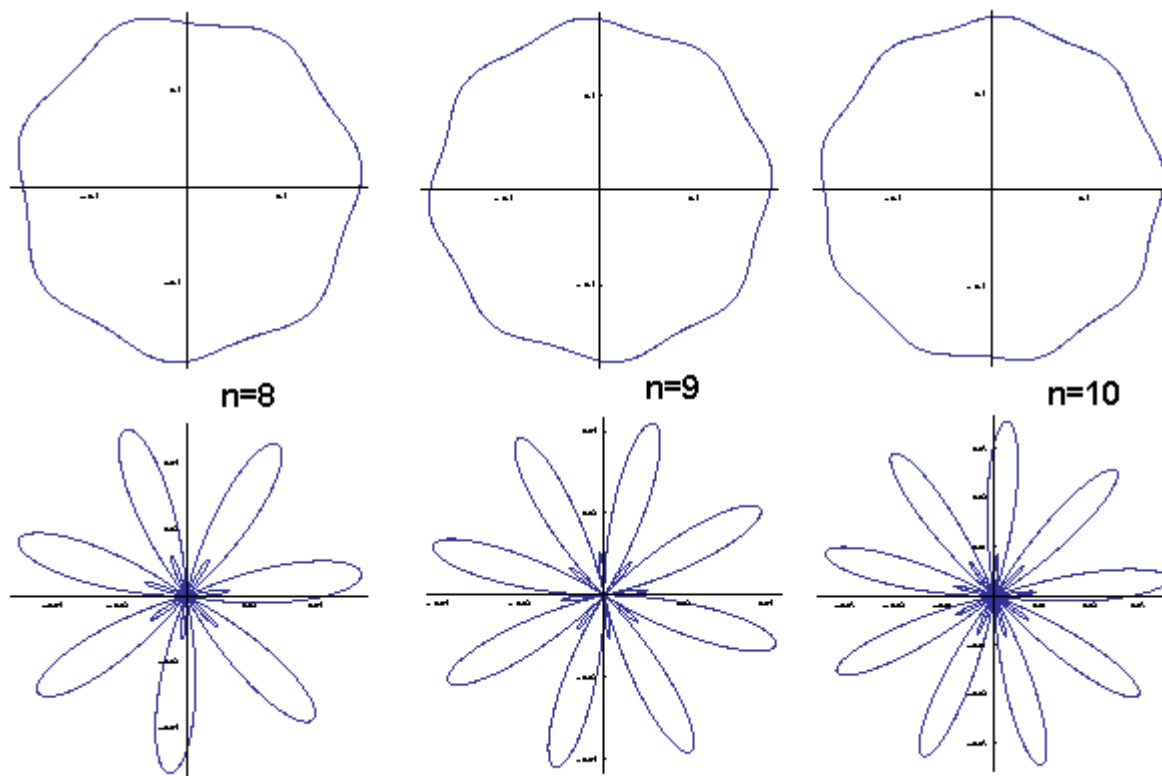


Рисунок-4. Центры при $n = 8, 9, 10$. Число «лепестков» и «сторон» многоугольника также равно $n - 1$.

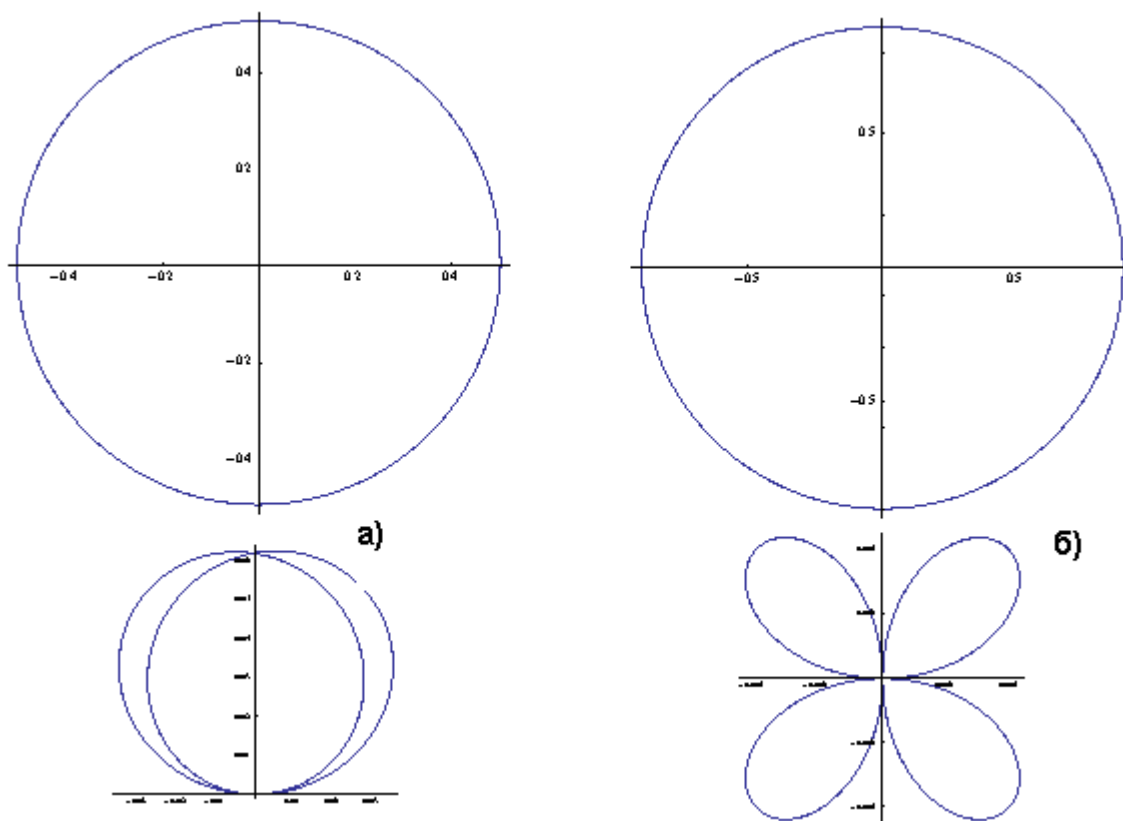


Рисунок-5а) $f(z) = e^z - z - 1$; 5б) $f(z) = \sin(z) - z$.

Литература

1. Пуанкаре А. Избранные труды в 3 томах. Том III. Математика. Теоретическая физика. Анализ математических и естественно - научных работ Анри Пуанкаре. М.: Наука, 1974. – 772 с.
2. Пуанкаре А. О кривых, определяемых дифференциальными уравнениями. – М.-Л.: ГИТ-ТЛ, 1947. – 392 с.
3. Немыцкий В.В и Степанов В.В. Качественная теория дифференциальных уравнений. М.-Л.: ГИТ-ТЛ, 1947. - 448 с.
4. Курс обыкновенных дифференциальных уравнений/ Н.П.Еругин, Й.З.Штокало, П.С.Бондаренко, И.А.Павлюк др. – К.: Вища школа, 1974. – 472 с.
5. Понтрягин Л.С. Обыкновенные дифференциальные уравнения. – М.: ГИФ-МЛ, 1961. – 312 с.

Филер З.Е., Андрюхин А.И. Компьютерные расчёты и визуализации характеристик устойчивости систем: проблема центра-фокуса и гармонические функции. Эта работа относится к известной проблеме определения параметров устойчивости систем. Теоретические обоснования решения этой задачи рассматриваются для случая центра. Обсуждаются решения для линейных и гармонических функций, а также однородных многочленов высших степеней. Приведены результаты компьютерных расчётов. В расчётах использовался пакет Mathematica.

Ключевые слова: проблема центра-фокуса, устойчивость системы, визуализация, компьютер.

Filer Z.E., Andryukhin A.I. Computer calculations and visualization systems stability characteristics: the center-focus problem and analytical functions. This work refers to the well-known problem of determining the parameters of the systems stability. Theoretical basis of solving this problem are being considered for the center of the case. Solutions for linear and harmonic functions, as well as the homogeneous polynomials of higher degrees are discussed. Computer calculation results are given. The calculations used Mathematica package.

Keywords: center-focus problem, system stability, visualization, computer.

О мере количественной информации в пространствах состояний сложных систем

Г.В. Аверин, А.В. Звягинцева
averin.gennadiy@gmail.com, zviagintsevaav@gmail.com

Аверин Г.В., Звягинцева А.В. О мере количественной информации в пространствах состояний сложных систем. Предлагаются подходы к изучению феномена информации, исходя из применения мер информации при создании моделей описания темпоральных данных, характеризующих сложные системы. Выполнено экспериментальное обоснование справедливости закона сохранения меры количественной информации на основе использования существующих естественнонаучных представлений о континуальности пространств состояний сложных систем различной природы. В качестве примера представлены возможности использования предложенного метода при описании опытных данных, характеризующих физическую и биологическую системы. Научная значимость метода связана с возможностью построения на основе феноменологического подхода информационных моделей для различных классов сложных систем.

Ключевые слова: сложные системы, меры информации, законы сохранения, феноменологические модели.

Введение

Представления об информации во многом формируют содержание актуальных направлений современной науки. Однако, изучение множества публикаций по вопросу о сущности информации формирует чувство глубокой неудовлетворенности, так как однозначного и ясного определения этого понятия до сих пор дать никому не удалось. Сегодня распространено мнение, что информация наряду с материей и энергией является первичной категорией и одним из атрибутов реального мира и поэтому в строгом смысле не может быть определена. В этом плане академик В. Глушков дал очень общее определение информации как меры неоднородности распределения материи и энергии в пространстве и времени, меры изменений, которыми сопровождаются все протекающие в мире процессы.

Исключительность информации связана как с ее универсальностью, так и специфичностью по отношению к различным областям знаний. Возможно, поэтому формулировки данного понятия отличаются чрезвычайной противоречивостью. До настоящего времени большинство авторов, формулирующих понятия «информация», не ставили перед собой задачу построения общей теории информации. Поэтому нет смысла искать что-то общее во множестве существующих определений, в первую очередь, следует научиться измерять и оценивать информацию применительно к объектам и процессам различной природы. При этом следует придерживаться логики, позволяющей развить основные понятия, такие как референты информации, данные, события, знания и т.д. по отношению к реальным процессам для обеспечения формализованного описания наиболее общих, но простых случаев представления информации о физических, биологических и социальных системах.

Общим предметом изучения различных наук служат наблюдаемые факты изменений в состояниях объектов и систем, представляющие собой закономерный результат различных видов взаимодействий в природе и обществе. С помощью измерений дается количественная характеристика свойств объектов путем определения значений параметров в той или иной системе единиц; наблюдения преимущественно позволяют устанавливать факты (события, эффекты, явления), свойственные различным изменениям в состояниях систем. Полученная количественная информация о сложных системах часто хранится в форме массивов данных вида «объекты–показатели–время», при этом данные об объектах при моделировании параметризуются и представляются в многомерных информационных пространствах. Такая идеальная информационная модель может быть применена к множеству простых случаев, характеризующих процессы развития и поведения различных физических, биологических и социальных систем. Задачи данного класса связаны с созданием временных (темпоральных) баз данных (БД), представляющих собой целое направление в теории СУБД [1]. Темпоральные реляционные базы данных используют трехмерные таблицы для хранения информации. При

этом реляционные таблицы БД можно рассматривать как частный случай темпоральных таблиц данных. В теории такая логическая модель данных имеет непосредственное отношение к многомерным информационным пространствам, в которых атрибуты объектов в виде количественных показателей соответствуют атрибутивным переменным в принятых системах координат. В свою очередь, при совершении объектами процессов время выступает общим параметром по отношению к данным переменным. Состояния объектов в таких пространствах представляются многомерными точками, а процессы изменения состояний – многомерными кривыми. Если предложить меру (меры) информации для таких пространств и использовать принципы непрерывности и континуальности их представления, то можно установить законы сохранения для такой меры – своего рода логические аналоги законов сохранения величин в естествознании. Некоторые авторы считают, что отсутствие в современной теории информации законов сохранения можно рассматривать как свидетельство незавершенности этой теории.

Постановка задачи

Универсальность закона сохранения энергии, согласно теореме Нётер, является следствием принципа однородности времени, что определяет независимость от момента времени законов, характеризующих поведение физических систем. В свою очередь, Пуанкаре, а также Фальк и Юнг отмечали, что уравнение сохранения энергии является математическим выражением, отражающим континуальные закономерности пространств состояний физических систем [2, 3]. Данное уравнение изначально обосновано в термодинамике экспериментально и потом логически распространено на физические системы.

Указанные принципы свойственны многим сложным системам, так как однородность времени и континуальность пространств состояний систем являются общими закономерностями для множества процессов и явлений, наблюдаемых в природе и обществе. Исходя из этого, примем гипотезу, что для систем различной природы могут существовать скалярные величины, которые являются функциями состояния систем и для которых могут быть сформулированы законы сохранения этих величин, справедливые в определенной предметной области. В данном случае нельзя оперировать понятием «энергия», которое свойственно предметной области физики, а следует использовать более общие категории.

Для того, чтобы получить общий логический аналог «закона сохранения энергии» для биологических, экономических или социальных систем необходимо использовать понятия, которые применимы во многих областях знаний. Естественно в данном случае говорить об информационной мере по отношению к состояниям объектов, специфичной для каждого класса систем.

Используем информационный подход и связанные с ним понятия и определения, позволяющие провести формализацию задач моделирования состояний систем в информационных пространствах переменных. Только выполнив изучение самых различных способов измерения и оценки информации для множества простых случаев, можно приступить к анализу сущности феномена информация.

Поэтому целью данной статьи является экспериментальное обоснование справедливости закона сохранения меры количественной информации на основе использования существующих естественнонаучных представлений о континуальности пространств состояний сложных систем различной природы.

Для решения этой задачи будем опираться исключительно на анализ опытных данных и естественнонаучные представления о процессах изменения состояний систем во времени. Покажем, что в различных предметных областях информационный подход и системные аналогии, связанные с логикой построения феноменологических моделей, позволяют предложить общие методы описания структурированных опытных данных темпорального вида и сформулировать законы сохранения скалярных величин для систем различной природы.

О представлении информации при описании сложных систем

Одна из фундаментальных проблем науки связана с количественными свойствами информации. Существующие научные представления в данной области определяют во многом формирование оснований теории информации, кибернетики, семиотики, теоретической информатики и т.д. В других областях знаний также существуют свои определения информации, однако в этих областях пока не ставится задача формализации понятия информация. В настоящее время имеется несколько десятков определений понятия «информация», наиболее известными из которых являются определения К. Шеннона и У. Эшби. Согласно К. Шеннону под информацией понимают снятую неопределенность наших знаний об объектах, процессах и явлениях, в то время как по определению У. Эшби информация – это мера изменения во времени и пространстве структурного разнообразия систем. Анализируя существующие формулировки проф. И. Бекман приводит более 80 определений [4]. Различные точки зрения о сущности информации исходят из того, что она является неотъемлемым свойством материи, субъективной реальностью, формой отражения материи, информационным содержанием живой и неживой природы, результатом взаимодействия материальных систем, особым видом энтропии, целевой функцией и т.д. Понятие информации является настолько сложным и многоаспектным, что на современном этапе развития науки ему крайне тяжело дать общее определение. Тем не

менее, существуют различные подходы к оценке и измерению информации и, соответственно, способы введения меры информации. Под мерой информации обычно понимают некоторую непрерывную действительную аддитивную функцию, определенную на множестве событий. Одна из таких мер в виде информационной энтропии была использована К. Шенноном в теории информации путем установления связей с вероятностями состояний систем.

Понятие информационной энтропии введено в научный обиход, исходя из аналогий с обобщенным определением энтропии в статистической термодинамике, которое сформулировано Планком. Исходя из этого,

К. Шенноном зависимость для энтропии $h(z) = -\sum_{i=1}^p w_i \cdot \log w_i$ (w_i – вероятность i -того состояния системы) без

теоретического обоснования и по соглашению распространена на любые физические системы, исходя из аналогий и соображений простоты определения энтропии.

Теория информации основана на вероятностных закономерностях, свойственных процессам и явлениям. Однако множество ситуаций не укладываются в информационную модель Шеннона. Большой проблемой является нечеткость определения понятий состояния физической системы и вероятности ее состояния. Не всегда представляется возможным установить перечень всех состояний системы, а также оценить их вероятности. Также слабо прослеживается связь информационной энтропии с разнообразными событиями и свойствами, несущими в себе информацию о состоянии и поведении систем.

Понятие информации имеет непосредственное отношение как к состояниям сложных систем, так процессам изменения этих состояний. При этом характеристиками информации могут выступать и сведения в виде данных, и различные факты в виде событий. Нести в себе информацию могут также и знания, которые в идеальном случае представляются моделями объектов, процессов и явлений. Таким образом, состояния систем и связанные с этим понятия и закономерности формируют статические представления, а понятия, определения и законы, относящиеся к процессам, могут формировать динамические представления о сущности информации. Исходя из этого, в теории должны быть как модели, характеризующие информационное пространство состояний систем, так и модели, позволяющие описывать процессы в этом пространстве. Для привнесения в теорию реального содержания исходные информационные и математические модели должны основываться на использовании величин, которые позволяют верифицировать полученные зависимости по отношению к данным опыта. Соответствующие величины часто носят феноменологический характер, и их использование позволяет получать адекватные модели различных систем, которые уже несут в себе элементы знаний.

В контексте вышесказанного в настоящей статье под данными подразумеваем любые сведения об объекте исследования, его свойствах, параметрах и состоянии, а также окружающей среде в конкретный момент времени. Данные несут в себе статическую часть информации об объекте и выступают в качестве одного из ее референтов.

В качестве второй характеристики информации будем рассматривать любые факты об объекте исследования – сведения и сообщения о каком-либо событии, процессе или деятельности, свойственном объекту и наблюдаемом с течением времени. Все такие факты будем называть событиями, несущими в себе динамическую часть информации об объекте.

В свою очередь, под знаниями в идеальном простом случае будем понимать достоверное и адекватное модельное описание объекта исследования, процессов его развития или поведения, отражающее основные особенности и закономерности и предоставляющее новую информацию об объекте. При этом модельные описания понимаем в широком смысле как вербальные, логические, информационные, математические и т.п. модели.

Считаем, что данные, события и знания в комплексе будут нести полную информацию об объекте исследования и определять качественные уровни информации о системе в целом. При этом данные позволят создать формализованное информационное пространство, а вероятностный подход при анализе событий, выделенных по заданному признаку, даст возможность ввести меру информации, определенную на множестве таких событий. Связь между мерой и данными позволит создать вероятностные модели объектов исследования и, тем самым, формировать новые знания.

Таким образом, референтами информации выступают данные, события и знания, формирующие различные иерархические уровни ее представления, где каждый следующий уровень отражает новые качества по отношению к предыдущему. В какой-то степени такое представление имеет отношение к информационным DKIW-моделям, используемым в теории информации.

Для применения подобного подхода при определении меры информации необходимо предложить типовую натурную модель сложной системы, которая может быть использована по отношению к объектам и процессам различной природы.

Будем представлять физическую, информационную, природную или социальную систему в виде совокупности однородных объектов, например, изделий, устройств, установок, организаций, предприятий, природных и техногенных объектов, регионов, стран, граждан, и т.д. В качестве параметров объектов могут использоваться различные величины физического, информационного, экономического, социального и др. характера. Все объекты системы имеют равное число характерных свойств, которые определяются

наблюдаемыми параметрами z_1, z_2, \dots, z_n и формируют количественные данные о системе в целом. Каждый объект совершает естественный процесс изменения или развития, в связи с чем его параметры зависят от времени $z_1(t), z_2(t), \dots, z_n(t)$. В результате наблюдений за объектами системы формируется массив темпоральных данных, где каждая таблица имеет структуру «объекты – параметры», а множество таблиц упорядочено во времени с заданным шагом.

Данной логической модели данных поставим в соответствие идеальную модель информационного пространства. Рассмотрим n -мерное пространство $H^n\{z_1, z_2, \dots, z_n\}$, охватывающее все наблюдаемые состояния объектов. Будем подразумевать под состоянием каждого объекта совокупность его наблюдаемых параметров z_1, z_2, \dots, z_n , тогда все состояния в каждый момент времени будут отображаться точками $M = M(z_1, z_2, \dots, z_n)$. Процессы изменения состояний объектов в пространстве H^n будут иметь вид кривых для точек M .

Для построения моделей описания данных, информация о которых может быть представлена в приведенном выше виде, используем принципы моделирования, сформулированные в работах [5 – 7].

Теперь для представления закона сохранения меры количественной информации с целью упрощения и наглядности рассмотрим систему, состояния которой определяются двумя переменными. При этом воспользуемся основными идеями теории системодинамики [5]. Энтропию состояния будем рассматривать как функцию состояния, которая определяется параметрами этого состояния. В данном случае для любой точки M дифференциал энтропии может быть представлен в виде [5 – 7]:

$$ds = c_1 \cdot \frac{dz_1}{z_1} + c_2 \cdot \frac{dz_2}{z_2}. \quad (1)$$

В свою очередь, в пространстве переменных H^2 в качестве некоторого индекса, также характеризующего состояние объекта, зададим меру относительных изменений или геометрическую вероятность точки M , которые будут иметь вид:

$$\theta = \frac{z_1 \cdot z_2}{z_{1*} \cdot z_{2*}}, \quad z_{k*} = z_{k,0} \quad \text{или} \quad z_{k*} = z_{k,m}. \quad (2)$$

Данная величина в информационном пространстве будет формировать определенную непрерывную среду моделирования, по отношению к которой предполагается описывать процессы изменения состояний объектов. Функции вида (2) могут иметь самый разный вид, исходя из особенностей описания поведения системы [5, 7].

Вероятность состояния объекта будем определять алгоритмически по отношению ко всей группе изучаемых однородных объектов, исходя из идей работ [5, 7]. Энтропию состояния будем рассматривать как одну из мер оценки и измерения количественной информации. С учетом связи энтропии состояния и его вероятности $dw = \theta \cdot ds$ получим аналог закона «сохранения энергии» для систем различной природы, состояния которых описываются двумя показателями [5].

Сделав обозначение, $R_* = z_{1*} \cdot z_{2*}$ определим изменение вероятности состояния $dw = \theta \cdot ds$, тогда:

$$dw = \theta \cdot ds = \theta \cdot \left(c_1 \cdot \frac{dz_1}{z_1} + c_2 \cdot \frac{dz_2}{z_2} \right) = \frac{1}{R_*} (c_1 \cdot z_2 \cdot dz_1 + c_2 \cdot z_1 \cdot dz_2).$$

Преобразуя данное уравнение, получим зависимость:

$$dw = du + \frac{z_2 \cdot dz_1}{R_*}, \quad (3)$$

где изменение скалярной величины du равно: $du = \frac{1}{R_*} (c_1 \cdot z_2 \cdot dz_1 + c_2 \cdot z_1 \cdot dz_2) - \frac{1}{R_*} z_2 \cdot dz_1$.

Применяя к величине u признак Эйлера для пфаффовых форм, получим, что du является полным дифференциалом (функцией состояния) при выполнении следующего условия:

$$c_1 - c_2 = 1. \quad (4)$$

В этом случае $du = c_2 \cdot d\theta$, $di = d(u + \theta) = c_1 \cdot d\theta$ и величины u и i зависят только от величины θ , характеризующей среду моделирования. Величина u является аддитивной функцией и может также выступать в качестве меры количественной информации.

В результате данного простого вывода получены в виде уравнений (3) и (4) логические аналоги уравнения сохранения энергии и уравнения Майера в термодинамике для безразмерных переменных. Следует отметить, что найденные соотношения справедливы для любых континуальных пространств состояний систем независимо от природы изучаемых данных. Поэтому вполне естественно, что суть величин u и i далека от соответствующих аналогов (энергия и энтальпия) в термодинамике. Это же относится и к уравнению (3), которое нельзя рассматривать как уравнение сохранения энергии в обычном физическом представлении.

Отметим, что уравнения вида (3) и (4) можно получить для трех и более переменных z_k , например, так как это сделано в работе [5]. Теперь используя форму закона сохранения скалярной величины u попытаемся обработать существующие опытные данные для некоторых систем различной природы.

Примеры построения информационных моделей для сложных систем

Если приведенные выше гипотезы являются справедливыми, то на основе обработки опытных данных можно получить эмпирические аналоги уравнений (3) и (4) и тем самым установить для изучаемой сложной системы закон сохранения некоторой фундаментальной скалярной величины.

Физическая система. Для построения информационной модели системы, состоящей из химических элементов периодической таблицы Менделеева, примем в качестве параметров: z_1 – радиус атома, пм; z_2 – атомную массу элемента, а.е.м.

Рассмотрим сложное совместное событие одновременного наблюдения этих показателей в пространстве состояний системы. Найдем статистическую вероятность этого события на основе опытных данных, которые имеются в базе данных, используя алгоритмы сортировки и группировки данных для таблицы «объекты-показатели» и оценивая относительные частоты событий [5, 7]. После этого ищем эмпирическую связь между статистической вероятностью событий и энтропией состояния системы с помощью методов пробит-анализа [8]. Значения постоянных коэффициентов c_k определяются в процессе построения регрессионной зависимости.

Для периодической таблицы Менделеева регрессионное уравнение получено в виде пробит-зависимости вероятности совместных событий от энтропии состояния системы:

$$w = \frac{1}{\sqrt{2 \cdot \pi}} \int_{-\infty}^{\text{Pr}} \exp\left(-\frac{t^2}{2}\right) dt; \quad \text{Pr} = -4,030 + s; \quad s = 1,109 \cdot \ln\left(\frac{z_1}{z_{1H}}\right) + 0,540 \cdot \ln\left(\frac{z_2}{z_{2H}}\right). \quad (5)$$

Коэффициент корреляции зависимости (5) составил 0,96, ошибка – 10,2%, результаты обработки данных для 90 химических элементов приведены на рисунке 1. Показатели z_1, z_2 относились к значениям z_{1H}, z_{2H} , которые соответствуют свойствам водорода и равны: $z_{1H} = 53$ пм.; $z_{2H} = 1,0078$ а.е.м.

Из уравнения состояния (5) видно, что $c_1 - c_2 = 0,569$. Так как соотношение (4) не выполняется ($c_1 - c_2 \neq 1$) уравнение (3) будет давать определенные погрешности. Поэтому, используя вид уравнения сохранения (3) определим его параметры путем обработки всего массива данных для пространства состояний, образованного двумя свойствами химических элементов. Результаты обработки данных приведены на рисунке 2, соответствующее уравнение в конечно-разностной форме имеет вид:

$$\theta \cdot \Delta s = \Delta u + 5,304 \cdot \frac{z_2}{R_*} \cdot \Delta z_1 - 56,38, \quad (6)$$

где $\Delta s = s - s_0$; $\Delta u = c_2 \cdot (\theta - 1)$; $\Delta z_1 = z_1 - 53$ пм – приращения величин относительно опорной точки (свойства водорода); $R_* = 53,41$ пм·а.е.м.

Значение величин s и u для водорода приняты равными нулю. Коэффициент корреляции для уравнения (6) составляет 0,997, относительная ошибка – 6,5%.

Уравнение (6) может быть приведено к общему виду уравнения сохранения энергии $dw = du + z_2 \cdot dz_1$ путем выбора наиболее оптимальных значений параметров в опорной точке и установления эквивалентности между величинами. Анализ показывает, что подобные результаты могут быть получены для комбинаций из различных двух параметров свойств химических элементов.

Биологическая система. Покажем, что предложенные методы могут найти применение при оценке биологических организмов в информационных пространствах их показателей. С этой целью используем массивы информации из базы данных AnAge Database [9], которая представляет собой результат большой работы многих ученых. Нынешняя версия базы включает сведения о 4083 видах позвоночных и охватывает количественные характеристики амфибий, рептилий, рыб, птиц и млекопитающих. В базу внесены данные о продолжительности жизни, массе тела при рождении и во взрослом состоянии, скорости роста, интенсивности метаболизма, а также некоторые другие характеристики (всего более 25 показателей). Имеется информация о систематике видов позвоночных животных.

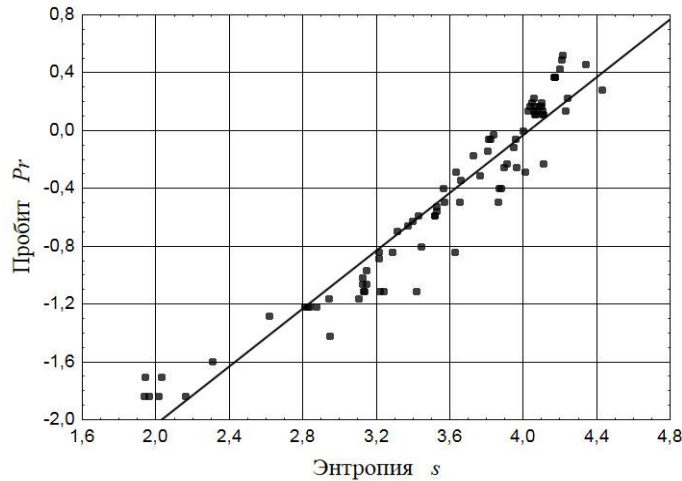


Рисунок 1 – Зависимость вероятности w от величины s для радиуса атома и атомной массы элементов согласно уравнения (5)

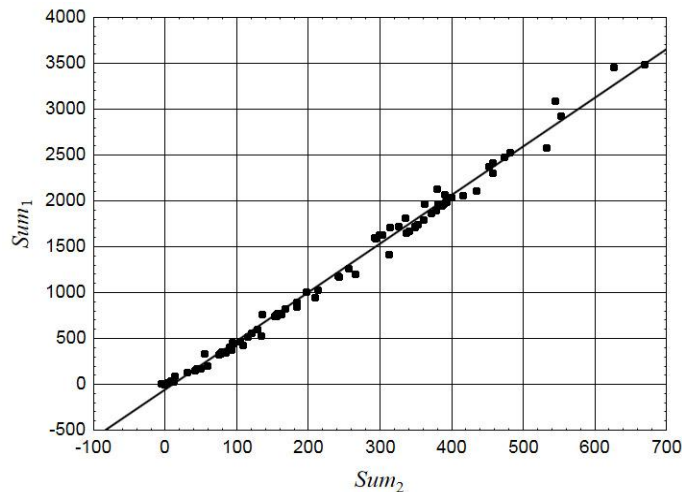


Рисунок 2 – Результаты обработки данных согласно уравнения (6):
 $Sum_1 = \theta \cdot \Delta s - \Delta u$; $Sum_2 = (z_2/R_*) \cdot \Delta z_1$

В качестве показателей биологических видов для построения уравнения состояния примем: максимальную продолжительность жизни в неволе z_1 , лет; вес взрослой особи z_2 , кг.

Регрессионное уравнение для описания показателей видов животных получено в виде пробит-зависимости вероятности совместных событий наблюдения показателей z_1, z_2 от энтропии состояния [10]:

$$w = \frac{1}{\sqrt{2 \cdot \pi}} \int_{-\infty}^{\text{Pr}} \exp\left(-\frac{t^2}{2}\right) dt; \text{Pr} = 3,767 + s; s = 0,778 \cdot \ln\left(\frac{z_1}{z_{1,m}}\right) + 0,184 \cdot \ln\left(\frac{z_2}{z_{2,m}}\right). \quad (7)$$

Коэффициент корреляции зависимости (7) составил 0,97, относительная ошибка – 9,8 %. Результаты обработки данных для 2548 видов животных приведены на рисунке 3.

Показатели z_1, z_2 относились к величинам $z_{1,m}, z_{2,m}$, которые соответствуют максимально наблюдаемым показателям видов и соответственно равны: $z_1 = 211$ лет; $z_2 = 136000$ кг.

Получим на основе обработки опытных данных эмпирический аналог уравнения (3) и тем самым установим некоторый закон сохранения для изучаемой биологической системы. Из уравнения состояния (7) следует, что $c_1 - c_2 = 0,594$. Используя вид уравнения сохранения (3) определим параметры уравнения для двух биологических показателей позвоночных животных.

Результаты обработки данных даны на рисунке 4, а соответствующее уравнение в конечно-разностной форме имеет вид:

$$\theta \cdot \Delta s = \Delta u - 0,466 \cdot \frac{z_2}{R_*} \cdot \Delta z_1 - 7,4 \cdot 10^{-7}, \quad (8) \quad \text{где}$$

$\Delta s = s - s_0$, $\Delta u = u - u_0 = c_2 \cdot \theta$, $\Delta z_1 = 211 - z_1$ (лет) – приращения величин относительно максимально наблюдаемых значений переменных; $R_* = 28,7 \cdot 10^6$ (лет·кг). Значение величин s_0 и u_0 при геометрической вероятности $\theta = 1$ приняты равными нулю. Коэффициент корреляции для уравнения (8) составляет 0,95, средняя относительная ошибка – 7,2 %.

Полученные соотношения позволяют количественно сравнивать между собой биологические виды по совокупности физиологических показателей.

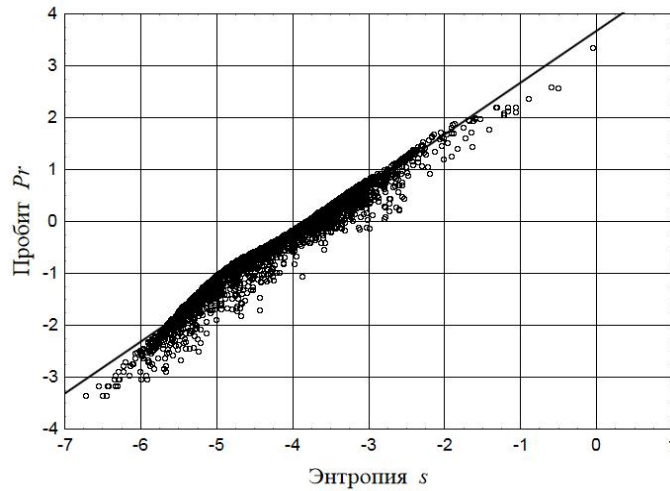


Рисунок 3 – Зависимость вероятности w от величины s для продолжительности жизни и веса позвоночных животных согласно уравнения (7)

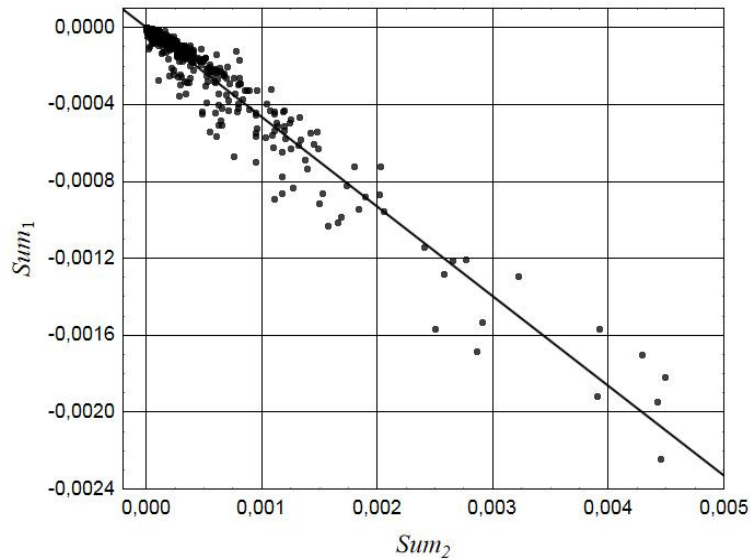


Рисунок 4 – Результаты обработки данных согласно уравнения сохранения (8):

$$Sum_1 = \theta \cdot \Delta s - \Delta u ; Sum_2 = (z_2 / R_*) \cdot \Delta z_1$$

Таким образом, базы данных информации в комплексе с феноменологическими методами описания данных позволяют построить информационные модели систем и предложить логические аналоги законов сохранения в предметных областях.

Сравнительная оценка мер количественной информации

В заключение сформулируем несколько дискуссионных вопросов в области теории информации, которые требуют обоснования в свете дальнейших исследований. Самый важный вопрос дискуссии связан с проблемой: какая величина или система величин наиболее полно отражает содержание информации о состояниях объектов и их изменениях во времени и может выступать адекватной мерой количественной информации? Речь может идти об энтропии s пространства состояний системы вида (1), скалярной величине $u = c_2 \cdot \theta$, которая входит в уравнение сохранения (3), а также информационной энтропии Шеннона $h(z)$. Естественно, что такой сложный феномен, как информация, может характеризоваться множеством мер.

Полученные выше логические аналоги закона сохранения «энергии» для физической и биологической систем в виде соотношения (3) указывают на существование меры для информационного пространства состояний любой системы, если для этого пространства свойственны непрерывные закономерности. Однако, получаемые закономерности будут отражать особенности предметной области и формулируемые при этом определения, положения и соотношения только математически будут соответствовать общепринятым в естествознании и, в частности, в термодинамике. Поэтому соотношение (3) будет носить специфический характер для каждой сложной системы, а также определенной комбинации ее параметров z_k . Скорее всего, мер количественной информации для пространств состояний сложных систем может быть бесконечно много.

Информационная энтропия $h(z) = -\sum_{i=1}^p w_i \cdot \log w_i$ (p – количество опытных точек) качественно отличается от предлагаемых мер количественной информации, т.к. определяется только числом возможных состояний и вероятностями этих состояний. Для каждого состояния M можно предложить частную энтропию в виде $h_i = -w_i \cdot \log w_i$, однако данная величина применительно к информационному пространству H^n не будет аддитивной, так как ей присуща явная нелинейность в связи с использованием вероятности состояния w_i .

Энтропия s свойственна всему информационному пространству и отражает в своей сущности существующие непрерывные закономерности. Данная величина позволяет построить в пространстве H^n естественные криволинейные координаты [5, 11].

В свою очередь, скалярная величина u представляется математической функцией, характеризующей изменение состояния объекта по отношению к однородному информационному пространству, в котором отсутствовали бы непрерывные закономерности, статистически свойственные сложной системе. Данная величина является аддитивной и также может выступать в виде меры количественной информации. Из приведенного выше материала видны явные аналогии, связанные с построением феноменологических моделей в термодинамике. Однако, логически переносить представления о фундаментальности величины u , которые сложились в физике в виде энергетической концепции, в теорию информатики, пока преждевременно. Для такого вывода пока недостаточно обработано опытных данных применительно к различным сложным системам.

Выводы

Предложенный метод построения феноменологических моделей применительно к темпоральным массивам данных позволяет применить в информатике существующие принципы математического моделирования систем и процессов на основе законов сохранения скалярных величин. Научная значимость метода связана с возможностью построения на основе общего подхода феноменологических моделей для различных классов сложных систем. Это актуальное направление исследований позволяет развить методы теоретической информатики.

Однако сложность задачи модельного описания данных для систем различной природы состоит в том, что теория должна опираться на обширные опытные данные. Уже видна обширность такой задачи, так как для систем различной природы необходимо построить множество моделей и в каждом конкретном случае сформулировать закон сохранения некоторой фундаментальной скалярной величины. Примеры таких моделей, приведенные в данной статье, указывают на реальность решения такой научной задачи. Экспериментальная проверка сформулированного закона сохранения меры количественной информации по отношению к различным сложным системам позволит сделать окончательный вывод о его фундаментальности.

Литература

- Костенк Б.Б., Кузнецов С.Д. История и актуальные проблемы темпоральных баз данных // Citforum. – 2007. – Режим доступа: <http://citforum.ru/database/articles/temporal/> (30.10.16).
- Пуанкаре А. О науке / Пер. с франц. М.: Наука, 1983. – 560 с.
- Falk G., Jung H. Axiomatik der Thermodynamik // Hdb. Phys. III/2, Berlin, 1959. – pp. 119 – 175.
- Бекман И.Н. Информатика. Курс лекций. – Режим доступа: <http://profbeckman.narod.ru/InformLec.htm> (27.10.16).

Аверин Г.В. 2014. Системодинамика. Донецк: Донбасс. – 405 с. – Режим доступа: <http://www.chronos.msu.ru/ru/trules/item/sistemodinamika-2> (02.11.16).

Аверин Г.В., Константинов И.С., Звягинцева А.В. О континуальном подходе к модельному представлению данных // Вестник компьютерных и информационных технологий, 2016. № 10. – С. 47–52.

Звягинцева А.В. Вероятностные методы комплексной оценки природно-антропогенных систем / Под науч. ред. д.т.н., проф. Г.В. Аверина. – М.: Издательский дом «Спектр», 2016. – 258 с.

Bliss C. The method of probits // Science V. 79, no. 2037, 1934. pp. 38 – 39. DOI: 10.1126/science.79.2037.38.

AnAge: The Animal Ageing and Longevity Database. Available at: <http://genomics.senescence.info/species/> (accessed August 29, 2016).

Г.В. Аверин, А.В. Звягинцева. Построение уравнений состояний сложных систем на основе событийной оценки индикативных показателей. Научные ведомости Белгородского государственного университета. Сер. Экономика. Информатика. Выпуск 40, 2016.

Аверин Г.В. О принципе существования и законе возрастания энтропии в свете общесистемных представлений системодинамики // Системный анализ и информационные технологии в науках о природе и обществе, 2015, № 1(8)–2(9). – С. 12 – 42. – Режим доступа: <http://sait.csm.donntu.org/> (02.11.16).

Аверин Г.В., Звягинцева А.В. О мере количественной информации в пространствах состояний сложных систем. Предлагаются подходы к изучению феномена информации, исходя из применения мер информации при создании моделей описания темпоральных данных, характеризующих сложные системы. Выполнено экспериментальное обоснование справедливости закона сохранения меры количественной информации на основе использования существующих естественнонаучных представлений о континуальности пространств состояний сложных систем различной природы. В качестве примера представлены возможности использования предложенного метода при описании опытных данных, характеризующих физическую и биологическую системы. Научная значимость метода связана с возможностью построения на основе феноменологического подхода информационных моделей для различных классов сложных систем.

Ключевые слова: сложные системы, меры информации, законы сохранения, феноменологические модели.

Averin G.V., Zviagintseva A.V. On the quantitative information measure in the complex systems spaces. On the quantitative information measure in the complex systems spaces. The approaches to the study of the information phenomenon, based on the application for information in the creation of temporal data models descriptions complex systems describing. An experimental study of justice the law of conservation measures quantitative information based on the existing natural scientific ideas use about the continuity of the different nature complex systems spaces. As an example, presents the possibility of using the proposed method in describing the experimental data characterizing the physical and biological systems. The scientific importance of the method associated with the approach possibility based on fenomenogicheskogo information models for different complex systems classes.

Keywords: complex systems, information measures, conservation laws, phenomenological models.

СЕКЦИЯ «СОВРЕМЕННЫЕ ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ»

УДК 004.432.2

Проектирование и разработка компилятора С-подобного языка программирования общего назначения с поддержкой исполнения кода на этапе компиляции

Н.М. Ткачёв, А.И. Андрюхин
Донецкий национальный технический университет
m4dbrat@gmail.com, alexandruckin@rambler.ru

Ткачёв Н.М., Андрюхин А.И. Проектирование и разработка компилятора С-подобного языка программирования общего назначения с поддержкой исполнения кода на этапе компиляции. В статье представлен краткий обзор современных принципов проектирования языков программирования и сформулированы требования к проектируемому языку. Спроектирована начальная итерация языка программирования общего назначения и кратко описана структура разрабатываемого компилятора. Перспективами развития языка является расширение системы типов и внедрение в компилятор комплексных средств анализа кода для предотвращения ошибок и исполнения кода на этапе компиляции.

Введение

Задача проектирования новых языков программирования в наше время является как никогда актуальной, поскольку широкое распространение компьютерных технологий ставит задачи максимальной оптимизации процесса разработки и написания кода в различных предметных областях. Именно из-за широкого использования тех или иных технологий становятся более заметными их недостатки, потому новые средства призваны их исправлять [1].

Высокоуровневые средства программирования значительно ускоряют решение типичных задач, но зачастую их гибкости может не хватать, поскольку обстоятельства решаемой задачи требуют более точного контроля над ситуацией. В частности, наиболее распространённой является проблема производительности, когда управление абстракциями приводит к вычислениям, не влияющим на конечный результат. Другой проблемой являются «протекающие абстракции» - явление, при котором программисту приходится отвлекаться от решения задачи на попытки понять внутреннее устройство используемых им средств. Как правило, такая проблема возникает при недостаточном внимании при проектировании средств разработки или несовместимости парадигм среды разработки, языка и библиотек. Например, Win32 API пытается сочетать объектно-ориентированную модель на основе обмена сообщениями вместе с низкоуровневым процедурным стилем языка С, из-за чего программисту приходится выполнять множество неочевидных действий и увеличиваются временные затраты на отладку.

Язык программирования является низшим слоем, на котором держится процесс разработки, потому важно убедиться в том, чтобы разработчики могли чётко выражать в нём свои намерения (абстракции и алгоритмы, оперирующие ими), по минимуму отвлекаясь на языковые особенности («бороться» с языком).

Цель работы: исследование современных принципов проектирования языков программирования и их проблем, проектирование языка на их основе полученных сведений, а также разработка и развитие компилятора этого языка.

1 Цели при проектировании языков программирования

Ввиду развития и распространения множества различных парадигм программирования был выделен ряд требований к языкам, выполнение которых позволяет оптимизировать процесс разработки.

1. Язык программирования должен иметь синтаксис и модель работы, которые позволяют максимально просто записывать некоторые манипуляции и в то же время позволяют чётко их видеть при чтении кода. Такие языки, как Java, могут требовать написания сотен строк повторяющегося кода для решения задач, которые языки наподобие Haskell могут решить в одну строку. Однако необходимо помнить, что более краткий код может использовать более сложные понятия, потому на его понимание может уйти гораздо больше времени, чем на чтение аналогичного, но более длинного кода. Баланс между этими сторонами должен соблюдаться как программистами, так и проектировщиками языка. Наиболее распространённой формой языка в этом плане являются языки с C-подобным синтаксисом и объектно-ориентированной моделью с некоторыми элементами функционального программирования (например, lambda-выражениями и замыканиями).

2. Язык, спроектированный для определённой предметной области, должен позволять прозрачно манипулировать понятиями данной области. Язык общего назначения должен быть достаточно гибким для формирования абстракций различных предметных областей, а также давать возможность построения связей между этими областями.

3. Язык должен предотвращать типичные для языков более низкого уровня логические ошибки, превращая их в семантические. Обычно к этой цели стремятся языки с гибкой системой типов, позволяя программистам описывать допустимые и недопустимые связи между объектами в определённых общих понятиях.

4. Абстракции, встроенные в язык, должны нести минимальную нагрузку, необходимую для их применения. Язык C++ работает на принципе «платишь только за то, чем пользуешься», позволяя, например, отказаться от таблицы виртуальных функций при использовании классов [2], если программисту не нужен полиморфизм, а только наследование и инкапсуляция. Язык Rust создаёт новую модель управления памятью, которая предотвращает типичные ошибки в языках с ручной моделью (такие как попытки доступа к преждевременно уничтоженным объектам), при этом не используя сборки мусора или автоматического подсчёта ссылок. Используемая в этих целях абстракция lifetime [3] (время жизни объекта) является zero-cost abstraction (абстракция с нулевой стоимостью), поскольку с её помощью такие моменты, как оптимальное время удаления объекта, полностью рассчитываются на этапе компиляции без влияния на производительность конечной программы, в то же время увеличивая производительность разработки за счёт сокращения этапа отладки.

Как видно по примеру с Rust, последние два пункта тесно связаны с вычислениями на этапе компиляции. Перенос решений как можно большего количества задач с конечной программы на компилятор улучшает производительность как самой программы, так и процесса отладки, потому является перспективным направлением исследований.

2 Структура разрабатываемого компилятора

Общая структура программы описана на рис. 1. Каждый компилятор имеет 3 основные фазы [4].

1. Разбор исходного кода. Данная фаза состоит из двух процессов: лексический и синтаксический анализ. Лексический анализатор разбивает исходную программу на поток токенов (последовательность идентификаторов, ключевых слов, операторов и констант), после чего синтаксический анализатор проверяет соответствие образуемых конструкций грамматике языка. При отсутствии ошибок на этом этапе возможно формирование однозначной информации об описанных объектах и осуществляемых над ними действиях. Разрабатываемый компилятор использует связку генераторов Flex + GNU Bison для автоматизации разработки этих анализаторов, поскольку они позволяют описать грамматику языка в декларативной форме.

2. Семантический анализ. На этой стадии осуществляются такие операции, как построение промежуточного представления, сверка с таблицей символов, проверка типов, анализ потока управления, предварительная оптимизация и прочее. Основные цели: убедиться, что описанные действия верны в понятиях описанных абстракций, и определить корректные действия в понятиях генерируемой программы. Эти операции производятся с помощью таблицы символов путём анализа деревьев выражений. Каждый описанный тип данных в анализируемой программе сохраняется как объект подклассов Type (рис. 2). Аналогично хранятся инструкции (Statement), выражения (Expression) и прочие элементы.

3. Генерация кода программы. Действия в понятиях программы переводятся в понятия платформы, на которой будет работать целевая программа. На этой фазе могут осуществляться оптимизации, специфичные для данной ОС и/или архитектуры процессора. Традиционные компиляторы генерируют программы в ассемблерных инструкциях целевой платформы, затем используют программу-ассемблер для сборки рабочей программы. Перспективным направлением является компиляция кода в ANSI C, поскольку компиляторы языка C доступны на большом количестве платформ, а их встроенные оптимизаторы позволяют получить производительность, близкую к написанному вручную ассемблерному коду.



Рисунок 1 – Схема архитектуры компилятора

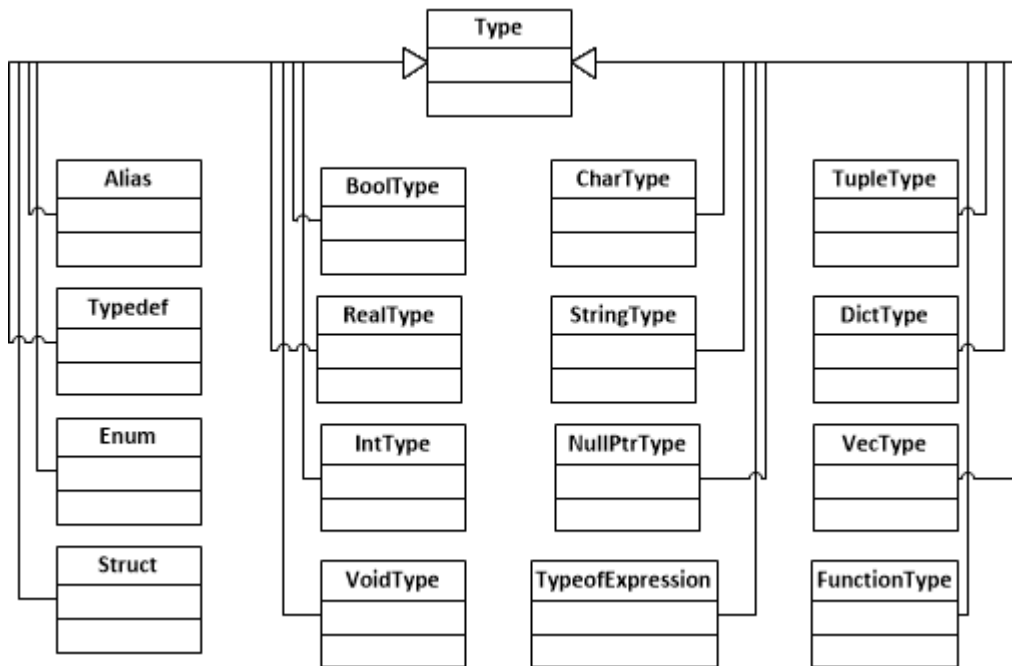


Рисунок 2 – Потомки класса Type, представляющие собой различные стандартные типы и способы построения пользовательских типов

3 Примеры конструкций проектируемого языка

На рис. 3 рассмотрен пример исходного кода. Синтаксис и многие конструкции языка являются производными от C-подобных языков, но с некоторыми структурными отличиями. Из данных примеров видны следующие отличия.

1. Директива `import` указывает компилятору путь к модулям, символы которых необходимо добавить в

глобальное пространство имён. Директива позволяет перечислять символы в иерархическом порядке, таким образом разрешая указать несколько символов в единожды указанном модуле. При импорте компилятор проходит соответствующий путь в файловой системе, пытаясь найти файл интерфейса указанного модуля. Файлы интерфейса генерируются при компиляции вместе с исходным кодом на C.

2. Типы-эnumерации (enum) являются алгебраическими типами данных, или помеченными объединениями (type-tagged union). Каждый вариант является не только некоторым значением, но ещё и сопровождающей это значение структурой данных. Все варианты могут иметь параметры разных типов. При использовании объектов таких типов в кратком синтаксисе скрывается множество операций по проверке истинного типа объекта, и компилятор гарантирует, что все операции над этими объектами типобезопасны.

3. Условные переходы, циклы и блоки кода вместо инструкций являются выражениями. Т. е. они имеют некоторое значение, вычисляемое кодом внутри них. Это позволяет: записывать некоторые операции более лаконично, сохранять чистоту внешней области видимости от временных переменных и обеспечивать константность переменных, требующих сложный процесс инициализации.

4. Цикл for имеет более простую форму записи, которая указывает имена индексной переменной и итератора, а также диапазон обхода без излишней информации о внутренних условиях работы цикла.

Кроме того, имеются ссылочные типы (в качестве абстракции над указателями) и чёткое разделение между типами nullable (которые могут иметь нулевой указатель в качестве значения) и non-nullable. Компилятор будет требовать проверку на null в случае попыток операций над объектами nullable-

```
// Исходный текст программы начинается с директив import, добавляющих
// в таблицу символов определения из других модулей.
import std:(io, stdlib.malloc);

// Определить новый тип на основе одного предыдущего
typedef Integer = Int;

// Определение структуры
struct Point3D {
    x, y, z : Real; // Поля
}

// Эnumерация
enum Color {
    Red, Green, Blue, // Обычные варианты
    RGB(red : Int, green : Int, blue : Int) // Параметризованный
    вариант
ТИПОВ.}
// Определение функции
func main(args : Vec[String]) : Int {
    var eight = 3 + 5; // Автоматически принимает тип Int
    var nine : Real = eight + 1; // Преобразуется к Real из Int

    // Конструкция if может использоваться как выражение
    // Доступ к элементам массива происходит через "("
    // Оператор "." для доступа к свойствам
    var condInt = if (args(1).isEmpty) eight else nine;

    for (i,v in 1..10) { // аналог for (int i=0, v=1; v <= 10; i++,
v++)...
        print(i, " : ", v, "\n");
    };

    var point : Point3D = (3, 4, 0.2);
    var gray = RGB(128, 128, 128); // Тип - Color

    return 0;
}
```

Рисунок 3 – Пример исходного кода на проектируемом языке

4 Перспективы развития

Язык создан с намерением его расширения функциональными и объектно-ориентированными возможностями с целью исследования возможностей вычислений на этапе компиляции.

Среди планируемых возможностей являются такие распространённые возможности современных языков, как интерфейсы, полноценная поддержка шаблонов (в примере кода виден стандартный шаблон Vec, который является абстракцией над статическими и динамическими массивами), lambda-выражения (как минимум в форме inline-операций), контрактное программирование и выполнение произвольного кода на этапе компиляции.

Объектная модель, в которой в основе будут лежать интерфейсы, сходна с моделью языка Rust [5]. В языке Rust отдельно описываются структуры данных (struct), интерфейсы (trait) и реализации (impl), что отличается от традиционного объектно-ориентированного подхода, в котором классы объединяют в себе все три элемента. Возможность описывать реализации различных интерфейсов для различных структур данных открывает возможности для полиморфизма без явных признаков наследования («приоритет композиции вместо наследования» как один из принципов ООП подсвечивает одну из проблем этой парадигмы). При этом возможен больший фокус на статическом полиморфизме вместо динамического (компилятор может «развернуть» абстракцию и подобрать правильные операции без использования таблиц виртуальных функций, и таким образом предотвратить лишние ошибки динамического приведения типов).

Выполнение произвольного кода на этапе компиляции (compile-time function execution, CTFE [6]) также позволит повысить производительность конечной программы и обнаружить многие ошибки до её запуска. Задачи, которые можно выполнить единожды на машине разработчика, можно убрать с каждого запуска программы пользователем. Часть кода, которая требует вызова функций операционной системы для вычисления необходимых для компиляции данных, может быть обозначена специальным атрибутом. В остальных случаях компилятор может использовать анализ потока управления [7], чтобы определить возможность исполнения при компиляции конкретных участков кода. То есть ставится вопрос: требуются ли для исполнения данного кода данные от пользователя или его среды. Если нет, то в конечную программу попадает только результат его вычисления. На деле эта задача сводится к выделению в исходном коде комплекса метапрограмм, иерархически зависящих от выходных данных друг друга (рис. 5). При этом входные данные верхнего уровня прописаны в коде в качестве констант-аргументов для некоторых используемых библиотек и некоторые результаты работы среды разработчика, а выходные данные нижнего уровня напрямую попадают в конечную программу и взаимодействуют с системой пользователя без предварительных инициализаций. Такой подход также позволит очистить исполняемый файл от «мёртвого» кода и даже от «мёртвых» полей данных, которые существуют для поддержки обычно полезной функциональности, но не влияющей на работу данной конкретной программы.



Рисунок 4 – Общая схема работы будущего компилятора с возможностями исполнения кода

Модель состояний типов (typestate) [8] может обеспечить корректность применения последовательностей операций в каждой точке программы на основе описанных условий. Анализируя различные инструкции и условные переходы, компилятор может предсказать определённые наборы значений, которые переменные могут принять в отдельных ветвях программы, и предотвратить использование ошибочных значений в отдельных операциях либо неверной последовательности действий над некоторым объектом. В сочетании с контрактным программированием она даёт возможность предотвратить многие исключительные ситуации на этапе компиляции, поскольку создатели библиотек смогут установить особые ограничения на входные данные, и компилятор будет вынуждать программистов, использующих их библиотеки, выполнить соответствующие проверки или гарантировать корректность каким-либо другим образом (например, последовательностью преобразований).

Выводы

В ходе выполнения данной работы были изучены основные современные подходы к проектированию языков программирования и рассмотрена модель работы типичного компилятора. С учётом данных принципов был спроектирован C-подобный язык программирования общего назначения и компилятор для него.

Дальнейшее развитие проекта состоит в расширении системы типов и внедрении комплексных средств анализа кода для обеспечения улучшения качества процесса разработки ПО.

Литература

11. Исполнение кода и диагностика ошибок программ на этапе компиляции. Н.М. Ткачёв, А.И. Андрияхин, Компьютерная и программная инженерия - 2015 год: - Донецк: ДонНТУ, 2015. – С.105-107.

12. Interview With Bjarne Stroustrup [электронный ресурс]. – Режим доступа: <http://www.stroustrup.com/devXinterview.html>.

13. Rust Programming Language. Lifetimes [электронный ресурс]. – Режим доступа: <https://doc.rust-lang.org/book/lifetimes.html>.

14. В.Э. Карпов "Классическая теория компиляторов", ISBN 5–230–16344–5, 2011. - С. 7-10.

15. Rust Programming Language. Traits [электронный ресурс]. – Режим доступа: <https://doc.rust-lang.org/book/traits.html>.

16. Wikipedia. Compile-time function execution [электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Compile_time_function_execution.

17. Википедия. Анализ потока управления [электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Анализ_потока_управления

18. Wikipedia. Typestate analysis [электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Typestate_analysis.

Ткачёв Н.М., Андрияхин А.И. Проектирование и разработка компилятора C-подобного языка программирования общего назначения с поддержкой исполнения кода на этапе компиляции. В статье представлен краткий обзор современных принципов проектирования языков программирования и сформулированы требования к проектируемому языку. Спроектирована начальная итерация языка программирования общего назначения и кратко описана структура разрабатываемого компилятора. Перспективами развития языка является расширение системы типов и внедрение в компилятор комплексных средств анализа кода для предотвращения ошибок и исполнения кода на этапе компиляции.

Ключевые слова: компилятор, C, Flex, GNU Bison, генерация кода, вычисления на этапе компиляции, система типов.

Tkachev Nikolay, Andryukhin Alexander Designing a C-like general purpose programming language with CTFE capabilities and building a compiler. The article provides a brief overview of modern principles of programming language design and forms a set of requirements for a designed language. It showcases initial iteration of designed general purpose language and briefly describes architecture of its compiler. In the future language will have its type system expanded, the compiler will be extended with complex code analysis measures for error prevention and compile-time code execution.

Key words: compiler, C, Flex, GNU Bison, code generation, compile-time function execution, type system.

Анализ существующих подходов к разработке программ для анализа прогноза погоды

Сидоренко А.С., Коломойцева И.А.

Донецкий национальный технический университет
aleksersidorenko@gmail.com, bolatiger@gmail.com

Сидоренко А.С., Коломойцева И.А. Анализ существующих подходов к разработке программ для анализа прогноза погоды. В статье рассмотрены использованные средства для разработки приложения-анализатора прогноза погоды согласно данным из нескольких источников, выделены их преимущества и недостатки. Исследованы аналогичные программные продукты сторонних компаний на предмет удобства в использовании, указаны их проблемы совместимости с другими программами. Описана собственная программная система анализа достоверности информации на погодных сайтах.

Ключевые слова: прогноз погоды, Java, Jsoup, XChart, сериализация, css-запросы, база данных.

Введение

Прогноз погоды всегда был, есть и будет одним из важнейших данных, которым интересуется любой современный человек. Давным-давно его пытались определять по теням, поведению животных другим признакам. С появлением метеостанций, которые с большей точностью определяют грядущую погоду, узнать прогноз стало возможным из газет и ТВ. Сегодня, с появлением Интернета стало доступно множество сайтов, которые со всеми подробностями покажут, какой будет погода в любое время и в любом месте земного шара.

Большинство пользователей сервисов всемирной паутины используют для поиска прогноза только те порталы, о которых они узнали случайным образом (из рекламы, от знакомых и т.п.). Они обычно не знают о существовании других погодных сайтов, не имеют достаточных навыков в пользовании ПК, чтобы найти их.

На базе любой ОС существует множество клиентских программ, которые отображают текущую погоду в указанном месте, а также прогноз на ближайшие 3-4 дня. Такие программы легки в использовании, имеют интуитивно понятный интерфейс, в котором разберется даже малознакомый с компьютером человек. Одним из примеров является погодный гаджет для рабочего стола в Windows Seven. Он не занимает много места даже в развёрнутом виде и, в то же время, достаточно информативен. Но у него, как и всех остальных программ, есть существенный недостаток: программа предназначена для работы только на определенном сайте и невозможно его использовать для поиска прогноза на других.

Помочь пользователям получить более достоверный прогноз погоды путём анализа информации на разных сайтах и предназначена программная система, описанная в данной статье.

Целью данной работы является анализ существующих программ, показывающих прогноз погоды, с целью выявления их достоинств и недостатков, а также описание собственной программной системы для генерации графиков достоверности прогнозов погоды на основе эмпирических данных, полученных по информации, представленной на разных сайтах.

1 Анализ существующих систем

При проектировании программы были изучены и проанализированы программы, которые работают с прогнозом погоды. Опишем основные из них.

1. YoWindow (см. рис. 1) – это программа, целью которой является визуальное отображение погодных условий в вашем регионе. Подобных утилит на данный момент много, но отличие YoWindow от остальных состоит в том, что кроме текстовых показателей погодных условий, вы можете посмотреть погоду прямо на картинке, так, как будто пользователь сам смотрит в свое окно. Так, например, если в данный момент по прогнозу облачно, то на Flash изображении будут передвигаться облака, а если дует сильный ветер, то элементы изображения будут сильно колыхаться. Кроме того, солнце на картинке перемещается соответственно времени дня, а если открыть программу ночью, то и «за окном» ночной пейзаж [4].



Рисунок 1 – Интерфейс программы YoWindow

2. Weather Watcher (см. рис. 2) – программа, которая автоматически получает текущие условия, прогноз на час, на день, детальный прогноз, предупреждения о сложных погодных условиях, и погодные карты для более 77000 городов мира. При чем, предупреждения о сложных погодных условиях доступно только для городов США.

Удобству при работе с программой способствует то, текущие условия быстро просматриваются наведением курсора мышки на значок системы Weather Watcher в трее.

Weather Watcher позволяет автоматически получать данные о погоде в заданные интервалы, отображать текущие условия в значке в трее, выбирать информацию, отображаемую в окне Weather Watcher, отображать погодные условия различными вариантами [5].

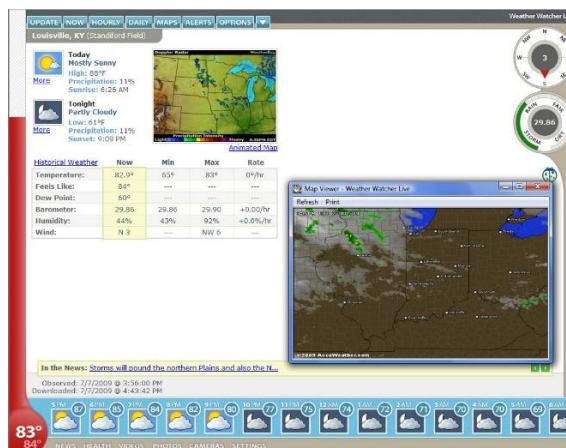


Рисунок 2 – Интерфейс программы Weather Watcher

3. PC Weather Machine (см. рис. 3) - интегрированная виртуальная метеостанция, которая предлагает больше метеорологической информации, чем большинство погодных модулей. Она позволяет держать пользователя в курсе информации о погоде, отображая метеорологические условия, карты погоды и городов для мест на выбор. Эта информация о погоде доступна по всему миру бесплатно [6].

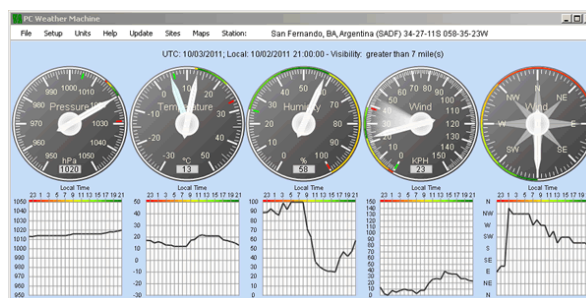


Рисунок 3 – Интерфейс программы PC Weather Machine

2 Проектирование программы для анализа прогноза погоды

В качестве языка программирования был выбран язык Java и интегрированная среда разработки NetBeans IDE.

После анализа существующих программ для просмотра прогноза погоды, была разработана диаграмма прецедентов для проектируемой программы (см. рис. 4). Диаграмма прецедентов описывает, с точки зрения действующего лица, группу действий в системе, которые приводят к конкретному результату. Варианты использования являются описаниями типичных взаимодействий между пользователем системы и самой системой. Они отображают внешний интерфейс системы и указывают форму того, что система должна сделать.

Диаграмма прецедентов (диаграмма вариантов использования) — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне [1].



Рисунок 4 – Диаграмма прецедентов

Исходя из спроектированной диаграммы прецедентов, была разработана архитектурная модель программы, которая показана на рисунке 5, а также схемы внешних и внутренних потоков, представленные на рисунках 6 и 7 соответственно.

Приложение будет состоять из 4 форм.

1. Главное окно – форма, в которой находится информация о текущей погоде в выбранном городе.
2. Окно прогноза погоды – форма с информацией о прогнозе погоды из нескольких источников на 3 дня.
3. Окно графиков – форма, в которой содержится поле для вывода графиков достоверности и выпадающий список для выбора типа выводимого графика.

4. Окно настроек – форма для изменения настроек и имеющая 2 вкладки. Первая – Города, в ней можно изменить текущий город или добавить новый, введя его названия и коды из сайта Gismeteo. Вторая – вкладка сетевых настроек, которая позволяет использовать или не использовать подключение через прокси-сервер.

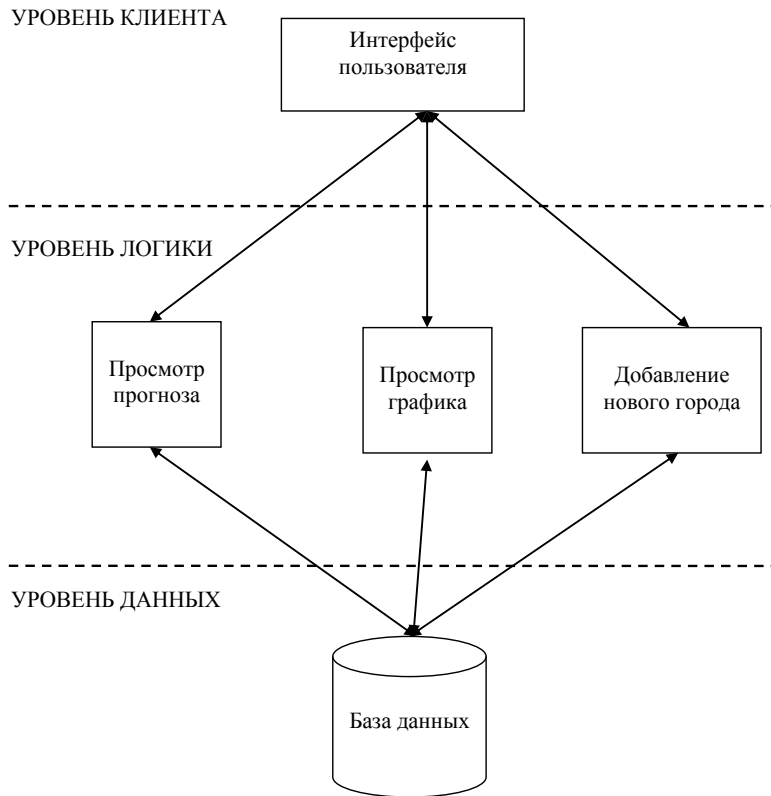


Рисунок 5 – Архитектурная модель программы

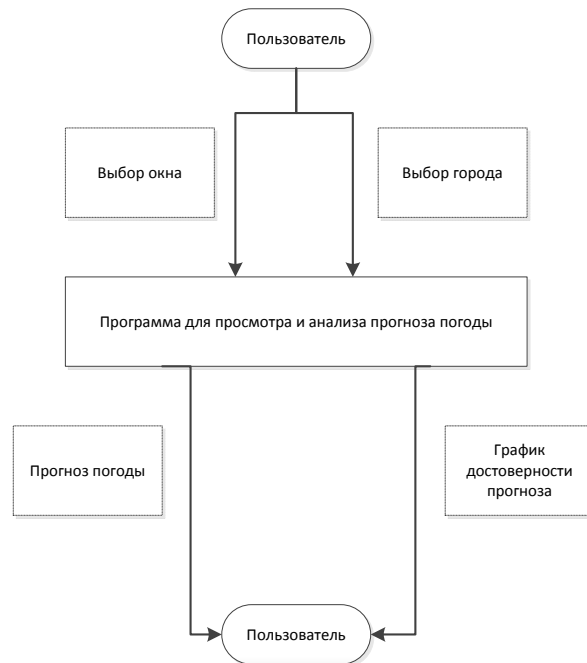


Рисунок 6 – Схема внешних потоков данных

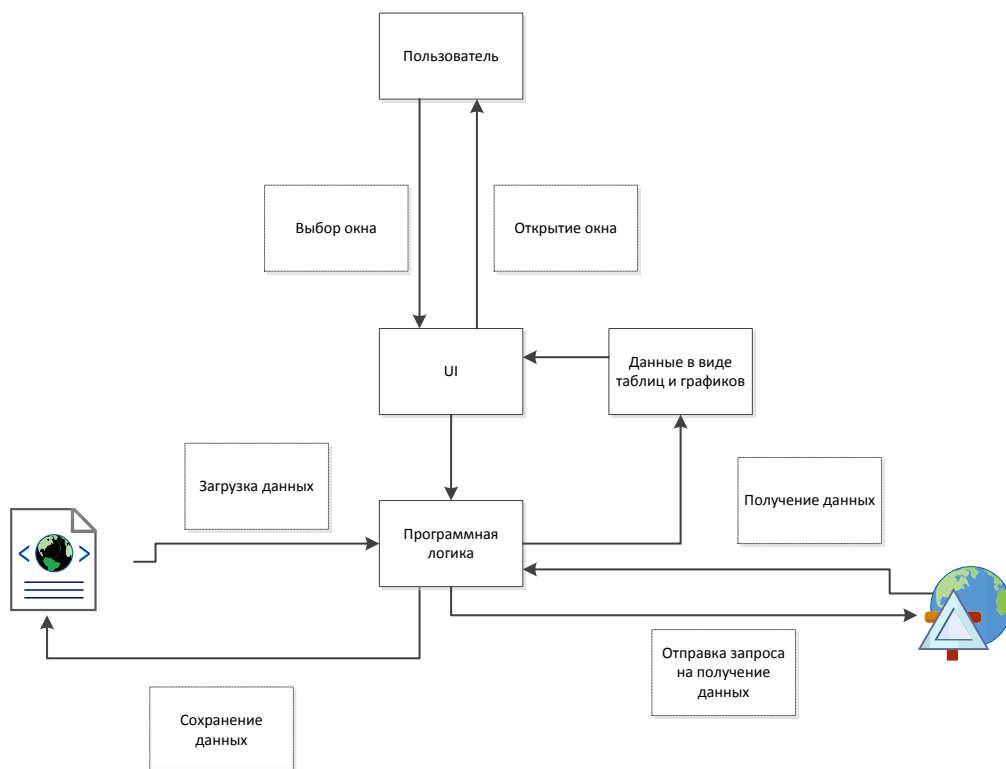


Рисунок 7 – Схема внутренних потоков данных

3 Основные использованные библиотеки

1. Jsoup – это библиотека Java для работы с реальной HTML-страницей. Библиотека обеспечивает очень удобный API для извлечения и обработки данных, используя лучшие методы из DOM, CSS, JQuery и других библиотек, которые чаще всего используются при разработке веб-приложений [2].

2. Помимо вывода данных в текстовом виде, в программе будет предусмотрена функция вывода графиков, которые показывают погрешность в прогнозе по двум критериям: температура и осадки. За вывод информации в виде графиков отвечает библиотека XChart. Она является небольшой по размерам и удобной для построения графиков, имея некоторый ряд данных. Библиотека позволяет отобразить график в программе или же сохранить его отдельным файлом [3].

4 Алгоритм запуска программы

Диаграмму действий программы во время запуска приложения можно рассмотреть на рисунке 8.

Диаграмма действий (англ. activity diagram) — UML-диаграмма, на которой показано разложение некоторой деятельности на её составные части [4].

Диаграмма последовательностей – это удобный способ для обозначения очередности прохождения одним за другим разных стимулов (сообщений), с помощью которых объекты взаимодействуют между собой.

Главный акцент – порядок и динамика поведения, то есть как и в каком порядке происходят события.

Отличие диаграмм классов в том, что диаграмма классов дает статическую картинку, то есть описание, которое не меняется во время выполнения программы.

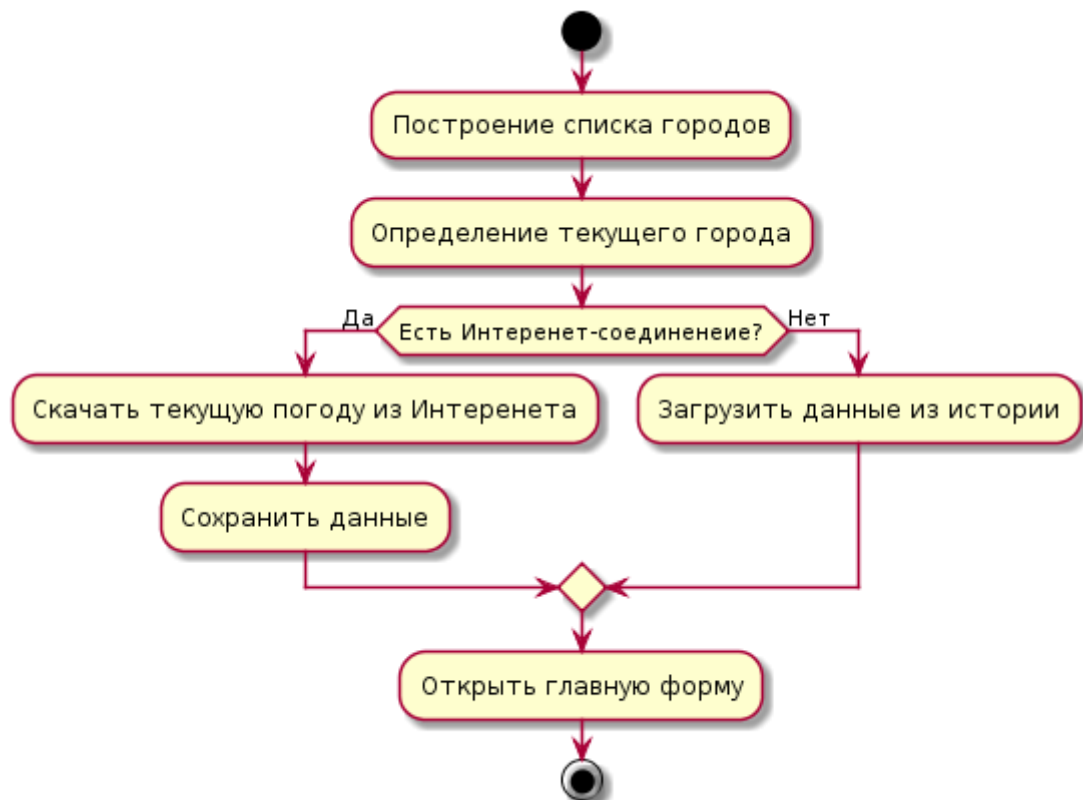


Рисунок 8 – Диаграмма действий

5 Проектирование структур и баз данных

Данная программа требует большого объема данных, поэтому роль базы данных играют файлы в формате .xml, хранящиеся в директории /data/. XML (Extensible Markup Language) - это новый SGML-производный язык разметки документов, позволяющий структурировать информацию разного типа, используя для этого произвольный набор инструкций [5]. Эти файлы можно поделить на 3 категории:

Эти файлы можно поделить на 3 категории:

- файлы прогноза погоды;
- файлы истории погоды;
- файлы настроек программы;
- файлы графиков, которые в последствии выводятся на экран.

Пример файла, содержащего информацию о прогнозе погоды с сайта Gismeteo.ua, представлен на рисунке 9. Данный файл будет использоваться для построения графиков достоверности, пример которого представлен на рисунке 10. Он, как и все остальные файлы данного рода, хранятся в соответствующих директориях, которые соответствуют названию источника, году и месяцу прогноза. Например, полный путь к файлу с информацией о прогнозе погоды в Донецке с сайта Gismeteo за 6 июня 2016 года будет выглядеть так: \data\donetsk\gismeteoForecast\2016\6\6.6.2016.xml. Файлы графиков имеют разрешение bmp и имеют название того, какой тип графика в них находится. Расположены в папке /plots/.

Файлы с настройками содержат в себе данные о городах и настройках прокси-сервера, их структуры расположены на рисунках 11 и 12 соответственно.

Диаграмма классов – диаграмма, которая демонстрирует классы системы, их атрибуты, методы и взаимосвязи между ними. Диаграмма классов представлена на рисунке 13.

```

1 <jweatheranalyzer.WeatherDayGismeteo>
2   <weather>
3     <jweatheranalyzer.WeatherGismeteo>
4       <IMG class="java.awt.image.BufferedImage">
340     <PART_OF_DAY>Ночь</PART_OF_DAY>
341     <TEMP>+13</TEMP>
342     <PRESSURE>740</PRESSURE>
343     <WIND_SPEED>1</WIND_SPEED>
344     <WIND_DIRECT>ЮЗ</WIND_DIRECT>
345     <HUMIDITY>82</HUMIDITY>
346     <DOWNFALL>Ясно</DOWNFALL>
347   </jweatheranalyzer.WeatherGismeteo>
348   <jweatheranalyzer.WeatherGismeteo>
349     <IMG class="java.awt.image.BufferedImage">
790     <PART_OF_DAY>Утро</PART_OF_DAY>
791     <TEMP>+17</TEMP>
792     <PRESSURE>740</PRESSURE>
793     <WIND_SPEED>2</WIND_SPEED>
794     <WIND_DIRECT>Ю</WIND_DIRECT>
795     <HUMIDITY>68</HUMIDITY>
796     <DOWNFALL>Ясно</DOWNFALL>
797   </jweatheranalyzer.WeatherGismeteo>
798   <jweatheranalyzer.WeatherGismeteo>
799     <IMG class="java.awt.image.BufferedImage">
1085     <PART_OF_DAY>День</PART_OF_DAY>
1086     <TEMP>+19</TEMP>
1087     <PRESSURE>739</PRESSURE>
1088     <WIND_SPEED>4</WIND_SPEED>
1089     <WIND_DIRECT>Ю</WIND_DIRECT>
1090     <HUMIDITY>57</HUMIDITY>
1091     <DOWNFALL>Малооблачно</DOWNFALL>
1092   </jweatheranalyzer.WeatherGismeteo>
1093   <jweatheranalyzer.WeatherGismeteo>
1094     <IMG class="java.awt.image.BufferedImage">

```

eXtensible Markup Language file

Рисунок 9– Файл с данными о прогнозе

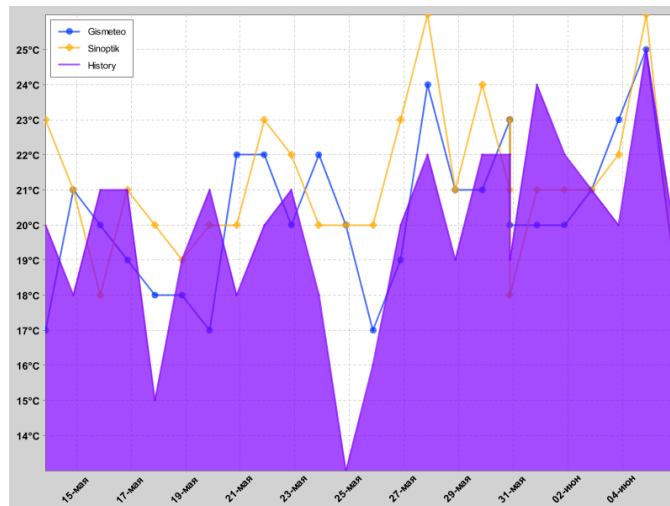


Рисунок 10 – Файл с графиком

```

1 <list>
2   <jweatheranalyzer.City>
3     <GISMETEO_CITYNAME>donetsk</GISMETEO_CITYNAME>
4     <CITYCODE>5080</CITYCODE>
5     <SINOPTIKUA_CITYNAME>Донецк</SINOPTIKUA_CITYNAME>
6     <chosen>true</chosen>
7   </jweatheranalyzer.City>
8   <jweatheranalyzer.City>
9     <GISMETEO_CITYNAME>Kyiv</GISMETEO_CITYNAME>
10    <CITYCODE>4944</CITYCODE>
11    <SINOPTIKUA_CITYNAME>Київ</SINOPTIKUA_CITYNAME>
12    <chosen>false</chosen>
13  </jweatheranalyzer.City>
14  <jweatheranalyzer.City>
15    <GISMETEO_CITYNAME>Moscow</GISMETEO_CITYNAME>
16    <CITYCODE>4368</CITYCODE>
17    <SINOPTIKUA_CITYNAME>Москва</SINOPTIKUA_CITYNAME>
18    <chosen>false</chosen>
19  </jweatheranalyzer.City>
20  <jweatheranalyzer.City>
21    <GISMETEO_CITYNAME>Luhansk</GISMETEO_CITYNAME>
22    <CITYCODE>5082</CITYCODE>
23    <SINOPTIKUA_CITYNAME>Луганск</SINOPTIKUA_CITYNAME>
24    <chosen>false</chosen>
25  </jweatheranalyzer.City>
26 </list>

```

Рисунок 11 – Структура файла с городами

```

<Connect.ConnectSettings>
  <proxyEnabled>false</proxyEnabled>
  <proxyIP>192.168.1.1</proxyIP>
  <proxyPort>552</proxyPort>
</Connect.ConnectSettings>

```

Рисунок 12 – Структура файла с настройками прокси-сервера

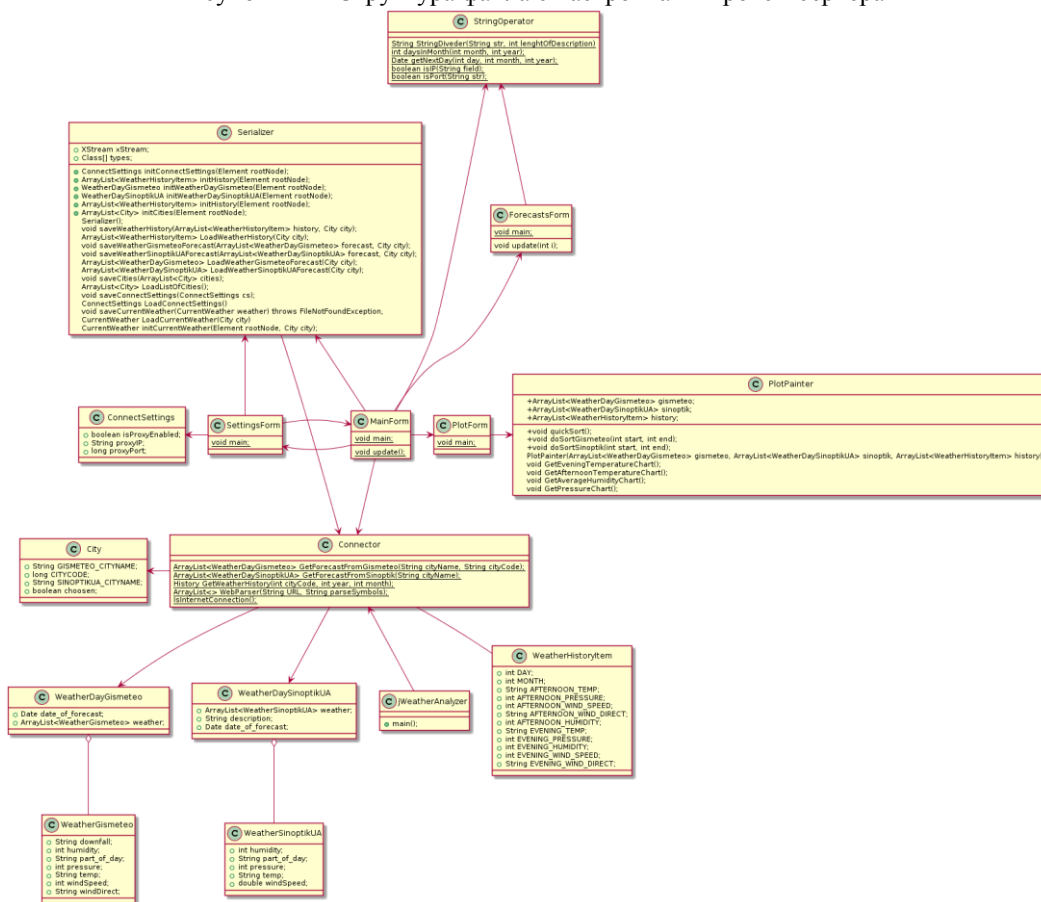


Рисунок 13 – Диаграмма классов

6 Результат разработки программы-анализатора прогноза погоды

После разработки алгоритмов, диаграмм классов была разработана программа на объектно-ориентированном языке Java, а также над ней было проведено модульное тестирование.

На рис. 14-18 представлены экранные формы программы.

Выводы

Выполненный проект представляет собой программную систему оценки достоверности прогноза погоды по информации на сайтах для ПК. Программная система была разработана на языке Java с использованием интегрированной среды разработки NetBeans IDE 8.1.

При написании программной системы были получены следующие результаты:

- выполнен системный анализ похожих программ, выявлены их достоинства и недостатки, в результате чего были сформулированы требования к программе в рамках данного дипломного проекта;
- сформулирован широкий набор требований, которые предъявляются программе, в частности, сформулированы требования к структуре БД, интерфейса пользователя, системные и программные требования;
- разработана структура БД, которая состоит из файлов настроек, городов, прогнозов по годам по каждому городу и дате, фактических данных о погоде;
- построена объектная модель программной среды;
- разработаны алгоритмы, по которым функционирует программа;
- разработан интерфейс пользователя;

- разработан инсталляционный пакет для программы и описан процесс установки программы;
- разработаны тесты и проведено тестирование системы, которое показало, что программный комплекс работает корректно.

Данная программа в перспективе предполагает улучшение и увеличение функционала. К таким улучшениям можно отнести работу с данными через серверы БД. Так как после установки программы пользователь получает просто программную оболочку и на сбор данных требуется время, то имеет смысл говорить о получении данных с удаленных серверов, которые будут отвечать за сбор данных с самих сайтов, а не предоставлять эту функцию пользователю. Это также решит вопрос с производительностью, потому что асинхронные CSS-запросы, которых при получении данных осуществляется большое количество, занимают время приблизительно в 10 секунд. Поэтому эти решения будут проанализированы и лучшие из них будут применены для оптимизации данного продукта

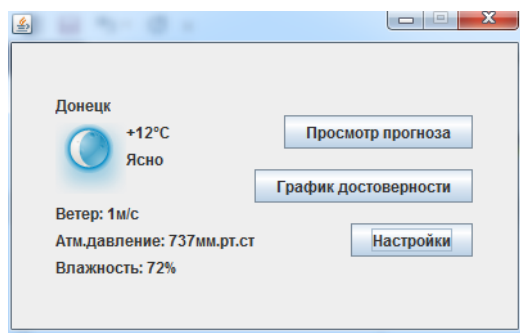


Рисунок 14 – Главное окно программы



Рисунок 15 – Окно просмотра прогноза погоды

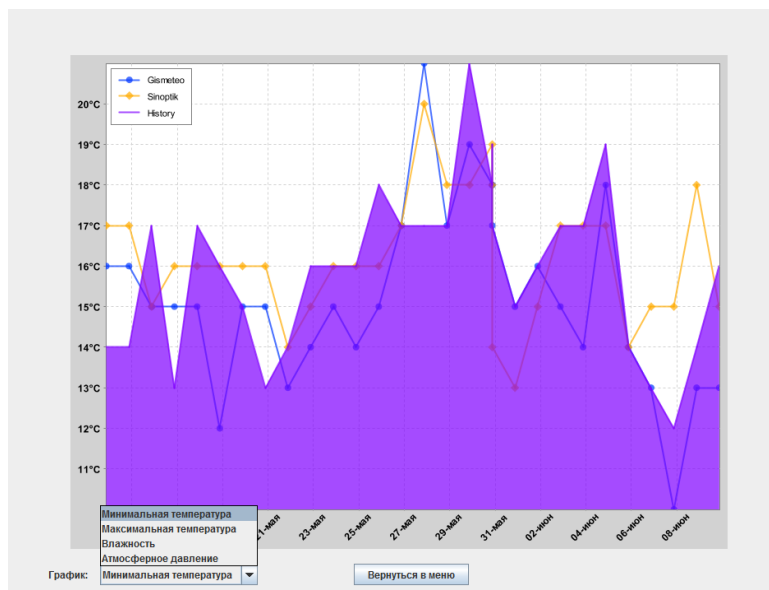


Рисунок 16 - Окно вывода графика с раскрытым выпадающим списком

Рисунок 17 – Окно «Настройки»

Рисунок 18 – Вкладка «Соединение» с включенными настройками прокси-сервера

Литература

1. Диаграмма прецедентов — Википедия [электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Диаграмма_прецедентов

2. Jsoup [электронный ресурс]. – Режим доступа: <https://jsoup.org/apidocs/>
3. Knowm.org | XChart [электронный ресурс]. – Режим доступа: <http://knowm.org/open-source/xchart/>
4. Диаграмма деятельности — Википедия [электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Диаграмма_деятельности
5. Язык XML - Описание технологии [электронный ресурс]. – Режим доступа: <http://www.codenet.ru/webmast/xml/part2.php>

Сидоренко А.С., Коломойцева И.А. Анализ существующих подходов к разработке программ для анализа прогноза погоды. В статье рассмотрены использованные средства для разработки приложения-анализатора прогноза погоды согласно данным из нескольких источников, выделены их преимущества и недостатки. Исследованы аналогичные программные продукты сторонних компаний на предмет удобства в использовании, указаны их проблемы совместимости с другими программами. Описана собственная программная система анализа достоверности информации на погодных сайтах.

Ключевые слова: прогноз погоды, Java, Jsoup, XChart, сериализация, css-запросы, база данных.

Sidorenko Aleksander, Kolomoitseva Irina. Analysis of existing approaches to the development of programs for the analysis of weather forecasting. The article describes the means used to develop weather forecasting application analyzer according to data from multiple sources are highlighted their advantages and disadvantages. The article describes the study a similar third-party software product for ease of use, given their problems with other software compatibility. Described own software system for the analysis of the reliability of information on weather sites.

Key words: weather forecast, Jsoup, XChart, serialization, css-queries, database.

Языки и средства web-программирования

В.Ю. Демчук
Министерство Связи ДНР
demchuk.vyu@minsvyazdnr.ru

Демчук В.Ю. Языки и средства web-программирования. Освещены популярные языки web-программирования. Описаны основные средства в виде шаблонов проектирования, шаблонизаторов, frontend и backend фреймворки, а также средств по разработке приложений IDE.

Ключевые слова: web, язык программирования, фреймворк, frontend, backend, IDE.

Введение

Современный набор технологий для разработки web приложений велик. Основной проблемой при старте нового проекта является выбор конкретных технологий на которых он будет реализован. Так как конкретную задачу можно решить практически на любом из современных языков и средств web - программирования, требуется определить основные критерии выбора стека технологий. Критериями выбора стека технологий являются: сроки реализации проекта, компетенция команды, бюджет на разработку, модульность и расширяемость, масштаб проекта, сложность поддержки проекта, взаимодействие с другими компонентами системы.

Данная проблема решалась специалистами IBM [1]. Автор статьи выделяет ряд нефункциональных требований, которые служат факторами определяющими вид технологии такие как: слабая или жесткая связанность, переносимость и функциональная совместимость, поддержка транзакций, надежность, безопасность. В ходе описания каждого критерия автор поясняет его реализацию на примере сервис ориентированной архитектуры приложений. В конце статьи автор резюмирует, что выбор конкретной технологии осуществляется по нескольким критериям, а выбор критериев определяется требованиями конкретных задач и приоритетами для этих задач.

Также данная проблема обсуждалась специалистами из Yandex [2]. В ходе статьи автор рассказывает историю создания языков программирования, о выборе языка в зависимости от реализации конкретной задачи.

Цель статьи — осветить популярные языки и средства web - программирования и обосновать выбор конкретной технологии при реализации поставленных задач.

1. Языки web - программирования.

Языки web - программирования можно разделить на 2 большие группы: «frontend» и «backend». Также эти группы звучат как «клиентские» и «серверные».

В свою очередь стек «frontend» технологий делится на:

- HTML
- CSS
- Javascript

Стек «backend» технологий делится на:

- Javascript
- Java
- Python
- PHP
- Ruby
- ...

1.1 Frontend

Обязательными «frontend» языками программирования является HTML и CSS. Все шаблонизаторы HTML страниц и CSS препроцессоры сводят все к простому HTML и CSS. Javascript предназначен для

взаимодействия пользователя с браузером, сервером, повышения уровня положительного опыта юзера при использовании web - приложений, и не только. Поговорим подробнее о HTML, CSS и Javascript.

1.1.1 HTML

HTML 5 — последняя версия языка гипертекстовой разметки. Разработкой спецификаций языка занимается W3C — консорциум всемирной паутины [3]. HTML 5 формирует семантику, новые способы общения с сервером, методы позволяющие сохранять информацию на стороне клиента, взаимодействие с видео и звуком, 2D/3D графика, использование различных устройств для ввода и вывода информации [4]. Рассмотрим наиболее интересные особенности HTML 5.

Семантика — описание контента путем определенных тегов.

- Использует секции и контуры.
- MathML — позволяет вставлять математические формулы.

Связь — способы общения с сервером.

– Web Sockets — позволяет создать постоянное соединение между страницей и сервером и обмениваться данными через него.

– Server-sent события — позволяет серверу отправлять события клиенту, а не по классической парадигме, где сервер может передавать данные только в ответ на запрос клиента.

– WebRTC - эта технология, где RTC создает возможность общения в реальном времени, позволяет подключаться к другим людям и контролировать видеоконференции непосредственно в браузере, без необходимости плагинов и внешний приложений.

– Local Storage — хранилище данных на стороне клиента.

– Online and offline events — это события, которые позволяют приложениям и расширениям обнаружить есть ли активное подключение к Интернет, а также определить, когда соединение портится или улучшается.

– DOM Storage — постоянное или сессионное хранилище позволяет web - приложениям хранить структурированные данные на стороне клиента.

– IndexedDB — web - стандарт для хранения значительных количеств структурированных данных в браузере и для быстрого их поиска, используя индексы.

– Мультимедиа — взаимодействие с видео и аудио.

– <video>, <audio> — тэги, которые позволяют взаимодействовать с видео и аудио.

– WebRTC — технология, где RTC создает возможность общения в реальном времени, позволяет подключаться к другим людям и контролировать видеоконференции непосредственно в браузере, без необходимости плагинов и внешний приложений.

– Camera API — позволяет контролировать, манипулировать и хранить изображения с камеры устройства.

– Графика и эффекты — способы представления.

– WebGL — приносит 3D в web, соответствует OpenGL ES 2.0, может использоваться в HTML5 через <canvas>.

– SVG — основанный на XML формат векторных изображений, который может быть непосредственно вставлен в HTML.

Производительность и интеграция.

Web Workers — позволяет делегировать выполнение Javascript в фоновые потоки.

XMLHttpRequest — позволяет извлечь асинхронно некоторые части страницы, что позволяет отобразить динамический контент, изменяющейся время от времени или от действий пользователя. Это технология, лежащая в основе Ajax.

History API — позволяет управлять историей браузера.

Drag and drop — позволяет перетаскивать элементы по сайту или на него.

Fullscreen API — позволяет использовать весь экран для web - приложения, без отображения UI браузера.

Доступ к устройствам — использование разных устройств для ввода и вывода информации.

– Touch events — обрабатывает события, создаваемые нажатиями пользователя по тач скрину.

– Геолокация — позволяет браузерам получать местоположение пользователя.

Определение ориентации устройства — позволяет среагировать, когда устройство, на котором работает браузер, меняет ориентацию.

Стилизация — оформление web — страниц.

Улучшенная типографика — загрузка не стандартных шрифтов и применение разных видов переноса текста.

Улучшенные границы — введение свойства «border-radius» для скругления углов.

1.1.2 CSS

CSS (Cascading Style Sheets) — это язык иерархических правил (таблиц стилей), используемый для представления внешнего вида документа, написанного на HTML или XML (включая различные языки XML, такие как SVG и XHTML) [5]. CSS описывает, каким образом элемент должен отображаться на экране, на бумаге, голосом или с использованием других медиа средств.

CSS является одним из основных языков свободной web - разработки, который стандартизован спецификацией W3C [6].

CSS применяется для оформления web - страниц, адаптации контента под различные устройства, представления страницы при печати, используется для создания анимаций компонентов HTML.

1.1.3 Javascript

Javascript — это кросс-платформенный, объектно-ориентированный скриптовый язык, являющийся небольшим и легковесным. Внутри среды исполнения Javascript может быть связан с объектами данной среды и предоставлять программный контроль над ними [7].

Современный Javascript – это язык программирования общего назначения. Он не предоставляет низкоуровневых средств работы с памятью, процессором, так как изначально был ориентирован на браузеры, в которых это не требуется [8].

Что же касается остальных возможностей – они зависят от окружения, в котором запущен Javascript. В браузере Javascript умеет делать всё, что относится к манипуляции со страницей, взаимодействию с посетителем и, в какой-то мере, с сервером: [8]

Создавать новые HTML-теги, удалять существующие, менять стили элементов, прятать, показывать элементы и т. п.

Реагировать на действия посетителя, обрабатывать клики мыши, перемещения курсора, нажатия на клавиатуру и т. п.

Посылать запросы на сервер и загружать данные без перезагрузки страницы (AJAX).

Получать и устанавливать cookie, запрашивать данные, выводить сообщения.

Javascript стандартизован ассоциацией Ecma International, деятельность которой посвящена стандартизации информационных и коммуникационных систем (изначально ECMA являлся акронимом European Computer Manufacturers Association). Эта стандартизованная версия Javascript, называемая ECMAScript, работает одинаково во всех приложениях, поддерживающих стандарт. Компании могут использовать открытый стандарт языка, чтобы разработать собственную реализацию Javascript. Стандарт ECMAScript задокументирован в спецификации ECMA-262 [9].

1.2 Backend

Стек backend языков программирования довольно обширный. Для анализа были выбраны 5 наиболее популярных из них. Для используется инструмент Github [10], который анализирует больше 2 миллионов активных репозиторий на github. Топ 5 web-backend языков программирования по версии Github на 31.10.2016:

- Javascript
- Java
- Python
- PHP
- Ruby

1.2.1 Javascript

Как было описано выше в пункте 1.1.3, Javascript используется как frontend инструмент. Однако данный язык также используют и на backend — на стороне сервера.

Серверный Javascript расширяет ядро языка, предоставляя объекты, относящиеся к запуску Javascript на сервере. Например, серверные расширения дают приложению возможность соединиться с реляционной БД, сохранять информацию между вызовами приложения или выполнять работу с файлами на сервере [11].

В отличие от страниц с чисто клиентским Javascript, HTML-страницы, использующие серверный Javascript, компилируются в байт-кодовые исполняемые файлы. Эти исполняемые приложения запускаются на выполнение web-сервером, имеющим машину времени выполнения Javascript. Исходя из этого, создание приложений Javascript это процесс из двух этапов.

1.2.2 Java

Java — объектно-ориентированный язык программирования. Изначально язык назывался Oak (James

Gosling) и разрабатывался для бытовой электроники, но впоследствии был переименован в Java и стал использоваться для написания апплетов, приложений и серверного программного обеспечения. Также Java применяется для разработки web-приложений [12].

Программы на Java могут быть транслированы в байт-код, выполняемый на виртуальной java-машине (JVM) — программе, обрабатывающей байт-код и передающей инструкции оборудованию, как интерпретатор, но с тем отличием, что байт-код, в отличие от текста, обрабатывается значительно быстрее.

Язык Java потребовался для создания интерактивных продуктов для сети Internet. Три ключевых элемента объединились в технологии языка Java:

Предоставляет для широкого использования свои апплеты (applets) — небольшие, надежные, динамические, не зависящие от платформы активные сетевые приложения, встраиваемые в страницы Web.

Высвобождает мощь объектно-ориентированной разработки приложений, сочетая простой и знакомый синтаксис с надежной и удобной в работе средой разработки.

Предоставляет программисту богатый набор классов объектов для ясного абстрагирования многих системных функций, используемых при работе с окнами, сетью и для ввода-вывода.

1.2.3 Python

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объем полезных функций. [13]

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).

1.2.4 PHP

PHP — интерпретируемый язык общего назначения, интенсивно применяемый для разработки веб - приложений. В настоящее время поддерживается подавляющим большинством хостинг - провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб - сайтов [14].

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией, несовместимой с GNU GPL.

PHP поддерживает динамическую типизацию данных, автоматическое управление памятью, предоставляет механизм по обработке исключений, поддерживает функциональное и объектно-ориентированное программирование.

1.2.5 Ruby

Ruby — динамический, интерпретируемый высокоуровневый язык программирования. Язык обладает независимой от операционной системы реализацией многопоточности, строгой динамической типизацией, сборщиком мусора, позволяет обрабатывать исключения, поддерживает объектно - ориентированный подход [15].

Ruby используется в качестве языка конфигураций для COMPASS (расширение SASS), основного языка фреймворка Ruby on Rails и др.

2. Средства web - программирования.

Среди средств web - программирования можно выделить шаблоны проектирования, шаблонизаторы, frontend и backend фреймворки, системы управления контентом (CMS), инструменты для web разработки.

2.1 Шаблоны проектирования.

Шаблон проектирования или паттерн (англ. design pattern) в разработке программного обеспечения — повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста [16].

2.1.1 MVC

Model-view-controller (MVC, «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким

образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области [17].

2.1.2 MVP

Model-View-Presenter (MVP) — шаблон проектирования, производный от MVC, который используется в основном для построения пользовательского интерфейса. [18]

MVP — шаблон проектирования пользовательского интерфейса, который был разработан для облегчения автоматического модульного тестирования и улучшения разделения ответственности в презентационной логике (отделения логики от отображения):

Модель (англ. Model) — предоставляет данные для пользовательского интерфейса.

Представление (англ. View) — реализует отображение данных (Модели) и маршрутизацию пользовательских команд или событий Presenter'у.

Presenter — управляет Моделью и Представлением. Например, извлекает данные из Модели и форматирует их для отображения в Представлении.

2.1.3 MVVM

Model-View-ViewModel (MVVM) — применяется при проектировании архитектуры приложения. MVVM используется для разделения модели и её представления, что необходимо для изменения их отдельно друг от друга. Например, разработчик задает логику работы с данными, а дизайнер соответственно работает с пользовательским интерфейсом [19].

В шаблонах проектирования MVC/MVP изменения в пользовательском интерфейсе не влияют непосредственно на Модель, а предварительно идут через Контроллер (англ. Controller) или Presenter. В таких технологиях как WPF и Silverlight есть концепция «связывания данных», позволяющая связывать данные с визуальными элементами в обе стороны. Следовательно, при использовании этого приема применение модели MVC становится крайне неудобным из-за того, что привязка данных к представлению напрямую не укладывается в концепцию MVC/MVP.

2.2 Шаблонизаторы.

Шаблонизатор — программное обеспечение, позволяющее использовать html-шаблоны для генерации конечных html-страниц. Основная цель использования шаблонизаторов — это отделение представления данных от исполняемого кода. Часто это необходимо для обеспечения возможности параллельной работы программиста и дизайнера-верстальщика. Использование шаблонизаторов часто улучшает читаемость кода и внесение изменений во внешний вид, когда проект целиком выполняет один человек [20].

Среди популярных шаблонизаторов:

- Javascript: Jade, EJS, TwigJs.
- Java: Thymeleaf, Apache Velocity, Mustache.
- Python: Jinja, Genshi, Mako.
- PHP: Blade, Twig, Smarty.
- Ruby: Haml, Slim, Erb.

2.3 Frontend фреймворки.

Frontend фреймворки можно разделить на 2 большие группы: CSS и Javascript библиотеки и фреймворки.

2.3.1 CSS фреймворки.

CSS-фреймворк — инструмент, созданный для упрощения работы верстальщика, скорости разработки и исключения максимально возможного числа ошибок верстки (проблемы совместимости различных версий браузеров и т. д.). Как и библиотеки скриптовых языков программирования, CSS-библиотеки, обычно имеющие вид внешнего css-файла, «подключаются» к проекту (добавляются в заголовок веб-страницы) [21].

Среди наиболее популярных CSS фреймворков выделяют: Foundation, Bootstrap, Material Design Lite.

Некоторые из CSS фреймворков написаны с использованием препроцессоров — скриптовые метаязыки, которые компилируются в обычные CSS-стили. Среди наиболее популярных препроцессоров выделяют: SASS, SCSS, LESS, PostCSS. Также при разработке фреймворков или собственных тем оформления используется методология SMACSS — масштабируемая и модульная архитектура для CSS.

2.3.2 Javascript библиотеки.

Javascript библиотеки — инструмент для облегчения процесса создания не больших модулей. Библиотеки предоставляют удобные обертки для работы с нативным Javascript. Среди наиболее популярных библиотек выделяют: ReactJS, jQuery, Polymer.

2.4 Backend фреймворки.

Backend фреймворки представлены большим количеством в современном web стеке. Каждый из фреймворков реализован на определенном шаблоне проектирования. Выделим несколько backend фреймворков в различных языках программирования:

- Javascript: NodeJS, AngularJS, BackboneJS, Ember, KnockoutJS.
- Java: Spring, Java Server Faces, Struts 2.
- Python: Django, TurboGears, Pyramid.
- PHP: Symfony, Yii2, Laravel, Phalcon, Zend Framework.
- Ruby: Ruby on Rails, Rack, Sinatra.

Особенно хотелось выделить NodeJS, AngularJS, Django, Laravel.

NodeJS — программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом. [22]

AngularJS — Javascript-фреймворк с открытым исходным кодом. Предназначен для разработки одностраничных приложений. Его цель — расширение браузерных приложений на основе MVC шаблона, а также упрощение тестирования и разработки. [23]

Django — свободный программный каркас для веб-приложений на языке Python, использующий шаблон проектирования MVC. Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого каркаса от некоторых других (например, Ruby on Rails). Один из основных принципов каркаса — DRY (англ. Don't repeat yourself) [24]

Laravel — бесплатный веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC. [25].

2.5 CMS.

CMS — система управления контентом. CMS ускоряет разработку сайта, упрощают его администрирование, предоставляют собственные API и стандарты по написанию кода, предоставляют готовые методы реализации приложения на основе шаблонов проектирования. Структура CMS — это ядро, модули и темы.

Наиболее популярные CMS системы написаны на языке программирования PHP. Среди них выделяют такие CMS как:

Drupal 8 — мощная CMS на основе фреймворка Symfony2. Drupal сообщество предоставляет большое количество модулей и тем распространяемых под свободной лицензией GPL как и свое ядро [26]. За счет большого количества универсальных модулей, время разработки веб сайта значительно уменьшается. Данная CMS применяется для создания магазинов, блогов, новостных порталов, социальных сетей, форумов и т.д.

Wordpress — система управления содержимым сайта с открытым исходным кодом. Выпущена под лицензией GNU GPL версии 2. Сфера применения — от блогов до достаточно сложных новостных ресурсов и интернет-магазинов. Встроенная система «тем» и «плагинов» вместе с удачной архитектурой позволяет конструировать проекты широкой функциональной сложности [27].

Joomla — CMS с открытым исходным кодом. Преимущественно используется для создания интернет магазинов.

2.6 Инструменты для веб разработки.

Инструменты для web можно разделить на две категории: которые распространяются под свободной лицензией и коммерческой.

Среди свободного ПО наиболее популярные:

- Текстовые редакторы: Sublime Text, Atom, Brackets.
- IDE: NetBeans, Eclipse.

Среди коммерческого ПО наиболее популярные.

IDE: WebStorm, PhpStorm, RubyMine, Pcharm, IntelliJ IDEA, Visual Studio.

Выводы

В ходе статьи были освещены популярные языки и средства для web-программирования. Как мы видим, современный web насчитывает большое количество frontend и backend фреймворков. За счет четкой организации кода внутри фреймворков и множеству решенных задач, они позволяют сократить время на разработку и сделать код более структурированным. Рекомендуется использовать фреймворки в зависимости от конкретной задачи. Современному web-разработчику требуется в общих чертах знать и понимать какие из функций выполняет frontend или backend фреймворк. За счет большого кругозора в сфере web-фреймворков можно грамотно определить стек технологий для решения конкретных задач.

Литература

1. <http://www.ibm.com/developerworks/ru/library/ws-jcajms/>
2. <https://habrahabr.ru/company/yandex/blog/272759/>
3. <https://www.w3.org/>
4. <https://developer.mozilla.org/ru/docs/HTML/HTML5>
5. <https://developer.mozilla.org/ru/docs/Web/CSS>
6. <https://www.w3.org/Style/CSS/#specs>
7. https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide/Введение_в_JavaScript
8. <https://learn.Javascript.ru/intro>
9. <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
10. <http://github.info/>
11. <http://spec.piramidin.com/servguide14/index.htm>
12. <http://progopedia.ru/language/java/>
13. <https://ru.wikipedia.org/wiki/Python>
14. <https://ru.wikipedia.org/wiki/PHP>
15. <https://ru.wikipedia.org/wiki/Ruby>
16. https://ru.wikipedia.org/wiki/Шаблон_проектирования
17. <https://ru.wikipedia.org/wiki/Model-View-Controller>
18. <https://ru.wikipedia.org/wiki/Model-View-Presenter>
19. <https://ru.wikipedia.org/wiki/Model-View-ViewModel>
20. <https://ru.wikipedia.org/wiki/Шаблонизатор>
21. <https://ru.wikipedia.org/wiki/CSS-фреймворк>
22. <https://ru.wikipedia.org/wiki/Node.js>
23. <https://ru.wikipedia.org/wiki/AngularJS>
24. <https://ru.wikipedia.org/wiki/Django>
25. <https://ru.wikipedia.org/wiki/Laravel>
26. <https://www.drupal.org/about/licensing>
27. <https://ru.wikipedia.org/wiki/WordPress>
28. <https://ru.wikipedia.org/wiki/Joomla>

Demchuk V.Yu. Languages and web-based software. This article describes the popular web-programming languages. This article describes the main tools in the form of design patterns, templating, frontend and backend frameworks and IDE tools to develop applications.

Ключевые слова: web, programming languages, framework, frontend, backend, IDE.

**Программная инженерия:
методы и технологии разработки информационно-
вычислительных систем
(ПИИВС-2016)**

**Сборник научных трудов
I Международной научно-практической конференции**

**г. Донецк
16-17 ноября 2016 года**