

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 23

2017

№ 6

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Ермаков А. Е.** Язык семантических трансформаций для компьютерной интерпретации текста 403
- Левин В. И.** Максимальные и минимальные полиинтервалы и оптимизация в условиях неопределенности 413
- Романов А. Ю., Ломотин К. Е., Козлова Е. С.** Применение методов машинного обучения для решения задачи автоматической рубрикации статей по УДК 418

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Акимкина Э. Э.** Инструментальный подход к организации сбора данных в хранилище систем поддержки принятия решений 424
- Акопов А. С., Бекларян А. Л., Хачатрян Н. К., Фомин А. В.** Система прогнозирования динамики добычи нефти с использованием имитационного моделирования 431
- Булычев Г. Г.** Моделирование динамики полого цилиндра методом пространственных характеристик 437

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Машкина И. В., Куприянов А. О.** Классификационная схема и модели современных вирусных программ 443

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Загидуллин Р. Ш., Черников А. С., Чибисов А. А.** Решение проблемы объединения сред MOODLE и Электронный университет МГТУ им. Н. Э. Баумана 449
- Юсупова Н. И., Сметанина О. Н., Гаянова М. М.** Модели и методы обработки информации при анализе образовательных программ (на примере подготовки информатиков) 456
- Гаврилова И. Л.** Опыт разработки и внедрения в учебный процесс вуза электронного учебно-методического комплекса 464

БАЗЫ ДАННЫХ

- Саак А. Э.** Диспетчеризация массивов с заявками параболического типа в Grid-системах 469

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОРГАНИЗАЦИОННЫХ И СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ

- Сумин В. И., Смоленцева Т. Е.** Формирование структуры иерархических многоуровневых организационных систем 476

Главный редактор:

СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.
ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.
КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.
ПОПКОВ Ю. С.,
акад. РАН, д. т. н., проф.
РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.
РЯБОВ Г. Г.,
чл.-корр. РАН, д. т. н., проф.
СОЙФЕР В. А.,
акад. РАН, д. т. н., проф.
СОКОЛОВ И. А., акад.
РАН, д. т. н., проф.
СУЕТИН Н. В., д. ф.-м. н., проф.
ЧАПЛЫГИН Ю. А.,
акад. РАН, д. т. н., проф.
ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.
ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.
ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.
АНТОНОВ Б. И.
БАРСКИЙ А. Б., д. т. н., проф.
ВАСЕНИН В. А., д. ф.-м. н., проф.
ВИШНЕКОВ А. В., д. т. н., проф.
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.
ДОМРАЧЕВ В. Г., д. т. н., проф.
ЗАБОРОВСКИЙ В. С., д. т. н., проф.
ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.
ЗАРУБИН В. С., д. т. н., проф.
КАРПЕНКО А. П., д. ф.-м. н., проф.
КОЛИН К. К., д. т. н., проф.
КУЛАГИН В. П., д. т. н., проф.
КУРЕЙЧИК В. В., д. т. н., проф.
ЛЬВОВИЧ Я. Е., д. т. н., проф.
МАРТЫНОВ В. В., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.
НЕЧАЕВ В. В., к. т. н., проф.
ПОЛЕЩУК О. М., д. т. н., проф.
САКСОНОВ Е. А., д. т. н., проф.
СОКОЛОВ Б. В., д. т. н., проф.
ТИМОНИНА Е. Е., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)
ФОМИЧЕВ В. А., д. т. н., проф.
ШИЛОВ В. В., к. т. н., доц.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.

Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

CONTENTS**INTELLIGENT SYSTEMS AND TECHNOLOGIES**

- Ermakov A. E.** Semantic Transformations Language for Computer Interpretation of Text 403
- Levin V. I.** Maximal and Minimal Polyintervals and Optimization in Condition of Uncertainty 413
- Romanov A. Yu., Lomotin K. E., Kozlova E. S.** Machine Learning Methods Application to Automatic Scientific Articles Classification According to UDC 418

MODELING AND OPTIMIZATION

- Akimkina E. E.** Instrument Approach to the Organization of Data Collection Warehouse Decision Support Systems 424
- Akopov A. S., Beklaryan A. L., Khachatryan N. K., Fomin A. V.** The Forecasting System Dynamics of Oil Production using Simulation Modeling 431
- Bulychev G. G.** Simulation of Dynamics of the Hollow Cylinder by the Method of Spatial Characteristics 437

CRYPTOSAFETY INFORMATION

- Mashkina I. V., Kupritanov A. O.** The Classification Scheme and Modern Viruses Models 443

INFORMATION TECHNOLOGIES IN EDUCATION

- Zagidullin R. S., Chernikov A. S., Chibisov A. A.** Integration of Moodle and Electronic University Systems at BMSTU 449
- Yusupova N. I., Smetanina O. N., Gayanova M. M.** Models and Methods of Information Processing in the Analysis of Educational Programs (on the Example of Training Computer Scientists) 456
- Gavrilova I. L.** Experience in Development and Implementation of the Learning Process of the University Electronic Educational Complex 464

DATABASE

- Saak A. E.** Sheduling of Sets of Parabolic Tasks in Grid Systems 469

INFORMATION TECHNOLOGIES IN THE ORGANIZATIONAL AND SOCIO-ECONOMIC SYSTEMS

- Sumin V. I., Smolentseva T. E.** The Formation of the Structure of Hierarchical Multi-Level Organizational Systems 476

Editor-in-Chief:

Stempkovsky A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.
Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.
Popkov Yu. S., Member of RAS,
Dr. Sci. (Tech.), Prof.
Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Soifer V. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.
Chaplygin Yu. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.
Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Vishnekov A. V., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.
Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kureichik V. V., Dr. Sci. (Tech.), Prof.
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Martynov V. V., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Poleschuk O. M., Dr. Sci. (Tech.), Prof.
Saksonov E. A., Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Timonina E. E., Dr. Sci. (Tech.), Prof.
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.
Grigorin-Ryabova E. V.
Lysenko A. V.
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

УДК 004.8

А. Е. Ермаков, канд. техн. наук,
ведущий специалист по компьютерной лингвистике, e-mail: ermakov@rco.ru,
ООО "ЭР СИ О", Москва

Язык семантических трансформаций для компьютерной интерпретации текста

Статья посвящена развитию подхода к семантической интерпретации текста, описанному автором в 2009 г. Подход основан на преобразовании результатов машинного синтаксического анализа предложения в сеть связанных фреймов, с последующей интерпретацией определенных конфигураций связей как новых фреймов, описывающих требуемые ситуации предметной области. Разработан формальный язык, позволяющий описать и нужным образом интерпретировать сколь угодно сложные конфигурации синтаксических связей между словами текста посредством описания и интерпретации отдельных пар связей между словами. Рассмотрены практические приложения к извлечению из текста описаний событий и фактов, а также к интерпретации оценочных высказываний об объектах.

Ключевые слова: компьютерный анализ текста на естественном языке, семантические представления, синтаксический анализ текста, семантическая интерпретация текста, сеть фреймов, трансформация пропозиций, извлечение событий и фактов, анализ оценочных высказываний

Введение

Начиная с 1970-х годов, существуют и развиваются различные подходы к семантическому представлению и интерпретации значений высказываний на естественном языке (ЕЯ), используемые в системах компьютерного анализа текстов. К основным из зарубежных подходов можно отнести теорию представления дискурсов, теорию концептуальных графов, эпизодическую логику [1], фреймово-семантический парсинг [2], подход на основе нотации абстрактного представления смысла [3]. Обстоятельным, но уже неполным введением в современное состояние компьютерной обработки ЕЯ является справочник [4].

Пионером в области формализации семантики ЕЯ в конце 1960-х годов стал американский философ и специалист в области математической логики Р. Монтегю. Согласно его идеям, значение ЕЯ-высказывания может быть представлено выражением на метаязыке — логической формулой, переменными которой являются значения элементов высказывания — слов и синтаксических конструкций. Монтегю разработал и первый формальный метаязык для интерпретации значений английских предложений [5] в терминах логики предикатов. В нашей стране основоположниками разработки метаязыка на базе аппарата лексических функций в начале 1970-х годов выступили Ю. Д. Апресян [6]

и И. А. Мельчук [7], решавшие задачу автоматического перевода. Современные достижения возглавляемого сегодня Апресяном коллектива воплощены в системе автоматического перевода ЭТАП-3 и ряде других приложений [8]. Например, с помощью аппарата лексических функций одно из значений слова *потушить* может быть описано формулой $Perf\ Caus(Im, Fin\ Lab(Вин, ОГОНЬ))$, где $ОГОНЬ$ — это базисное, далее не разложимое понятие, а $Perf, Caus, Fin, Lab$ — это базисные лексические функции, Im и $Вин$ — обозначения падежей слов-аргументов функций. Так, базисная функция $Lab(x, y)$ обозначает, что аргумент, обозначаемый словом x , подвергается действию аргумента, обозначаемого словом y . Значение предложения описывается математическим выражением. Например, предложению *Пожарные потушили загоревшийся склад* соответствует выражение $Perf\ Caus(ПОЖАРНЫЕ, Fin\ Lab(Perf\ Incep\ Lab(СКЛАД, ОГОНЬ), ОГОНЬ))$, которое в обратном переводе на русский язык имеет следующее значение: *Пожарные сделали так, что перестал подвергаться действию огня начавший подвергаться действию огня склад*. Задача создания семантического метаязыка заключается в выборе системы базисных функций и понятий, позволяющих описать значения всех других понятий и предложений, дальнейшее толкование которых либо невозможно, либо нецелесообразно. Так, метаязык, разработанный под руко-

водством В. А. Тузова и описанный в работе [9], содержит 72 базисные функции и около 600 базисных понятий, на основе которых была сделана попытка описать значения всех слов русского языка в целях последующей семантической интерпретации любого русского предложения.

Современные обзоры формальных семантических описаний и их приложений можно найти в работах зарубежных авторов [10, 11]. Дополнительно следует отметить широту перспектив для компьютерной семантики ЕЯ, открываемых теорией К-представлений — концептуальных представлений, которая развивается в Москве с 1980-х годов В. А. Фомичевым. Первоначально теория К-представлений (ТКП) называлась теорией К-исчислений и К-языков. ТКП является центральной частью Интегральной формальной семантики ЕЯ — научного направления на стыке математической информатики, математической лингвистики и компьютерной лингвистики [14].

В статье [12] и монографиях [13, 14] предлагается оригинальная (по гипотезе автора — универсальная) математическая модель для описания структурированных значений предложений и связанных ЕЯ-текстов, а также содержания посланий компьютерных интеллектуальных агентов в многоагентных системах [15]. Модель представляет собой определение нового класса формальных языков, названных стандартными концептуальными языками (СК-языками). Проведено исследование выразительных возможностей СК-языков и показано, что выражения СК-языков удобно использовать для построения семантических описаний предложений, выражающих высказывания, вопросы, команды, распоряжения, советы, включая предложения со сложными описаниями множеств, однородными членами предложения, причастными оборотами, придаточными цели и придаточными определительными предложениями. Проведено сравнение выразительных возможностей СК-языков с выразительными возможностями других, наиболее часто используемых подходов к формальному представлению значений ЕЯ-текстов: теории представления дискурсов, теории концептуальных графов, эпизодической логики, теории расширенных семантических сетей, теории неоднородных семантических сетей и компьютерной семантики русского языка. Работы [13, 14] также описывают алгоритмы семантико-синтаксического анализа ЕЯ-текстов, предоставляемые ТКП. В работе [14] представлено применение разработанных алгоритмов к проектированию компьютерных систем обработки ЕЯ-текстов.

Для иллюстрации выразительных возможностей наиболее развитых современных языков описания семантики приведем ряд интуитивно понятных примеров описания ЕЯ-выражений средствами СК-языков из работы В. А. Фомичева [15]:

"Профессор Новиков прилетел вчера из Праги" = > Ситуация(e1, прилет*(Время, x1) (Агент1, нек. чел.* (Квалиф, профессор) (Фамилия, "Новиков"): x2) (Местол, нек. город*(Название, "Прага"): x3)) ^ Раньше (x1, Сейчас);

"3 контейнера с керамикой из Индии" = > нек. множ*(Колич, 3) (Кач-состав, Контейнер1*(Содерж1, нек. множ*(Кач-состав, изделие*(Вид, керамика) (Страна, Индия)));

"Партия керамики, состоящая из коробок с номерами 3217, 3218, 3219" = > нек. партия2*(Колич, 3) (Предм-состав, (нек. коробка1*(Номер, 3217): x1 ^ нек. коробка1*(Номер, 3218): x2 ^ нек. коробка1*(Номер, 3219): x3));

Подобные метаязыки необходимы в компьютерных приложениях, требующих максимально подробной интерпретации смысла ЕЯ-выражений: в программах автоматического перевода, в вопросно-ответных и экспертных системах (наиболее мощная из которых — IBM Watson: https://ru.wikipedia.org/wiki/IBM_Watson) и даже в системах общения компьютерных агентов, ставших предметом практического исследования в последнее десятилетие. В то же время на практике существует множество задач, для которых, с одной стороны, выразительная способность таких языков оказывается избыточной, а с другой стороны, существуют значительные ограничения на человеческие ресурсы, которые могут быть затрачены для построения детальных семантических описаний значений всех слов и ЕЯ-конструкций в предметной области работы системы. К числу таковых относятся, например, задача извлечения из текста описаний заданных событий и фактов с их участниками, а также задача интерпретации высказываний об объектах как позитивных/негативных оценок этих объектов, известная также под названием "оценка тональности высказываний". Решением двух указанных задач автор активно занимался на протяжении последних десяти лет в компаниях "ЭР СИ О" (<http://www.rco.ru>) и "Крибрум" (<http://www.kribrum.ru>), специализирующихся на компьютерной лингвистике и анализе Интернет-контента, практически воплощая описываемый в данной статье подход к семантической интерпретации текста на основе традиционных семантических представлений в форме сетей фреймов.

В рамках данного подхода разобранный в начале статьи фраза *Пожарные потушили загоревшийся склад* может быть семантически проинтерпретирована как пара фреймов, описывающих пару связанных фактов: ПОТУШИТЬ { Субъект = ПОЖАРНЫЕ, Объект = СКЛАД } & ЗАГОРЕТЬСЯ {Субъект = СКЛАД}. Далее, при наличии дополнительных правил интерпретации конструкций с глаголами *потушить* и

загореться может быть сделан вывод, что в тексте описана негативная ситуация для объекта *склад* и позитивная для объекта *пожарные*. Если в тексте дополнительно указана информация, позволяющая идентифицировать данные объекты, например, *пожарные МЧС* и *склад ГСМ РЖД*, то фраза будет проинтерпретирована как позитивная оценка для МЧС и негативная для РЖД. Применение же трех правил интерпретации общезыковых конструкций *приступить к (чему-то)*, *завершение (чего-то)* и *операция по (чему-то)* позволит провести полностью аналогичную интерпретацию фразы *Пожарные приступили к завершению операции по тушению загоревшегося склада*, не вдаваясь в семантическое описание избыточных деталей.

Развиваемый автором подход, базовая реализация которого была разработана в компании "ЭР СИ О" и описана ранее в работе [16], основан на формальном описании подграфов в сети синтактико-семантических отношений между словами ЕЯ-выражения, соответствующих различным способам языкового описания искомой ситуации, с последующей интерпретацией определенных узлов подграфа как слотов фрейма в определенных ролях. Для каждого типа ситуаций создаются шаблоны — графы-образцы, которые ищутся как подграфы в сети синтактико-семантических отношений между словами, построенной синтаксическим анализатором, что позволяет абстрагироваться от особенностей поверхностно-синтаксической организации предложений, обеспечивая высокую инвариантность шаблонов к способам выражения ситуации в тексте. Существенными недостатками метода являются: необходимость использования графических средств просмотра/редактирования шаблонов в форме графов, громоздкость и сложность просмотра описаний, вызванные тем, что одновременная работа с множеством графов-шаблонов, каковых для одной ситуации-фрейма бывает более десятка, затруднительна.

Излагаемое далее развитие подхода избавлено от указанных недостатков и описывает языковые способы выражения ситуаций в форме выражений на специальном языке, названном автором языком семантических трансформаций (ЯСТ).

Семантический интерпретатор и сети фреймов

В терминологии искусственного интеллекта фреймом называется схема ситуации или предмета. Фрейм имеет имя, которое идентифицирует класс описываемых им ситуаций или предметов, а также содержит слоты, которые имеют свои имена, идентифицирующие роли участников ситуации или ее характеристики, части или свойства предмета: *ПОКУПКА* { *Покупатель*, *Продавец*, *Товар*, *Цена|Стоимость* }; *АВТОМОБИЛЬ* { *Марка*, *МощностьДвигателя*, *МаксимальнаяСкорость...* }. Фрейм со слотами, заполненными текстовыми

значениями, представляет конкретную ситуацию, описанную в тексте, и называется фреймом-экземпляром: *ПОКУПКА* { *Покупатель* = *КОРЕЙКО АЛЕКСАНДР*, *Продавец* = *БЕНДЕР ОСТАП*, *Товар* = *ПАПКА*, *Цена|Стоимость* = *МИЛЛИОН РУБЛЕЙ* }.

Семантический интерпретатор (СИ) работает с представлением текста каждого предложения в форме сети фреймов-экземпляров, в которой каждому слову предложения соответствует свой фрейм. Изначально имена фреймов представляют нормальные формы соответствующих слов. Фрейм имеет набор атрибутов — пар вида *ИмяАтрибута* = *ЗначениеАтрибута*, например, атрибуты с именем *semantic*, которые принимают значения обобщенной части речи (*semantic* = *_Noun*, *semantic* = *_Verb...*) и семантического разряда слова (*semantic* = *Event*, *semantic* = *Feature*, *semantic* = *Object:Animated...*). В формальном языке СИ имя фрейма представляет собой значение особого атрибута с именем *name*, при этом фрейм может иметь множество альтернативных имен: *name* = *покупать|покупка|купля-продажа*. Фрейм может представлять не только отдельное слово, но и многословную сущность, выступающую в языке как единое целое: *name* = *ООО "Рога и копыта"*, *semantic* = *_Noun|Organization:Named*, или *name* = *9 мая 1945*, *semantic* = *_Adverb|Date*. Слоты фрейма могут быть заполнены ссылкой на другие фреймы, обычно представляющие синтаксически подчиненные слова. Строковым значениям слотов соответствуют имена связанных фреймов. Вхождение фрейма В во фрейм А в качестве слота означает наличие направленной связи от фрейма А к фрейму В. Имена слотов соответствуют типам синтактико-семантических связей с их атрибутами, которые устанавливаются между словами синтаксическим анализатором текста.

Изначально фреймовое представление эквивалентно представлению в форме сети синтактико-семантических отношений [16], но в ходе работы СИ сеть пополняется новыми фреймами, не соответствующими никаким словам исходного текста, а также новыми связями между уже существующими и новыми фреймами, причем исходные фреймы вместе со своими связями могут удаляться из сети.

Например, фраза *Из карабина деда барон послал пулю в дикую утку* после синтаксического анализа преобразуется в следующую сеть связанных фреймов:

БАРОН, *ДЕД*, *ПУЛЯ*, *ДИКИЙ* — элементарные фреймы, не имеющие слотов.

```
КАРАБИН { #_Gen|Владелец = ДЕД }
ПОСЫЛАТЬ { #_Nom|Субъект = БАРОН,
#_Acc|Объект = ПУЛЯ, #в_Асс|Цель|Место =
УТКА, #из_Gen|Обстоятельство = КАРАБИН }
УТКА { Attribute = ДИКИЙ }
```

Здесь каждая строка начинается с имени фрейма, после которого в фигурных скобках перечис-

ляются его слоты в форме `СписокИменСлота = ИмяФреймаВСлоте`. Осмысленные имена слотов вида `Цель|Место` устанавливаются на основе данных, указанных в словаре моделей управления предикатов естественного языка, и могут вообще отсутствовать. Альтернативу им представляют универсальные имена слотов, которые устанавливаются на основе общего алгоритма преобразования синтактико-семантических отношений [16] и присутствуют обязательно, например: `Attribute` (тип связи), `#из_Gen` (`#предлог_ПадежСвязи` типа "управление с предлогом"), `#_Acc` (`#_ПадежСвязи` типа "управление без предлога").

Применение определенного правила СИ приведет к добавлению в сеть нового фрейма с полностью "осмысленными" ролями слотов: `СТРЕЛЬБА { Субъект|Стрелок = БАРОН, Объект|Цель = УТКА, Обстоятельство|Оружие = КАРАБИН, Обстоятельство|Снаряд = ПУЛЯ }`, что представляет собой семантическую интерпретацию фразы. Дополнительно правило позволяет удалить из сети проинтерпретированный фрейм `ПОСЫЛАТЬ`, осуществив таким образом полную семантическую трансформацию.

Семантические трансформации пропозиций и язык их описания

Пропозиция, по [17], — это языковой способ описания ситуации с характерной конфигурацией синтаксических связей, аналог глубинно-синтаксической структуры по [7]. Примеры различных пропозиций, выражающих одну и ту же ситуацию: *Х решил проблему. Х сумел решить проблему. Х добился успеха в решении проблемы. Х сумел добиться успеха в решении проблемы. Х использовал свой шанс добиться успеха в решении проблемы.* Первая пропозиция является базовой и выражена так называемой изосемической изоморфной конструкцией. Заметим, что вариации *Х решил проблему, проблема решена Х-м, решившему проблему Х-у* относятся к той же самой пропозиции, реализованной различными поверхностно-синтаксическими способами, и после синтаксического анализа и так называемых постсинтаксических трансформаций связей [16] будут представлены полностью идентичными сетями фреймов.

Правила семантической трансформации пропозиций позволяют приводить сколь угодно сложные пропозиции к базовым. Ключевая идея подхода заключается в том, что для семантической интерпретации любой пропозиции в требуемый фрейм достаточно описать трансформации всех пар входящих в нее элементарных пропозиций и применять их итерационно к результату предшествующих трансформаций до тех пор, пока сеть фреймов не перестанет трансформироваться. Полный набор правил трансформаций элементарных пропозиций обеспечивает корректную, с точки зрения здравого

смысла, интерпретацию любой пропозиции, допустимой в языке. Так, представленные выше пропозиции могут быть трансформированы к базовой применением следующих трансформаций элементарных пропозиций: *Х сумел "сделать"* -> *Х "сделал"*, *Х добился успеха в Y* -> *Х преуспел в Y*, *Х преуспел в решении Y* -> *Х решил Y*, *Х использовал шанс "сделать"* -> *Х "сделал"*, где "сделать" условно обозначает любой глагол в инфинитивной форме.

Типичное правило семантической трансформации имеет следующий вид:

```
verb { name = посылать|послать|отправлять|отправить } : Объект = пуля
-> стрельба { Субъект|Стрелок = verb:Субъект, Объект|Цель = verb:#в_Acc, Обстоятельство|Оружие = verb:#из_Gen, Обстоятельство|Снаряд = пуля } & del(verb)
```

Символ '->' отделяет левую часть правила, в которой задаются ограничения на пару связанных фреймов, от правой, в которой задаются действия по трансформации сети фреймов в том ее фрагменте, который удовлетворяет ограничениям из левой части. В левой части правила `Объект` задает имя слота, с которым фрейм `пуля` должен входить в связанный фрейм с атрибутом `name = посылать`, либо с другими перечисленными через оператор `|` именами. Псевдоним `verb`, произвольно выбранный для этого фрейма, используется, чтобы однозначно сослаться на него в правой части правила, так как он, в отличие от фрейма `пуля`, допускает множество альтернативных имен. В правой части `стрельба` задает имя нового создаваемого фрейма, а записи в фигурных скобках вида `Имя1Слота|Имя2Слота = ИмяФрейма:ИмяСлота` задают способ заполнения слотов нового фрейма: все фреймы, входящие во фрейм `ИмяФрейма` из левой части правила в качестве слотов в роли `ИмяСлота`, войдут в новый фрейм как слоты с именами `Имя1Слота|Имя2Слота`. Оператор `del()` инициирует удаление заданного фрейма-аргумента.

Формально синтаксис языка семантических трансформаций можно описать так:

```
FrameLeft:ИмяСлота = FrameLeft ->
FrameRight (& FrameRight)* (& ParentExpression)* (& del(ИмяФрейма))* , где:
FrameLeft:: = ИмяФрейма / ПсевдонимФрейма { Атрибут = Значение(|Значение)* (, Атрибут = Значение(|Значение))* }
FrameRight:: = FrameName { TransformRule (,TransformRule)* }
TransformRule:: = (ИмяСлота(|ИмяСлота)* = FrameName / FrameName:ИмяСлота) / (ИмяАтрибута = $Значение)
FrameName:: = ИмяФрейма / ПсевдонимФрейма
ParentExpression = parent(FrameName) { ИмяСлота(|ИмяСлота)* = FrameName }
```

Здесь кириллицей обозначены элементарные выражения — произвольные текстовые строки, а латиницей — сложные выражения, которые раскрываются в последующих строчках после символа ':: = '. Другие специальные символы относятся к описанию формата правил: '/' разделяет альтернативные выражения, допустимые в одной позиции, '*' обозначает возможность повторения выражения от нуля до произвольного числа раз, круглые скобки группируют выражения для применения к ним '/' и '*' как к единому целому. Левая часть правила, до символа '->', описывает условия, которым должна удовлетворять пара связанных фреймов, при соблюдении которых выполняются трансформации, описанные в правой части, после символа '->'.

Элементарные условия на атрибуты фрейма задаются в левой части правила в виде `Атрибут = Значение (|Значение) *`, где возможные значения атрибута с именем `Атрибут` разделяются оператором альтернативы '|'. В случае указания нескольких элементарных условий через запятую комплексное условие в левой части считается выполненным в том случае, если выполнены все элементарные условия. Помимо точного сравнения атрибута с заданными значениями посредством оператора '=' в языке поддерживается сравнение с семантикой "значение не равно ни одному из заданных альтернативных" посредством оператора '! = ', а также сравнение по подстроке с маской '*', например, условие `semantic = Object*` будет выполнено для любых значений атрибута `semantic`, начинающихся с префикса `Object` типа `Object:Animated`, `Object:Inanimated:~Mental`.

Каждое подвыражение `TransformRule` в правой части правила описывает действие по заполнению слота фрейма с указанными именами `ИмяСлота` ссылкой на фрейм, обозначенный как `FrameName`; или ссылкой на фрейм в его слоте, обозначенный как `FrameName:ИмяСлота`. Подвыражение вида `ИмяАтрибута = $Значение` трактуется как добавление атрибута с заданным `ИмяАтрибута` и `Значение`.

`FrameName` в составе выражения `FrameRight` в правой части может соответствовать одному из двух `ИмяФрейма` (или `ПсевдонимФрейма`) в выражениях `FrameLeft` в левой части правила — тогда указанные в подвыражениях `TransformRule` действия по заполнению слотов или добавлению атрибутов осуществляются с уже существующим фреймом. Если `FrameName` в правой части не совпадает ни с одним из `ИмяФрейма` или `ПсевдонимФрейма` в левой части, то в сети создается новый фрейм с именем `FrameName`, к которому и применяются действия, описанные в подвыражениях `TransformRule`.

ПсевдонимФрейма вместо `ИмяФрейма` в выражении `FrameLeft` используется тогда, когда правило допускает несколько имен у фрейма —

тогда таковые перечисляются как альтернативные значения атрибута `name` в фигурных скобках, как в приведенном выше примере правила: `verb { name = посылать|послать|отправлять|отправить }.` Введение псевдонима фрейма в случае множества альтернативных имен обеспечивает однозначность указания на фрейм в правой части правила: `Субъект|Стрелок = verb:Субъект.` ПсевдонимФрейма может иметь особый вид: `frame` или `frameЛюбойТекст`, что указывает на фрейм с любым именем, для которого при необходимости в фигурных скобках могут быть заданы дополнительные ограничения на прочие атрибуты. Ниже приведен пример соответствующего правила, которое добавляет в сеть фреймов новый фрейм `ДЕД` с атрибутами `semantic = _Noun` и `semantic = Object:Animated`, а затем добавляет его в существующий фрейм в качестве слота с именем `Владелец`, в итоге преобразуя пропозиции вида *дедовский "предмет" в "предмет" деда*:

```
frame:Атрибут = дедовский -> дед { semantic = $_Noun, semantic = $Object:Animated } & frame { Владелец = "дед" }
```

В правой части правила вместо `ИмяСлота` может использоваться символ `*` (не отражено в формальном описании синтаксиса во избежание путаницы со служебным символом `*`, обозначающим возможность произвольного повторения выражения в правиле), что означает обращение ко всем слотам фрейма. Ниже представлен пример соответствующего правила, которое добавляет во фрейм `frame_verb` глагола в инфинитиве все слоты фрейма `verb_service` с их именами, в итоге преобразуя пропозиции вида *"субъект" "так-то" начал "действовать" в "субъект" "так-то" действует*, и добавляя служебный глагол `verb_service` во фрейм полнозначного глагола `frame_verb` в качестве слота с именем `Модус`:

```
verb_service { name = закончить|окончить|завершить }:#_Inf = frame_verb { semantic = _Verb } -> frame_verb { * = verb_service:*, Модус = verb_service }
```

Подвыражение `ParentExpression` в правой части правила означает, что фрейм `FrameName` из части `ИмяСлота (|ИмяСлота) * = FrameName` войдет в слоты всех фреймов-родителей, содержащих фрейм `FrameName` из части `parent (FrameName)` в своих слотах, с теми же именами. Это иллюстрирует правило, которое преобразует пропозиции типа *написать (новую) серию статей в написать (новые) статьи, размышлять над (огромным) множеством проблем в размышлять над (огромными) проблемами*:

```
noun { name = ряд|серия|множество|масса|куча }:#_Gen = frame -> frame { * = noun:* } & parent(noun) { * = frame }
```

Сначала правило добавляет все слоты фрейма noun (например, СЕРИЯ) в слоты фрейма frame (например, СТАТЬЯ), а затем во фрейм, содержащий фрейм noun (СЕРИЯ) в качестве слота, добавляет фрейм frame (СТАТЬЯ) в качестве слота с теми же именами. Фреймов-родителей, содержащих один и тот же фрейм noun в своих слотах, может быть несколько, например, в случае однородных членов оборота *написать серию статей, издав (их)* — тогда правило включит фрейм СТАТЬЯ в качестве слота во все фреймы-родители НАПИСАТЬ, ИЗДАТЬ.

Приложения семантического интерпретатора

Интерпретация описаний событий и фактов

Стандартной задачей современных систем извлечения знания из текста, к которым относят системы knowledge management (КМ) и business intelligence (BI), системы сбора фактографической информации и ведения конкурентной разведки, является задача выделения в тексте описаний заданных классов ситуаций (событий или фактов), выраженных разными языковыми способами, в единообразной структурированной форме. Такой форме соответствует представление ситуаций фреймами, с которым работает СИ.

Первое из приведенных в статье правил семантических трансформаций позволяет интерпретировать пропозиции вида *Стрелок послал пулю в Цель из Оружия* как фрейм-ситуацию СТРЕЛЬБА { Стрелок, Цель, Оружие }.

Рассмотрим пример извлечения факта — фрейма ЗАКЛЮЧЕНИЕ_ДОГОВОРА { Сторона, Договор, Содержание }.

Правило 1 интерпретирует пропозиции вида *Сторона поставила подпись под Объектом* как *Сторона подписала Объект*:

```
v { name = поставить|ставить } :Объект =
подпись -> подписывать { Субъект = v:Субъект,
Объект = v:#под_Gen }
```

Правило 2 извлекает целевой факт в структурированной форме как фрейм ЗАКЛЮЧЕНИЕ_ДОГОВОРА, интерпретируя пропозиции вида *Сторона1 подписала Договор со Стороной2 о Содержании; Сторона1 и Сторона2 подписали договор о Содержании*, которые могли быть сформированы в результате трансформации сети фреймов Правил 1, либо изначально присутствовать в исходной сети:

```
verb { name = подписать|подписывать } :
Объект = contract { name = договор|соглашение|контракт } ->
Заключение_Договора { Сторона = verb:Субъект,
Сторона = contract:#c_Gen,
Договор = contract,
Содержание = contract:#o_Prep }
```

Правило 3

```
решить: #_Inf = frame_v -> frame_v {
Субъект = решить:Субъект,
Квантор = решить:Квантор },
```

дает возможность трансформировать пропозиции вида *Субъект решил "действовать"* к виду *Субъект "действует"*. Дополнительное выражение *Квантор = решить:Квантор* позволяет "притянуть" к фрейму ДЕЙСТВОВАТЬ слоты с именем Квантор от фрейма РЕШИТЬ в качестве возможных показателей отрицания: *Субъект не решил "действовать"* -> *Субъект не "действует"*.

Правило 4

```
принять:Объект = решение -> решить { Субъект =
принять:Субъект, #_Inf = решение:
#_Inf }
```

позволяет трансформировать пропозиции вида *Субъект "таким-то образом" принял решение "действовать"* к виду пропозиции, далее интерпретируемому Правил 3: *Субъект "таким-то образом" решил "действовать"*.

В результате работы всех четырех правил будет проинтерпретирована как фрейм ЗАКЛЮЧЕНИЕ_ДОГОВОРА сложная пропозиция вида *Сторона1 приняла решение поставить подпись под Договором со Стороной2 о Содержании*.

Более оптимальный способ написания Правила 4 не создает в сети новый фрейм РЕШИТЬ, а просто добавляет к существующему фрейму РЕШЕНИЕ слот фрейма ПРИНЯТЬ с именем Субъект:

```
принять:Объект = решение -> решение {
Субъект = принять:Субъект },
```

преобразуя пропозиции вида *Субъект принял решение "действовать"* в *решение Субъекта "действовать"*. Для интерпретации такой пропозиции достаточно расширить Правило 3:

```
sol { name = решить/решение } :#_Inf =
frame_v -> frame_v { Субъект = sol:Субъект,
Квантор = sol:Квантор }
```

Заметим, что если в правой части Правила 3 не указывать конкретные имена слотов Субъект и Квантор, а разрешить любые, указав * = решить:*, то семантическая трансформация в ряде случаев повысит полноту интерпретации: *безумное решение Субъекта "действовать"* -> *Субъект "действует" безумно*, а в ряде случаев приведет к искажению исходного смысла, понизив точность: *Субъект давно решил "действовать"* -> *Субъект давно "действует"*, что не эквивалентно. Соотношение приоритетов полноты и точности, определяемое особенностями прикладной задачи извлечения событий и фактов, определяет способы написания правил. Теоретически язык семантических трансформаций позволяет достичь одновременно сколь угодно высоких полноты и точности интерпретации содержания текста

путем написания правил, строго учитывающих все конкретные слова предметной области, что, однако, является трудоемкой задачей, сопоставимой с описанием значений слов в более "сильных" подходах к семантике (см. Введение).

Интерпретация оценочных высказываний

Задача извлечения из текста выраженных в нем оценок объектов (товаров, организаций, персон) становится все более востребованной в связи с масштабным развитием социальных сетей в Интернет, сообщения которых переполнены прямыми и косвенными характеристиками различных объектов, их действий и свойств. Лингвистическая специфика данной задачи была описана автором в [18]. Точная интерпретация оценок, выраженных в тексте нетривиальными синтаксическими конструкциями, в том числе со сложным образом выраженными отрицаниями, эффективно реализуется описываемым здесь механизмом СИ путем выделения фреймов специального вида.

Фреймы TONALITY выделяются при прямом выражении позитивной или негативной оценки следующими типами пропозиций: *Объект отстойный*, *Объект считается отстойным/отстоем*, *Объект отстой*, *Объект — отстойное "нечто"*, *Объект "что-то делает" плохо*, где в позиции *Объект* также допускаются конструкции вида "*свойство*" *Объекта* (*политика X-а*, *интерфейс X-а*), а перечень конкретных слов, допустимых в позиции "*свойство*", задается для каждой целевой предметной области. Фреймы TONALITY могут содержать слоты со следующими именами:

- EstimatedObject — объект оценки и/или его свойство: то, что оценивают;
- Estimation — слово, выражающее оценку.

Аналогично выделяются оценки, в которых целевой объект выступает в роли синтаксического субъекта предиката, характеризующего своего субъекта позитивно или негативно: *Объект сломался*, *поломка Объекта*, *надежная работа Объекта*.

Правила семантических трансформаций для выделения фреймов TONALITY не содержат конкретных оценочных слов в позициях, выделяемых как слоты, что позволяет значительно уменьшить число правил, необходимых для описания целевой предметной области. Так, число универсальных оценочных слов, приложимых практически к любым объектам, в русском языке составляет несколько тысяч, а кроме них существует множество специальных слов, оценочная семантика которых меняется в зависимости от классов объектов и их свойств: *быстрая загрузка телефона* — позитив, *быстрая разрядка аккумулятора* — негатив. Поэтому для выделения фреймов TONALITY используется несколько общих правил, которые распознают все синтаксические конструкции, потенциально способные выражать оценочные пропозиции указан-

ных выше типов, затем создают фреймы TONALITY и добавляют к ним в качестве слотов все слова, стоящие в позициях EstimatedObject и Estimation, например, все согласованные с существительным определения-прилагательные (*быстрый*, *но ненадежный мобильный Интернет*):

```
frame_noun { semantic = _Noun }:Attribute
  { semantic = _Adjective } = frame_adj
-> tonality { EstimatedObject = frame_noun, Estimation = frame_noun:Attribute }
```

Окончательно фреймы TONALITY обрабатываются в программном коде: слова-значения слотов EstimatedObject проверяются на входение в словари наименований целевых объектов и их свойств, а значения слотов Estimation — на входение в словари "универсальный позитив/негатив" или "специальный позитив/негатив для целевых объектов/свойств в позиции EstimatedObject".

Фреймы SITUATION_EMOTION и SITUATION_ACTION выделяются при описании в тексте таких ситуаций, которые характеризуют целевой объект либо через его отношение к другим сущностям, либо через отношение других сущностей к нему:

- SITUATION_EMOTION — характеризуют отношение синтаксического субъекта предиката к его синтаксическому объекту посредством выражения эмоций: *обожаю Объект*, *только идиоты испытывают восторг от Объекта* и наоборот: *Объект не любит пользователей*, *Объект довел меня до истерики*. В ситуациях этого класса оценка объекта "позитив/негатив" зависит от оценки субъекта (*я, потребитель, народ...* — позитив; *идиот, чайник...* — негатив), и наоборот;
- SITUATION_ACTION — описывают определенное действие, которое приносит пользу или вред синтаксическому объекту предиката от синтаксического субъекта. В таких ситуациях оценка объекта не зависит от оценки субъекта: *я помогаю Объекту* — позитив, *эти негодяи оказывают помощь Объекту* — также позитив (для Объекта хорошо, кто бы ему не помогал), тогда как оценка субъекта, напротив, зависит от оценки объекта: *Объект принес мне пользу* — позитив, *Объект способствует коррупции* — негатив.

Фреймы SITUATION_EMOTION и SITUATION_ACTION могут содержать слоты:

- EstimatedObject — объект и/или его свойство, оценку которого описывает ситуация;
- EstimatingSubject — тот, кто оценивает или действует на объект в роли EstimatedObject;
- EstimatedObjectPositive — сущность, к которой объект в слоте EstimatingSubject относится положительно (*EstimatingSubject борется за EstimatedObjectPositive*);
- EstimatedObjectNegative — сущность, к которой объект в слоте EstimatingSubject от-

носятся отрицательно (*EstimatingSubject борется/против EstimatedObjectNegaitive*);

- Estimation — слово, называющее ситуацию (*восторг, бороться*).

Окончательно эти фреймы обрабатываются в программном коде: оценка целевого объекта в слоте EstimatedObject определяется на основе оценочной семантики слова в роли EstimationNegative или EstimationPositive, в соответствии с классом ситуации — именем фрейма SITUATION_EMOTION или SITUATION_ACTION. Соответственно, для каждого класса целевых объектов формируются соответствующие словари позитивных и негативных сущностей. Например, для оценки политических деятелей в глазах народа позитивными являются сущности *пенсионер (пенсионеры ненавидят Объект), заработная плата, рабочее место (действия Объекта привели к росту заработной платы и открытию новых рабочих мест)*, а отрицательными — *олигарх, чиновник (олигархи и чиновники ненавидят Объект), безработица, инфляция (действия Объекта привели к росту безработицы и инфляции)*.

Дополнительно существует класс правил, которые добавляют во все оценочные фреймы слоты с именем Negation — показатели отрицания для последующего инвертирования оценки в программном коде. Например, для пяти пропозиций вида *Объект не плохой / не ломается / не вызывает восторг; у Объекта нет/мало недостатков; Объект перестал работать/сбоить; Объект утратил преимущества / устранил недостатки; хорошая работа объекта — миф* показатели отрицания будут "притянуты" в целевые оценочные TONALITY и SITUATION следующими правилами:

```
frame:Quantifier = frame_neg { name = не|ни|якобы } -> frame { Negation = frame_neg }
frame_no { name = нет|мало }:#_Gen = frame_n -> frame_n { Negation = frame_no }
v { name = перестать|прекратить|отказаться }:#_Inf = frame -> frame { Negation = v, Negation = v:Negation }
v { name = утратить|потерять|устранить }:#_Acc = frame -> frame { Negation = v, Negation = v:Negation }
frame_n { semantic = _Noun }:Apposition = frame_neg { name = ложь|неправда|обман|миф|рассказни } -> frame_n { Negation = frame_neg, Negation = frame_neg:Negation }
```

Первое правило просто маркирует ряд элементарных отрицаний, синтаксически подчиненных соответствующим словам текста напрямую: находит частицы *не, ни, якобы* как слоты фреймов с именем Quantifier и добавляет этим слотам альтернативные имена Negation.

В последующих правилах слово, выражающее отрицание, добавляется в качестве слота Negation к соответствующему фрейму, а подвыражения Nega-

tion = v:Negation и Negation = frame_neg: Negation позволяют "притянуть" к этому фрейму дополнительные слоты-отрицания в пропозициях типа: *у Объекта якобы нет недостатков, Объект не прекращает сбоить, Объект не утратил достоинств, хорошая работа Объекта — не миф*.

Для интерпретации осложненных пропозиций типа *Объект якобы начал утрачивать преимущества* третьим и четвертым правилами необходимо наличие дополнительного правила, которое "притягивает" слоты-отрицания от вспомогательного глагола к управляемому им глаголу-инфинитиву:

```
v { name = начать|продолжить }:#_Inf = frame -> frame { frame:Субъект = v:Субъект, Negation = v:Negation }
```

Интерпретатор языка семантических трансформаций

Описанный в общих чертах выше семантический интерпретатор (СИ) представляет собой интерпретатор языка семантических трансформаций и работает следующим образом.

На этапе инициализации правил запускается парсер формального языка описания семантических трансформаций, который преобразует правила из текстовой формы во внутренние логические структуры данных для быстрого применения к сетям фреймов, а также строит индексные структуры для быстрого поиска возможных правил-кандидатов на применение к фреймам предложения на основе слов — имен фреймов, содержащихся в предложении и правилах.

Алгоритм СИ обеспечивает независимость результата применения правил семантических трансформаций от порядка их применения, который является произвольным в том смысле, что определяется исключительно внутренними сортировками структур данных для быстрого доступа. СИ обрабатывает текст последовательно по предложениям, применяя к сети фреймов предложения множество активных правил. В активное множество входят только те правила, которые либо содержат точное сравнение атрибута name со словами из предложения, либо содержат сравнения name с операторами != и *, либо вообще не содержат сравнений атрибута name (ограничения установлены на другие атрибуты).

На первой итерации к сети фреймов предложения применяется каждое правило из множества активных и, в случае удовлетворения какой-либо парой связанных фреймов условиям в левой части правила, проводится трансформация фреймов из этой пары или связанных с ними фреймов, описанная в правой части правила — добавление новых слотов или их имен, атрибутов. Уже существующие во фрейме слоты, их имена и значения атрибутов повторно не добавляются. Указатели на все фрей-

мы, трансформированные правилами, а также указатели на фреймы, связанные с трансформированным входящими и исходящими связями, запоминаются как активное множество фреймов. На следующей итерации активные правила применяются только к активным фреймам, трансформируя их и связанные с ними фреймы с формированием нового активного множества фреймов для новой итерации применения активных правил. В норме процесс трансформации сети сходится: после некоторой итерации не оказывается трансформированных фреймов, и семантическая интерпретация предложения завершается (хотя возможно написать "особые" правила для бесконечной трансформации сети).

Чтобы обеспечить инвариантность результата трансформаций к порядку применения правил, а также избежать неустойчивости процесса трансформаций, удаление фреймов со всеми входящими и исходящими связями, описанное в правилах оператором `del()`, осуществляется после завершения всех трансформаций, добавляющих информацию в сеть предложения.

В финале из сети фреймов могут быть удалены фреймы, полностью покрытые другими фреймами. Под покрытым понимается фрейм, все слоты которого входят в число слотов какого-либо другого, покрывающего фрейма, а сам покрытый фрейм не входит в качестве слота ни в какие другие фреймы. Это позволяет исключить из результатов неполные, промежуточные интерпретации, которые входят в состав более полных. Например, для фразы *замечательный Объект сломался* будет выделен оценочный фрейм `TONALITY { EstimatedObject = Объект, Estimation = замечательный }`, соответствующий подфразе *замечательный Объект*, который будет покрыт более полным оценочным фреймом `TONALITY { EstimatedObject = Объект, Estimation = замечательный, Estimation = сломаться }`, на основании которого и будет сформирована окончательная оценка высказывания — негатив (по логике принятия решения наличие хотя бы одного негативного слота `Estimation` во фрейме `TONALITY` формирует негативную оценку).

Заключение

Описанный в статье семантический интерпретатор успешно использован в электронных сервисах мониторинга информации в социальных сетях, предоставляемых системой "Крибрум" (<http://www.kribrum.ru>), для выявления позитивных/негативных оценок объектов мониторинга, сбора фактов по объектам мониторинга и авторам сообщений. На момент участия автора в проекте "Крибрум" в 2015 г. для интерпретации оценочных высказываний и фактов, несущих оценку объектов мониторинга, использовалось более 1700 правил семантических трансформаций, а для выделения

более 20 классов событий и фактов, излагаемых авторами сообщений от первого лица и характеризующих "профиль" автора, — более 400 правил. В настоящий момент автор развивает приложения семантического интерпретатора в проектах компании "ЭР СИ О" (<http://www.rco.ru>), где описанный подход был изначально рожден.

Список литературы

1. **Schubert L. K., Hwang C. H.** Episodic Logic meets Little Red Riding Hood: A comprehensive, natural representation for language understanding. Iwanska L., Shapiro S. C. (eds.) *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, MIT/AAAI Press, Menlo Park, CA, and Cambridge, MA, 2000, pp. 111–174.
2. **Das D., Chen D., Martins A. F. T., Schneider N., Smith N. A.** Frame-Semantic Parsing // *Computational Linguistics*. 2014. Vol. 40, N. 1. 4. P. 9–56.
3. **Banarescu L., Bonial C., Cai S., Georgescu M., Griffitt K., Hermjakob U., Knight K., Koehn P., Palmer M., Schneider N.** Abstract Meaning Representation for Sembanking. // *Proceedings of the 7th ACL Linguistic Annotation Workshop and Interoperability with Discourse*, Sofia, Bulgaria, August 8–9, 2013. 2013. URL: (www.aclweb.org/anthology/W13-2322; проверено 12.03.2016)
4. **Clark A., Fox C., Lappin S.** *The Handbook of Computational Linguistics and Natural Language Processing*. Wiley-Blackwell, 2010. 802 p.
5. **Dowty D. R., Wall R. E., Peters S.** *Introduction to Montague Semantics*. Dordrecht, Reidel, 1981. 316 p.
6. **Апресян Ю. Д.** *Лексическая семантика*. М.: Наука, 1974. 366 с.
7. **Мельчук И. А.** *Опыт теории лингвистических моделей "Смысл ⇔ Текст"*. М.: Наука, 1974. 314 с.
8. **Apresjan J. D., Boguslavsky I. M., Iomdin L. L., Tsinman L. L.** Lexical Functions in NLP: Possible Uses // *Computational Linguistics for the New Millenium: Divergence or Synergy?* *Proceedings of the International Symposium held at the Ruprecht-Karls-Universität Heidelberg. Festschrift in Honour of Peter Hellwig on the occasion of his 60th Birthday*. 21–22 July, 2000. Manfred Klenner, Henriëtte Visser (Eds.). Heidelberg: Peter Lang, 2002. P. 55–72.
9. **Тузов В. А.** *Компьютерная семантика русского языка*. СПб.: Изд-во СПбГУ, 2003. 400 с.
10. **Bos J.** A Survey of Computational Semantics: Representation, Inference and Knowledge in Wide-Coverage Text Understanding // *Language and Linguistics Compass*. 2011. N. 5/6. P. 336–366.
11. **Lappin S., Fox C.** *The Handbook of Contemporary Semantic Theory*. John Wiley & Sons, 2015. 704 p.
12. **Фомичев В. А.** Математические основы представления смысла текстов для разработки лингвистических информационных технологий. Часть I // *Информационные технологии*. 2002. № 10. С. 16–25; Часть II // *Информационные технологии*. 2002. № 11. С. 34–45.
13. **Фомичев В. А.** *Формализация проектирования лингвистических процессоров*. М.: МАКС Пресс. 2005. 368 с.
14. **Fomichev V. A.** *Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms*. New York, Dordrecht, Heidelberg, London: Springer, 2010. 354 p.
15. **Фомичев В. А.** Математические основы представления содержания посланий компьютерных интеллектуальных агентов. М.: ГУ-ВШЭ, издательство "ТЕИС", 2007. 176 с.
16. **Ермаков А. Е., Плешко В. В.** Семантическая интерпретация в системах компьютерного анализа текста // *Информационные технологии*. 2009. № 6. С. 2–7.
17. **Всеволодова М. В., Деменьтева О. Ю.** *Проблемы синтаксической парадигматики: коммуникативная парадигма предложений*. М.: КРОН-ПРЕСС, 1997. 176 с.
18. **Ермаков А. Е., Киселев С. Л.** Лингвистическая модель для компьютерного анализа тональности публикаций СМИ // *Компьютерная лингвистика и интеллектуальные технологии: труды Международной конференции Диалог'2005*. М.: Наука, 2005. С. 131–135.

Semantic Transformations Language for Computer Interpretation of Text

The article is devoted to the development of approach to semantic interpretation of natural language text. The approach was initially described by the author in 2009. The approach is implemented in two stages.

In the first stage, the syntactic parsing is performed for each phrase of the text, producing the network of connected frames. Each frame corresponds to some entity which was mentioned in the phrase of text. The entity could be a single word, a phrase or some name. The frame has attributes whose values correspond to the parts of speech and semantic attributes of the corresponding entity. The frame has a set of slots that contain links to other frames corresponding syntactically subordinated entities in the phrase. Slots are characterized by roles that correspond to the types of syntactic and semantic relationships between entities in the phrase.

At the second stage, the lookup of the frames subnetworks is performed under given restrictions. The restrictions on the frames subnetwork configuration describes the desired situation of the subject area — facts and events. Formally, these configurations correspond to the target rules described by a specially developed formal language — semantic transformations language. These rules allow to add new frames, new frame attributes, as well as new connection between the new and old frames. The semantic transformations language which allows us to describe and interpret arbitrarily complex configurations between frames is described in details. The ability by means of the approach to semantically interpret arbitrarily complex phrase by applying simple rules is illustrated in the article with a lot of examples.

The algorithmically efficient implementation of the semantic interpreter which provides a frames network transformation is described. The algorithm applies the rules by iterations until network is stabilized. The essential feature of the algorithm is invariance to the order of application of the rules.

The event and fact extraction and sentiment analysis tasks are considered as illustrating examples of the practical applications of this approach.

Keywords: computational natural language processing, computational semantic, semantic representation, formal semantic languages, syntactical analysis of text, semantic interpretation of text, frame network, transformation of semantic structure of phrase, events and facts extraction, sentiment analysis

References

1. **Schubert L. K., Hwang C. H.** Episodic Logic meets Little Red Riding Hood: A comprehensive, natural representation for language understanding. Iwanska L., Shapiro S. C. (eds.) *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, MIT/AAAI Press, Menlo Park, CA, and Cambridge, MA, 2000, pp. 111–174.
2. **Das D., Chen D., Martins A. F. T., Schneider N., Smith N. A.** Frame-Semantic Parsing, *Computational Linguistics*, 2014, vol. 40, no. 1. 4, pp. 9–56.
3. **Banarescu L., Bonial C., Cai S., Georgescu M., Griffitt K., Hermjakob U., Knight K., Koehn P., Palmer M., Schneider N.** Abstract Meaning Representation for Sembanking, *Proceedings of the 7th ACL Linguistic Annotation Workshop and Interoperability with Discourse*, Sofia, Bulgaria, August 8–9, 2013, 2013. URL: (www.aclweb.org/anthology/W13-2322); (data of access 12.03.2016)
4. **Clark A., Fox C., Lappin S.** *The Handbook of Computational Linguistics and Natural Language Processing*. Wiley-Blackwell, 2010. 802 p.
5. **Dowty D. R., Wall R. E., Peters S.** *Introduction to Montague Semantics*. Dordrecht, Reidel, 1981. 316 p.
6. **Apresjan Ju. D.** Leksicheskaja semantika (Lexical semantic), Moscow, Nauka, 1974. 366 p.
7. **Mel'chuk I. A.** Opyt teorii lingvisticheskikh modelej "Smysl \leftrightarrow Tekst" (The experience of linguistic models "Sense \leftrightarrow Text"), Moscow, Nauka, 1974. 314 p.
8. **Apresjan J. D., Boguslavsky I. M., Iomdin L. L., Tsinman L. L.** Lexical Functions in NLP: Possible Uses, *Computational Linguistics for the New Millenium: Divergence or Synergy? Proceedings of the International Symposium held at the Ruprecht-Karls-Universität Heidelberg. Festschrift in Honour of Peter Hellwig on the occasion of his 60th Birthday*. 21–22 July, 2000. Manfred Klenner, Henriëtte Visser (Eds.). Heidelberg: Peter Lang, 2002, pp. 55–72.
9. **Tuzov V. A.** *Komp'juternaja semantika russkogo jazyka* (Computational semantic of russian language). SPb.: Izd-vo SPbGU, 2003. 400 p.
10. **Bos J.** A Survey of Computational Semantics: Representation, Inference and Knowledge in Wide-Coverage Text Understanding, *Language and Linguistics Compass*, 2011, no. 5/6, pp. 336–366.
11. **Lappin S., Fox C.** *The Handbook of Contemporary Semantic Theory*. John Wiley & Sons, 2015, 704 p.
12. **Fomichev V. A.** Matematicheskie osnovy predstavlenija smysla tekstov dlja razrabotki lingvisticheskikh informacionnyh tehnologij (Mathematical base of sense of text representation for development of linguistic informational technologies). Part I, *Informacionnye tehnologii*, 2002, no. 10, pp. 16–25.; Part II, *Informacionnye tehnologii*, 2002, no. 11, pp. 34–45.
13. **Fomichev V. A.** *Formalizacija proektirovanija lingvisticheskikh processorov* (Formalization of linguistic processors designing), Moscow, MAKS Press. 2005. 368 p.
14. **Fomichev V. A.** *Semantics-Oriented Natural Language Processing: Mathematical Models and Algorithms*, New York, Dordrecht, Heidelberg, London: Springer, 2010. 354 p.
15. **Fomichev V. A.** *Matematicheskie osnovy predstavlenija sodержanija poslanij komp'juternyh intellektual'nyh agentov* (Mathematical base of representation of content of computer intelligent agents' messages). Moscow, GU-VShJe, izdatel'stvo "TEIS", 2007. 176 p.
16. **Ermakov A. E., Pleshko V. V.** Semanticheskaja interpretacija v sistemah komp'juternogo analiza teksta (Semantic interpretation in systems of machine analysis of text), *Informacionnye tehnologii*, 2009, no. 6, pp. 2–7.
17. **Vsevolodova M. V., Demen'teva O. Ju.** *Problemy sintaksicheskij paradigmatiki: kommunikativnaja paradigma predlozhenij* (Problems of syntactical paradigmatics: communicative paradigm of sentences), Moscow, KRON-PRESS, 1997. 176 p.
18. **Ermakov A. E., Kiselev S. L.** Lingvisticheskaja model' dlja komp'juternogo analiza tonal'nosti publikacij SMI (Linguistic model for machine analysis of tonality of public medic stories), *Komp'juternaja lingvistika i intellektual'nye tehnologii: trudy Mezhdunarodnoj konferencii Dialog'2005*, Moskva, Nauka, 2005, pp. 131–135.

В. И. Левин, д-р техн. наук, проф., e-mail: vilevin@mail.ru,
Пензенский государственный технологический университет

Максимальные и минимальные полиинтервалы и оптимизация в условиях неопределенности

Введено понятие сравнения полиинтервалов в целях выделения большего и меньшего полиинтервалов. Новое понятие основано на принятой в интервальной математике методике расширения числовых операций на интервалы. Доказана основная теорема, определяющая необходимые и достаточные условия сравнимости полиинтервалов. Приведен пример использования результатов статьи для принятия оптимальных решений в экономике.

Ключевые слова: полиинтервалы, их сравнение, максимальный (минимальный) полиинтервал

Введение

В период Второй мировой войны появилось много новых технологий: обнаружение воздушных целей с помощью радаров, управление огнем зенитной артиллерии, шифровка и дешифровка информации в системах связи и т.д. Все эти технологии были связаны с исследованием неопределенности и использовали соответствующие математические методы, главным образом, теорию вероятностей. После войны эти исследования были продолжены и распространены на гражданскую сферу — технику, экономику, социум. При этом под неопределенностью стали понимать не только случайность возможных исходов, но и их неединственность или незнание, дрейф переменных, семантическую неопределенность целей, многокритериальность при принятии решений, недоопределенность модели или структуры изучаемой системы и т.д.

Эти новые подходы к описанию неопределенности систем привели к созданию новых математических методов их изучения: теория нечетких множеств [1], многозначная и непрерывная логика [2], теория сверхслучайных процессов [3] и т.д. Одним из популярных методов стала также интервальная математика, изучающая величины, определяемые с точностью до интервалов возможных значений [4, 5]. Однако одиночные интервалы, изучаемые в интервальной математике, не охватывают всех практических ситуаций. Например, неопределенный период времени, в течение которого возможно проведение некоторой военной операции, может содержать несколько последовательных временных интервалов. Эти новые неопределенные объекты, имеющие вид последовательностей интервалов неопределенности, были введены в работе [6] и названы полиинтервалами. Полиинтервалы являются расширением интервальной модели неопределенности систем. В работе [6] было построено исчисление полиинтервалов, основанное на операциях над полиинтервалами, аналогичных операциям над интервалами в интервальной математике. Этими операциями являются сложение, вычитание, умно-

жение и деление. Данная статья посвящена построению и исследованию другой важной операции над полиинтервалами — сравнению. Эта операция вводится на основе теории сравнения интервалов [7] и может быть использована для оптимизации полиинтервальных величин и функций.

Постановка задачи

Базой для исчисления полиинтервалов [6] служит исчисление интервалов, называемое иначе интервальной математикой [4, 5]. Интервал вводится как множество всех возможных значений неполностью определенной величины \tilde{a} , задаваемой лишь ее нижней a_1 и верхней a_2 границами. Формально величину \tilde{a} можно записать в виде следующего числового множества — ограниченного интервала неопределенности:

$$\tilde{a} \equiv [a_1, a_2] = \{a \mid a_1 \leq a \leq a_2\}. \quad (1)$$

Согласно (1) неизвестное истинное значение неопределенной величины \tilde{a} достоверно лежит в пределах интервала $[a_1, a_2]$, не выходя за его границы a_1 и a_2 . При этом все значения величины \tilde{a} в пределах указанного интервала считаются равно возможными в том смысле, что нет никаких оснований предпочитать одно значение другому. Понятие равновозможности здесь не означает задание равномерного вероятностного или какого-либо иного равномерного распределения величины \tilde{a} внутри указанного интервала. Над интервалами вида (1) вводятся алгебраические операции, обобщающие соответствующие операции над числами. Для этого используется теоретико-множественная конструкция

$$\tilde{a} \circ \tilde{b} = \{a \bullet b \mid a \in \tilde{a}, b \in \tilde{b}\}, \circ \tilde{a} = \{\bullet a \mid a \in \tilde{a}\}. \quad (2)$$

Согласно (2), любая операция над интервалами (\circ) определяется на базе соответствующей операции над точными числами (\bullet), при условии, что конкретные значения этих чисел пробегают все возможные значения из соответствующих интерва-

лов. Из определения (2) следуют простые правила выполнения операций над интервалами:

$$\begin{aligned}
 [a_1, a_2] + [b_1, b_2] &= [a_1 + b_1, a_2 + b_2]; \\
 [a_1, a_2] - [b_1, b_2] &= [a_1 - b_2, a_2 - b_1]; \\
 k \cdot [a_1, a_2] &= \begin{cases} [ka_1, ka_2], k > 0, \\ [ka_2, ka_1], k < 0; \end{cases} \\
 [a_1, a_2] \cdot [b_1, b_2] &= [\min_{i,j} (a_i \cdot b_j), \max_{i,j} (a_i \cdot b_j)]; \\
 [a_1, a_2] / [b_1, b_2] &= [a_1, a_2] \cdot [1/b_2, 1/b_1] \text{ при } 0 \notin [b_1, b_2].
 \end{aligned}
 \tag{3}$$

Развивая понятие интервала, в работе [6] было введено понятие полиинтервала как последовательности нескольких непересекающихся одиночных интервалов

$$\tilde{A} = (\tilde{a}, \tilde{b}, \dots, \tilde{d}), \tag{4}$$

где $\tilde{a}, \tilde{b}, \dots, \tilde{d}$ — одиночные интервалы вида (1).

В (4) считается, что каждый следующий одиночный интервал сдвинут вправо от предыдущего и не пересекается с ним. Операции над полиинтервалами были введены в работе [6] аналогично операциям над интервалами с помощью теоретико-множественной конструкции типа (2):

$$\tilde{A} \circ \tilde{B} = \{a \cdot b \mid a \in \tilde{A}, b \in \tilde{B}\}, \circ \tilde{A} = \{a \mid a \in \tilde{A}\}. \tag{5}$$

Здесь \tilde{A} — полиинтервал вида (4), \tilde{B} — другой полиинтервал того же вида, но с другими составляющими его одиночными интервалами. На базе определения (5) операций над полиинтервалами в работе [6] были выведены правила конструктивного выполнения следующих алгебраических операций с полиинтервалами: сложение, вычитание, умножение полиинтервала на число, умножение и деление полиинтервалов. Задача настоящей статьи заключается в том, чтобы на базе того же определения вывести правило конструктивного выполнения еще одной, весьма важной операции над полиинтервалами — определение максимального и минимального из двух полиинтервалов. Важность этой операции связана с тем, что к ее выполнению (равно как и к выполнению аналогичной операции над одиночными интервалами) сводятся многие классы задач оптимального планирования разнообразных систем и процессов, функционирующих в условиях неопределенности [8]. На базе этой операции также появляется возможность сравнения интервалов и их упорядочения по отношениям "больше", "меньше" и "равно".

Решение задачи

Следуя работе [6], будем представлять полиинтервалы (4) в теоретико-множественных терминах следующим образом:

$$\tilde{A} = \tilde{a} \cup \tilde{b} \cup \dots \cup \tilde{d}. \tag{6}$$

Пусть заданы два полиинтервала \tilde{A} и \tilde{B} следующего вида:

$$\tilde{A} = \bigcup_{i=1}^m \tilde{a}^i, \tilde{B} = \bigcup_{j=1}^n \tilde{b}^j, \tag{7}$$

где $\tilde{a}^i = [a_1^i, a_2^i], i = \overline{1, m}$, и $\tilde{b}^j = [b_1^j, b_2^j], j = \overline{1, n}$, — одиночные интервалы, в совокупности составляющие \tilde{A} и \tilde{B} соответственно. Требуется определить максимальный и минимальный из этих интервалов. Другими словами, требуется выполнить операции

$$\tilde{C} = \tilde{A} \vee \tilde{B}, \tilde{D} = \tilde{A} \wedge \tilde{B}, \tag{8}$$

где \vee — операция взятия максимума, а \wedge — операция взятия минимума, а \tilde{C} и \tilde{D} — максимальный и минимальный интервалы. Эти две операции мы определим формально, как и другие операции над полиинтервалами, введенные в работе [6] (сложение, вычитание, умножение), с помощью теоретико-множественной конструкции (5). Таким образом, максимум и минимум двух полиинтервалов определяются в виде

$$\begin{aligned}
 \tilde{C} &= \tilde{A} \vee \tilde{B} = \{a \vee b \mid a \in \tilde{A}, b \in \tilde{B}\}, \\
 \tilde{D} &= \tilde{A} \wedge \tilde{B} = \{a \wedge b \mid a \in \tilde{A}, b \in \tilde{B}\},
 \end{aligned} \tag{9}$$

т.е. операции взятия максимума (минимума) двух полиинтервалов определяются на основе соответствующих операций над двумя точно заданными величинами при условии, что конкретные значения этих величин пробегают все возможные значения из соответствующих полиинтервалов.

Исходя из определений операции взятия максимума и минимума двух полиинтервалов (9) нетрудно установить формулу для конструктивного выполнения этих операций. Для этого используем следующую базовую формулу, позволяющую выполнить произвольную операцию \circ над этими полиинтервалами \tilde{A} и \tilde{B} в виде суперпозиции этой операции над одиночными интервалами \tilde{a}^i, \tilde{b}^j , составляющими \tilde{A} и \tilde{B} [6]:

$$\begin{aligned}
 \tilde{A} \circ \tilde{B} &= \bigcup_{i=1}^m \bigcup_{j=1}^n (\tilde{a}^i \circ \tilde{b}^j) \text{ или в развернутом виде} \\
 \left(\bigcup_{i=1}^m \tilde{a}^i \right) \circ \left(\bigcup_{j=1}^n \tilde{b}^j \right) &= \bigcup_{i=1}^m \bigcup_{j=1}^n (\tilde{a}^i \circ \tilde{b}^j). \tag{10}
 \end{aligned}$$

Подставляя теперь в формулу (10) вместо \circ конкретные операции \vee и \wedge , получим необходимые формулы для конструктивного выполнения операций взятия максимума и минимума двух полиинтервалов:

$$\begin{aligned}
 \tilde{A} \vee \tilde{B} &= \bigcup_{i=1}^m \bigcup_{j=1}^n (\tilde{a}^i \vee \tilde{b}^j) \text{ или в развернутом виде} \\
 \left(\bigcup_{i=1}^m \tilde{a}^i \right) \vee \left(\bigcup_{j=1}^n \tilde{b}^j \right) &= \bigcup_{i=1}^m \bigcup_{j=1}^n (\tilde{a}^i \vee \tilde{b}^j); \tag{11}
 \end{aligned}$$

$$\begin{aligned}
 \tilde{A} \wedge \tilde{B} &= \bigcup_{i=1}^m \bigcup_{j=1}^n (\tilde{a}^i \wedge \tilde{b}^j) \text{ или в развернутом виде} \\
 \left(\bigcup_{i=1}^m \tilde{a}^i \right) \wedge \left(\bigcup_{j=1}^n \tilde{b}^j \right) &= \bigcup_{i=1}^m \bigcup_{j=1}^n (\tilde{a}^i \wedge \tilde{b}^j). \tag{12}
 \end{aligned}$$

Формулы (11), (12) сводят вычисление максимума и минимума двух полиинтервалов \tilde{A} и \tilde{B} к вычислению максимумов и минимумов всех пар одиночных интервалов $(\tilde{a}^i, \tilde{b}^j)$, составляющих \tilde{A}

и \tilde{B} . Однако непосредственное использование этих формул для вычисления максимального (минимального) из полиинтервалов \tilde{A} и \tilde{B} не всегда удобно, так как, во-первых, это требует вычисление максимума (минимума) для каждой пары одиночных интервалов $(\tilde{a}^i, \tilde{b}^j)$, $i = \overline{1, m}, j = \overline{1, n}$, во-вторых не для каждой пары полиинтервалов \tilde{A} и \tilde{B} существует максимальный (минимальный) полиинтервал. Поэтому гораздо практичнее сначала установить существование максимального (минимального) из двух заданных полиинтервалов, используя подходящий критерий существования, и только после этого устанавливать максимальный и минимальный полиинтервал. Простой критерий существования максимального (минимального) из двух заданных полиинтервалов дает нижеследующая теорема. Она же сразу устанавливает, какой из полиинтервалов является максимальным, а какой — минимальным.

В работе [7] были введены отношения между интервалами на основе теории множеств. При этом для любых двух интервалов $\tilde{a} = [a_1, a_2]$, $\tilde{b} = [b_1, b_2]$ по определению принимается:

$$\tilde{a} = \tilde{b}, \text{ если } a_1 = b_1, a_2 = b_2; \quad (13)$$

$$\tilde{a} \geq \tilde{b}, \text{ если } \tilde{a} \vee \tilde{b} = \tilde{a}, \tilde{a} \wedge \tilde{b} = \tilde{b}; \quad (14)$$

$$\tilde{a} \text{ не сравнимо } \tilde{b}, \text{ если } \tilde{a} \neq \tilde{b}, \tilde{a} \not\geq \tilde{b}, \tilde{b} \not\geq \tilde{a}. \quad (15)$$

На основании определений (13)—(15) было показано [7], что 1) для того чтобы интервалы \tilde{a} и \tilde{b} находились в отношении $\tilde{a} \geq \tilde{b}$, необходимо и достаточно, чтобы выполнялись условия $a_1 \geq b_1, a_2 \geq b_2$; 2) для того чтобы интервалы \tilde{a} и \tilde{b} были несравнимы, необходимо и достаточно, чтобы выполнялись условия $a_1 < b_1, a_2 > b_2$ или $b_1 < a_1, b_2 > a_2$. Другими словами, для выполнения отношения $\tilde{a} \geq \tilde{b}$ требуется, чтобы интервал \tilde{a} был сдвинут относительно интервала \tilde{b} вправо обеими своими границами, а для несравнимости этих интервалов требуется, чтобы один из них (безразлично какой) полностью накрывал другой.

Отношения между полиинтервалами мы введем теперь аналогично отношению между интервалами. Именно, для любых двух полиинтервалов \tilde{A} и \tilde{B} вида (7) по определению принимаем

$$\tilde{A} = \tilde{B}, \text{ если } m = n, \tilde{a}^1 = \tilde{b}^1, \tilde{a}^2 = \tilde{b}^2, \dots, \tilde{a}^m = \tilde{b}^m,$$

$$\tilde{A} \geq \tilde{B}, \text{ если } \tilde{A} \vee \tilde{B} = \tilde{A}, \tilde{A} \wedge \tilde{B} = \tilde{B},$$

$$\tilde{A} \text{ не сравнимо } \tilde{B}, \text{ если } \tilde{A} \neq \tilde{B}, \tilde{A} \not\geq \tilde{B}, \tilde{B} \not\geq \tilde{A}.$$

Теорема. Для того чтобы два полиинтервала \tilde{A} и \tilde{B} вида (7) были сравнимы и находились в отношении $\tilde{A} \geq \tilde{B}$, необходимо и достаточно, чтобы входящий в состав \tilde{A} минимальный одиночный интервал \tilde{a}^1 и входящий в состав \tilde{B} максимальный одиночный интервал \tilde{b}^n находились в отношении $\tilde{a}^1 \geq \tilde{b}^n$.

Доказательство. Достаточность. Пусть выполнено условие $\tilde{a}^1 \geq \tilde{b}^n$. Тогда, по определению

полиинтервалов, для всех i, j справедливо неравенство $\tilde{a}^i \geq \tilde{b}^j$. Отсюда следует, что для всех i, j имеем $\tilde{a}^i \vee \tilde{b}^j = \tilde{a}^i, \tilde{a}^i \wedge \tilde{b}^j = \tilde{b}^j$. Таким образом, по формулам (11), (12) получаем

$$\tilde{A} \vee \tilde{B} = \bigcup_{i=1}^m \tilde{a}^i = \tilde{A}, \tilde{A} \wedge \tilde{B} = \bigcup_{j=1}^n \tilde{b}^j = \tilde{B}.$$

Два последних соотношения, согласно (17), означают, что полиинтервалы \tilde{A} и \tilde{B} сравнимы и находятся в отношении $\tilde{A} \geq \tilde{B}$, что и требовалось доказать.

Необходимость. Пусть два полиинтервала \tilde{A} и \tilde{B} сравнимы и находятся в отношении $\tilde{A} \geq \tilde{B}$. Тогда, в соответствии с определением этого отношения (17), верны следующие два равенства $\tilde{A} \vee \tilde{B} = \tilde{A}, \tilde{A} \wedge \tilde{B} = \tilde{B}$. Выражая в них операции \vee и \wedge по формулам (11), (12), а полиинтервалы \tilde{A}, \tilde{B} — по формулам (7), перепишем их в виде

$$\bigcup_{i=1}^m \bigcup_{j=1}^n (\tilde{a}^i \vee \tilde{b}^j) = \bigcup_{i=1}^m \tilde{a}^i,$$

$$\bigcup_{i=1}^m \bigcup_{j=1}^n (\tilde{a}^i \wedge \tilde{b}^j) = \bigcup_{j=1}^n \tilde{b}^j. \quad (16)$$

В первом выписанном равенстве правая часть зависит только от интервалов $\tilde{a}^i, i = \overline{1, m}$, поэтому, чтобы имело место равенство, его левая часть также должна зависеть только от интервалов $\tilde{a}^i, i = \overline{1, m}$, а это возможно только при выполнении условия $\tilde{a}^i \geq \tilde{b}^j$ для всех i, j . Аналогично выводится выполнение этого условия из второго выписанного равенства. Выполнение данного условия означает, что любой одиночный интервал \tilde{a}^i , входящий в состав полиинтервала \tilde{A} , находится в отношении \geq к любому одиночному интервалу \tilde{b}^j , входящему в состав полиинтервала \tilde{B} . В частности, справедливо отношение $\tilde{a}^1 \geq \tilde{b}^n$, что и требовалось доказать.

Формулы (11), (12) дают нам конструктивные правила выделения большего и меньшего из двух имеющихся полиинтервалов для тех случаев, когда они существуют. Теорема дает простое правило проверки существования большего и меньшего из двух имеющихся полиинтервалов. Теперь алгоритм решения различных задач исследования систем с полиинтервальными характеристиками, требующих сравнения полиинтервалов, можно представить следующим образом.

Шаг 1. Построение абстрактной математической модели, представляющей решение задачи как попарное сравнение некоторого числа полиинтервалов, являющихся числовыми значениями характеристик изучаемой системы в условиях неопределенности, в целях последующего выделения полиинтервалов, являющихся решением задачи.

Шаг 2. Анализ подлежащих сравнению пар полиинтервалов в целях выявления пар сравнимых и несравнимых полиинтервалов. Анализ проводится с помощью условий (16)—(18). При этом пары, удовлетворяющие условию (16) (равные полиинтервалы) или условию (17) (полиинтервалы, находящиеся

в отношении \geq), относим к парам сравнимых полиинтервалов, а пары, удовлетворяющие условию (18), — к парам несравнимых полиинтервалов.

Шаг 3. Построение структурной математической модели решения задачи в виде частично ориентированного графа, с использованием шагов 1, 2. Вершинами графа являются полиинтервалы, выделенные на шаге 1, его ребрами — линии, соединяющие вершины равных полиинтервалов, его дугами (ориентированными ребрами) — линии, соединяющие вершины неравных полиинтервалов в направлении от меньших полиинтервалов к большему. При этом вершины, отвечающие несравнимым полиинтервалам, не соединяются никакими линиями. При построении граф-модели используются пары равных полиинтервалов, пары полиинтервалов с отношением \geq и пары несравнимых полиинтервалов, полученные на шаге 2.

Шаг 4. Вычисление по структурной граф-модели вершины, соответствующей полиинтервалу, представляющему собой решение задачи. Чаще всего в качестве такого полиинтервала берется экстремальный — максимальный или минимальный полиинтервал. Возможны и другие варианты.

Пример. При поступлении на службу работник выбирает между фирмами *A*, *B* и *C*. Фирма *A* предлагает ему месячную зарплату (в зависимости от заказов фирмы) в размере $10\,000 \pm 2\,000$ руб. или $15\,000 \pm 2\,000$ руб., аналогично фирма *B* — в размере $5\,000 \pm 1\,000$ руб. или $8\,000 \pm 1\,000$ руб., а фирма *C* — в размере $6\,000 \pm 1\,000$ руб. или $9\,000 \pm 1\,000$ руб. Работнику нужно выбрать фирму с максимальной зарплатой.

Решение. *Шаг 1.* В фирме *A* первую заработную плату работника можно представить в виде интервала $[a_1^1, a_2^1] = [8\,000, 12\,000]$, 2-ю — как интервал $[a_1^2, a_2^2] = [13\,000, 17\,000]$. Аналогично, в фирме *B* 1-ю зарплату работника можно представить в виде интервала $[b_1^1, b_2^1] = [4\,000, 6\,000]$, а 2-ю — в виде интервала $[b_1^2, b_2^2] = [7\,000, 9\,000]$, а в фирме *C* 1-ю зарплату можно представить в виде интервала $[c_1^1, c_2^1] = [5\,000, 8\,000]$, а 2-ю — в виде интервала $[c_1^2, c_2^2] = [8\,000, 10\,000]$. Итак, месячную зарплату работника в фирмах *A*, *B* и *C* можно представить соответственно следующими полиинтервалами:

$$\tilde{A} = \bigcup_{i=1}^2 [a_1^i, a_2^i] = [8\,000, 12\,000] \cup [13\,000, 17\,000];$$

$$\tilde{B} = \bigcup_{j=1}^2 [b_1^j, b_2^j] = [4\,000, 6\,000] \cup [7\,000, 9\,000];$$

$$\tilde{C} = \bigcup_{i=1}^2 [c_1^i, c_2^i] = [5\,000, 8\,000] \cup [8\,000, 10\,000].$$

Абстрактная математическая модель решения задачи представляет собой совокупность полиин-

тервалов \tilde{A} , \tilde{B} и \tilde{C} , попарное сравнение которых должно на следующих шагах выделить максимальный полиинтервал, являющийся решением задачи.

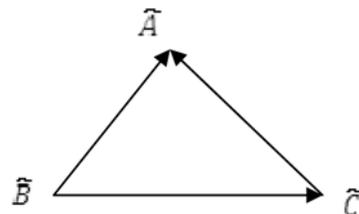
Шаг 2. Попарно сравниваем полиинтервалы \tilde{A} , \tilde{B} , \tilde{C} с помощью условий (16)–(18) и теоремы.

Пара (\tilde{A}, \tilde{B}) . Сравним полиинтервалы \tilde{A} , \tilde{B} с помощью теоремы, находим $a^1 = [8\,000, 12\,000]$, $b^2 = [7\,000, 9\,000]$, здесь $8\,000 > 7\,000$, $12\,000 > 9\,000$, и поэтому, согласно (14), $a^1 \geq b^2$, откуда по теореме $\tilde{A} \geq \tilde{B}$.

Пара (\tilde{A}, \tilde{C}) . С помощью тех же условий находим $a^1 = [8\,000, 12\,000]$, $c^2 = [8\,000, 10\,000]$, здесь $8\,000 \geq 8\,000$, $12\,000 > 10\,000$ и потому согласно (14), $a^1 \geq c^2$, откуда по теореме $\tilde{A} \geq \tilde{C}$.

Пара (\tilde{B}, \tilde{C}) . С помощью тех же условий находим $b^1 = [4\,000, 6\,000]$, $c^2 = [8\,000, 10\,000]$, здесь $4\,000 < 8\,000$, $6\,000 < 10\,000$ и потому $b^1 \not\geq c^2$, откуда по теореме 1 $\tilde{B} \not\geq \tilde{C}$. Аналогично имеем $c^1 = [5\,000, 8\,000]$, $b^2 = [7\,000, 9\,000]$, где $5\,000 < 7\,000$, $8\,000 < 9\,000$, поэтому $c^1 \not\geq b^2$, откуда по теореме $\tilde{C} \not\geq \tilde{B}$. Вместе с тем, согласно условию (16), $\tilde{B} \neq \tilde{C}$. Таким образом, в соответствии с (18) полиинтервалы \tilde{B} , \tilde{C} не сравнимы.

Шаг 3. По результатам шагов 1, 2 получаем структурно-математическую модель решения задачи в виде частично ориентированного графа (см. рисунок). Вершины этого графа — полиинтервалы \tilde{A} , \tilde{B} , \tilde{C} , его дуги ориентированы в направлении от меньших вершин \tilde{B} , \tilde{C} к большей вершине \tilde{A} , а его ребро не ориентировано и соединяет несравнимые вершины (полиинтервалы) \tilde{B} , \tilde{C} .



Шаг 4. По структурной граф-модели (см. рисунок) находим вершину, соответствующую максимальному полиинтервалу. Непосредственно из рисунка видно, что этой вершиной является \tilde{A} , так как соответствующий полиинтервал \tilde{A} больше других полиинтервалов \tilde{B} и \tilde{C} . Обратим внимание, что оба полиинтервала, не являющиеся максимальными, — \tilde{B} и \tilde{C} — не сравнимы между собой, но это не повлияло на решение задачи. Это решение — выбор работником из трех предложивших ему работу фирм *A*, *B*, *C* фирмы *A* как предложившей максимальную зарплату.

Обсуждение

Как показано в статье, дальнейшее развитие известной операции сравнения интервалов неопределенности с выделением максимального и минимального интервалов [7] приводит к новой операции сравнения полиинтервалов с выделением максимального и минимального полиинтервалов.

Таким образом, и для такой более сложной по сравнению с полиинтервалом модели неопределенности, как полиинтервал, оказывается возможным сравнивать полиинтервалы и выбирать максимальный (минимальный) из них. Это открывает возможность решения задач оптимизации для неопределенных систем и процессов с полиинтервальными параметрами. Полиинтервальная модель неопределенности является более сложной, чем интервальная. Она встречается достаточно часто в военном деле, экономике, технике, социальной сфере и других областях и поэтому заслуживает изучения и разработки. Это изучение и разработку применительно к проблеме оптимизации естественно осуществлять, используя подходы интервальной математики к оптимизации [7] и развивая их в направлении учета многоинтервальности. И здесь выявляется важный факт: сравнение двух полиинтервалов, как показывает теорема, сводится к сравнению их крайних интервалов, благодаря этому сложность решения проблемы оптимизации систем с полиинтервальными параметрами такая же, как и для систем с интервальными параметрами. Это делает задачи оптимизации систем с полиинтервальными параметрами реально разрешимыми.

Заклучение

В настоящей статье сформулирована задача дальнейшего изучения новой модели неопределенности — полиинтервала, обобщающей известную модель неопределенности — интервал — на случай области из нескольких последовательных интервалов неопределенности. С помощью известной из

интервальной математики теоретико-множественной конструкции, вводящей операции над интервалами, в том числе, операции взятия максимума (минимума), аналогичным путем введена операция взятия максимума (минимума) двух полиинтервалов. Разработана методика сведения операции взятия максимума и минимума полиинтервалов к аналогичным операциям с интервалами. С ее помощью доказана основная теорема, сводящая сравнение двух полиинтервалов, с выделением максимального и минимального из них, к сравнению двух интервалов — крайнего левого интервала одного полиинтервала и крайнего правого интервала другого. На примере из области экономики показана практическая польза разработанной теории и методов.

Список литературы

1. **Заде Л. А.** Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 176 с.
2. **Левин В. И.** Непрерывная логика. Пенза: Изд. ПГТА, 2008. 496 с.
3. **Горбань И. И.** Феномен статистической устойчивости. Киев: Наукова Думка, 2014. 370 с.
4. **Алефельд Г., Херцбергер Ю.** Введение в интервальные вычисления. М.: Мир, 1987. 370 с.
5. **Левин В. И.** Интервальная математика и исследование систем в условиях неопределенности. Пенза: Изд-во Пензенского технологического ин-та, 1998. 55 с.
6. **Левин В. И.** Полиинтервалы, их исчисление и приложение // Системы управления, связи и безопасности. 2016. № 3. С. 239—246.
7. **Левин В. И.** Методы оптимизации систем в условиях интервальной неопределенности параметров // Информационные технологии. 2012. № 4. С. 52—59.
8. **Вощинин А. П., Сотиров Г. Р.** Оптимизация в условиях неопределенности. М.: МЭИ, София: Техника, 1989. 226 с.

V. I. Levin, D. Sc., Professor, e-mail: vilevin@mail.ru,
Penza State Technological University, Penza, 440039, Russian Federation

Maximal and Minimal Polyintervals and Optimization in Condition of Uncertainty

The article introduced the concept of comparison of polyintervals, in order to separate larger and smaller polyinterval. The new concept is based on the adopted in the interval mathematics methodology to expand numeric operation to intervals. The main theorem which determines the necessary and sufficient conditions for comparability of polyintervals is proved. An example of using the results of the article to make better decisions in the economy is given.

Keywords: polyintervals, comparison of polyintervals, maximal (minimal) polyinterval

References

1. **Zade L. A.** Ponyatie lingvisticheskoi peremennoi i ego primenenie k prinyatiyu priblizhennykh reshenii, Moscow, Mir, 1976, 176 p. (in Russian).
2. **Levin V. I.** Nopreryvnaya logika, Penza, Publishing house of PGTA, 2008, 496 p. (in Russian).
3. **Gorban' I. I.** Fenomen statisticheskoi ustoichivosti, Kiev, Naukova Dumka, 2014, 370 p. (in Russian).
4. **Alefel'd G., Khertsberger Yu.** Vvedenie v interval'nye vychisleniya, Moscow, Mir, 1987, 370 p. (in Russian).
5. **Levin V. I.** Interval'naya matematika i issledovanie sistem v usloviyakh neopredelennosti, Penza, Publishing house of Penzenskoi tekhnologicheskii iniversity, 1998, 55 p. (in Russian).
6. **Levin V. I.** Poliinterval'y, ikh ischislenie i prilozhenie, *Sistemy Upravleniya, Svyazi i Bezopasnosti*, 2016, no. 3, pp. 239—246 (in Russian).
7. **Levin V. I.** Metody optimizatsii sistem v usloviyakh interval'noi neopredelennosti parametrov, *Informatsionnye Tekhnologii*, 2012, no. 4, pp. 52—59 (in Russian).
8. **Voshchinin A. P., Sotirov G. R.** Optimizatsiya v usloviyakh neopredelennosti, Moscow, Publishing house of MEI, Sofiya, Tekhnika, 1989, 226 p. (in Russian).

А. Ю. Романов, канд. техн. наук, ст. преп., e-mail: a.romanov@hse.ru,
 Национальный исследовательский университет "Высшая школа экономики",
К. Е. Ломотин, студент, e-mail: ke.lomotin@gmail.com,
Е. С. Козлова, студентка, e-mail: hse.kozlovaes@gmail.com

Применение методов машинного обучения для решения задачи автоматической рубрикации статей по УДК

Выполнено исследование применимости современных методов машинного обучения к задаче автоматической генерации кодов УДК научных статей. В качестве классификаторов рассматриваются такие модели, как искусственные нейронные сети, логистическая регрессия и бустинг. Разработаны графовые алгоритмы и прототип программного модуля для генерации кода УДК.

Ключевые слова: классификация текстов, машинное обучение, искусственная нейронная сеть, бустинг, логистическая регрессия, УДК

Введение

На современном этапе задача обработки текстов на естественных языках является одной из насущных. Она напрямую связана с растущим количеством текстовой информации в сети Интернет. Научные статьи, книги, руководства, журналы, справочники — все это многообразие текстов в большинстве своем представлено в цифровом виде. Вместе с тем возникает проблема ориентации в таком большом потоке информации, поскольку данные в основном не упорядочены, и поиск полной информации по конкретному вопросу становится все более трудоемким. В этой связи использование автоматических классификаторов является одним из перспективных способов решения данной проблемы [1].

Задачи классификации текстов возникают в различных областях: например, при обработке электронной корреспонденции (фильтры спама); поиске информации; контекстной рекламе; в системах документооборота; при создании тематических каталогов; при наполнении материалами тематических Интернет-порталов и т. д. [2, 3]. Для решения этих задач применяют различные варианты классификаторов, такие как: метрические (SVM, KNN [4]); нейросетевые (различные перцептроны, сверточные нейронные сети [5], ART, SOM [6]); вероятностные (байесовский классификатор [7]); основанные на решающих правилах (решающие деревья, случайный лес [8]) и др. Однако вопрос автоматической классификации научных статей по УДК с помощью машинного обучения в работах по данной проблематике практически не рассматривался. Следовательно, цель данного исследования — разработать систему, которая сможет с высокой точностью генерировать коды УДК для научных статей, основываясь на принципах машинного обучения.

Особенностью большинства методов классификации текстов является то, что алгоритм настройки параметров модели (обучение) требует обучающей

выборки (некоторого набора данных), а также ответа эксперта в случае обучения с учителем. В данном случае в качестве обучающего набора данных выступают тексты научных статей, а в качестве ответа эксперта — код УДК, присвоенный статье автором или модераторами ресурса. Таким образом, первым этапом исследования является сбор данных (выполнен в ходе предыдущего исследования [9]), в результате которого была составлена обучающая выборка статей.

После преобработки данных необходимо подобрать гиперпараметры моделей. В рассматриваемой задаче — это количество нейронов в скрытом слое нейронной сети или глубина решающего дерева. После прохождения данного этапа модели готовы к обучению и тестированию, на основе результатов которого проводится анализ и делаются выводы о применимости той или иной модели к решаемой задаче.

Обработка кодов УДК

УДК, или универсальная десятичная классификация [10] — это система классификации информации, которая широко используется во всем мире для систематизации произведений науки, литературы и искусства, периодической печати, различных видов документов и организации картотек. Центральной частью УДК являются основные таблицы, охватывающие всю совокупность знаний и построенные по иерархическому принципу деления от общего к частному с использованием цифрового десятичного кода.

Основные разделы УДК по ГОСТ 7.90—2007 [11]:
 0. Общий отдел. Наука и знание. Информация. Документация. Библиотечное дело. Организации. Публикации в целом.

1. Философия. Психология.
2. Религия. Богословие.
3. Общественные науки.

4. (Резерв для будущего применения.) Свободен с 1961 года. Содержание перенесено в раздел 8.
5. Математика. Естественные науки.
6. Прикладные науки. Медицина. Технология.
7. Искусство. Развлечения. Зрелища. Спорт.
8. Язык. Языкознание. Лингвистика. Литература.
9. География. Биографии. История.

Код УДК представляет собой последовательность цифр, разделенных для удобства точкой через каждые три разряда. Каждой подтеме соответствует свой разряд и некоторый определитель (цифра, цифра с ".0", цифра с апострофом, цифра с дефисом); возможна запись диапазона тем с помощью разделителя "./". Для кода УДК определены операции отношения одного кода к другому (символ ":"), двойного отношения ("::") и соединения ("+"). В общем случае могут использоваться определители языка, этносов и народностей, места (с возможностью локализации в пределах административной единицы), формы документа, времени, климатических зон и районов. Также имеются так называемые "общие определители", которые начинаются с символов "-0" и могут означать свойства, материалы или личные характеристики [12].

Генерировать код такой сложной структуры, имея лишь текст научной статьи, — это задача, с которой зачастую с трудом может справиться даже подготовленный человек. Для автоматических систем она все еще остается невыполнимой, несмотря на эксперименты и разработки в области обработки текстов на естественных языках [13]. Поэтому для работы с УДК в контексте данного исследования коды УДК для текстов из обучающей выборки были приведены к более простому виду по следующим правилам:

1. Удаляются все специальные текстовые определители (места, времени и т. д.).

2. Удаляются квадратные скобки, которые применяются для обозначения разнохарактерных связей в индексе с двумя и более знаками отношения, когда необходимо показать, что один из знаков отношения относится к совокупности других индексов, соединенных этим знаком и являющихся единым понятием, например, 621.311.25:[621.362:531.84] — электростанции с магнитогидродинамическим генератором.

3. Принимается допущение, что части кода, связанные символом отношения или двойного отношения, равнозначны и могут быть записаны как отдельные более широкие по значению коды через символ ";". Это возможно в силу одного из правил образования составных индексов: на первом месте в отношении ставится раздел, который является основным для документа. Зачастую основная тема может быть сформулирована достаточно размыто, чтобы части отношения были связаны слабо.

4. Удаляются правые операнды операции соединения. В ходе предыдущих исследований [14] было замечено, что в записи этой операции совершается

большое число ошибок, таких как соединение кода с диапазоном или общим определителем из списка: -02, -03 или -05. УДК в рассматриваемой задаче является экспертной меткой класса, поэтому ошибки в коде ведут к росту ошибки классификации.

Обучение моделей классификаторов

В таблице приведены типы классификаторов и их метрики качества, которые были выбраны нами для реализации поставленной задачи. Каждая из моделей имеет свои гиперпараметры, подбор которых подразумевается по умолчанию и часто не упоминается при описании модели, поскольку не представляет научной новизны. Рассмотрим задействованные модели классификаторов более подробно.

Искусственная нейронная сеть. Несмотря на то, что формальный нейрон Маккалока-Питтса [15] реализует, по сути, логистическую регрессию, композиция таких нейронов составляет совершенно новую модель, в некоторых аспектах подражающую работе мозга. Нейронные сети прямого пространства в последнее время обрели большую популярность и их успешно применяют для классификации различных объектов, в том числе и текстов [5, 16]. В данной работе мы рассматривали нейронную сеть с несколькими слоями нейронов, где есть входной слой, выходной, а также один или несколько скрытых [9]. Такая архитектура называется многослойным перцептроном.

Практика показывает, что для решения большинства задач достаточно одного скрытого слоя. Однако исследования многослойных структур свидетельствуют о том, что с их помощью можно находить неявные зависимости, разделять более сложные кластеры объектов и т. д. [6, 17]. Поэтому в работе рассмотрены также и двухслойные перцептроны. Большее число слоев, как правило, используется для решения узкого круга специфических задач [5, 18].

В таблице приведены значения метрик качества для нейросети с одним и двумя слоями. Очевидно, что различия между ними практически нет. Не-

Результаты работы классификаторов

| Классификатор | Тип термов | Accuracy score | Precision score | Recall score | F1 score |
|--|---------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Искусственная нейронная сеть (один скрытый слой) | Слова Биграммы | 0,6892 0,7035 | 0,6965 0,7109 | 0,7039 0,7182 | 0,6990 0,7133 |
| Искусственная нейронная сеть (два скрытых слоя) | Слова Биграммы | 0,6954 0,6815 | 0,6991 0,7031 | 0,7027 0,7247 | 0,7003 0,7103 |
| Деревья с использованием алгоритма AdaBoost | Слова Биграммы | 0,4771 0,4718 | 0,4955 0,4896 | 0,4771 0,4718 | 0,4821 0,4767 |
| Логистическая регрессия | Слова Биграммы | 0,7650 0,7098 | 0,7633 0,7093 | 0,7650 0,7098 | 0,7629 0,7022 |

большие отличия могут быть вызваны особенностями данных, которые далеки от идеальных. Реальные тексты могут содержать псевдоключевые слова и другие информационные "выбросы". Графически это явление можно представить как возмущения на поверхности метрики в пространстве гиперпараметров модели. Например, на рис. 1 (см. вторую сторону обложки) изображена поверхность, отражающая зависимость метрики точности распознавания для перцептрона с двумя скрытыми слоями от числа нейронов в этих слоях.

Точность распознавания возрастает примерно до тех пор, пока число нейронов в обоих слоях не начинает соответствовать числу классов (9). Результаты тестирования нейросетей с достаточным числом нейронов группируются вокруг плоскости на уровне точности распознавания, достигающем 0,65. Вследствие того, что для построения этой поверхности использована нейросеть со второй по эффективности функцией активации — гиперболическим тангенсом, — эта плоскость оказалась примерно на 0,5 ниже, чем значения, приведенные в таблице. В остальном же рельеф соответствует оптимальным гиперпараметрам.

Анализ подобных поверхностей может являться обоснованием для автоматического подбора гиперпараметров при построении системы классификаторов.

Все нейронные сети в ходе экспериментов обучались по градиентному методу "Adam" с адаптивной скоростью обучения, разработанному командой ученых из Амстердама (Нидерланды) и Торонто (Канада) [19].

Адаптивный бустинг. Одним из самых популярных ансамблевых методов является алгоритм адаптивного бустинга AdaBoost [20]. Заключается он в том, что очень слабые классификаторы — решающие пни (решающее дерево с глубиной 1) или деревья с маленькой глубиной — последовательно учатся на взаимных ошибках, после чего их ответы объединяются определенным образом. В результате получается весьма хорошо настраиваемая модель, применимая только на деревьях с очень маленькой глубиной, поскольку она склонна к переобучению [21]. В какой-то момент результаты должны стабилизироваться, что и происходит у классификаторов, использующих слова (рис. 2). Для классификаторов, использующих биграммы, стабилизация не возникает на этом промежутке, что является признаком более сложных нелинейных зависимостей результата от признаков.

Логистическая регрессия. Следующим используемым классификатором стала логистическая регрессия, представляющая собой линейную регрессию, результаты которой являются аргументами логистической функции, что позволяет определять вероятность принадлежности текста к определенному классу [22]. Таким образом, получен достаточно эффективный классификатор, результаты тести-

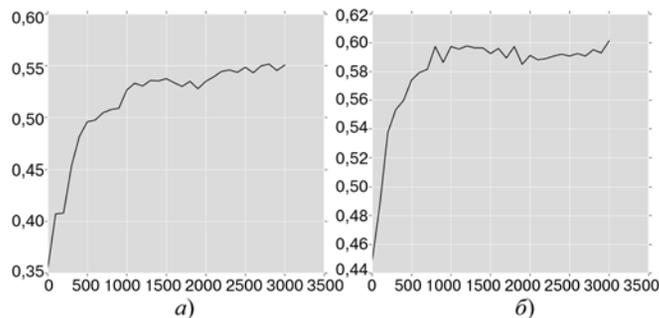


Рис. 2. Зависимость результата классификации от числа последовательных классификаторов в алгоритме AdaBoost: а — для биграмм; б — для слов

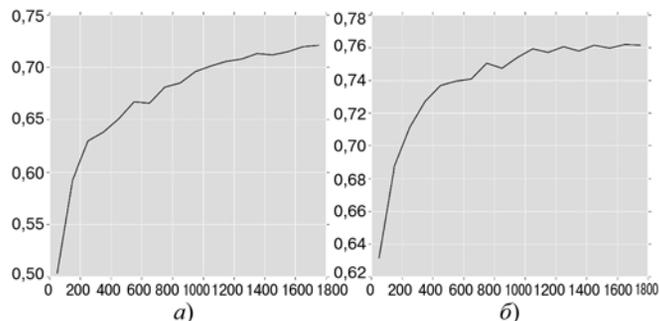


Рис. 3. Зависимость результатов работы логистической регрессии от числа термов: а — для биграмм; б — для слов

рования которого приведены на рис. 3. Отдельно стоит отметить, что этот классификатор показал самые высокие результаты по всем используемым метрикам, что следует из приведенной таблицы.

Анализ результатов тестирования классификаторов. Следует отметить, что лучшие результаты дала логистическая регрессия для слов; за ней с небольшим отставанием идут логистическая регрессия для биграмм и нейронные сети. Алгоритм AdaBoost, построенный на решающих деревьях, показал худший результат.

В большинстве случаев при равном числе термов биграммы показывают худшие результаты, чем слова, практически по всем метрикам качества. Тем не менее рост качества в зависимости от числа термов у них продолжается дольше, это позволяет сделать вывод о том, что потенциал их применения на большем объеме данных у них выше.

Можно заметить, что качество лучших моделей приближается к отметке 0,7. Вероятно, это вызвано ошибками в экспертных метках класса и несовершенством метода отбора ключевых термов. Отсюда следует вывод о том, это наравне с гиперпараметрами модели следует улучшать качество данных и вспомогательных алгоритмов.

Генерация кода УДК

В ходе исследования было разработано два подхода к генерации кода УДК. Первый подход заключается в том, что переход на следующий уровень

УДК основывается на ответе классификатора на предыдущем уровне. Например, классификатор отнес текст к теме "б" на первом уровне. Тогда последующие классификаторы будут уточнять ответ, и классификация будет проводиться лишь в пределах темы "б". Если классификатор дает ответ в виде оценки принадлежности текста к тому или иному классу (ответ в виде некоторого показателя принадлежности), то этот подход можно интерпретировать как рекурсивное нахождение самых дальних вершин в дереве по алгоритму, формулировка которого выглядит следующим образом: пока текущий узел не лист — найти вершину, к которой от текущей ведет ребро с самым большим показателем принадлежности текста к подтеме, инцидентной текущей, и принять ее за текущую вершину.

Достоинством такого подхода является высокая скорость сходимости. Это обусловлено тем, что глубина дерева конечна и невелика, а также тем, что над классификацией одного текста работает лишь малая часть всех классификаторов системы. Это качество очень важно для построения приложения, так как конечный пользователь (сетевой бот, база данных, в которую интегрирована система классификации, или человек) не всегда может ожидать ответа в течение длительного времени. В случае, когда классификаторы интегрированы в более сложную систему, задержка может нарушить ее работу. Если же система классификации работает на сервере, то сервер, вероятнее всего, разорвет соединение по тайм-ауту.

Главный недостаток алгоритма состоит в том, что он не гарантирует нахождение лучшего результата: не происходит обход всего дерева с присвоением веса ребрам, поэтому учитываются лишь веса ребер, инцидентных текущей вершине. Если у соседней ветви дерева показатели распознавания лучше, то они будут проигнорированы, в то время как из низких показателей нижних уровней будет выбран наибольший, хоть и не предельно оптимальный, в контексте всего дерева.

Второй подход состоит в построении ориентированного взвешенного дерева для всей структуры УДК, заложенной в систему при обучении. Системой выполняется оценка принадлежности текста всем классам и подклассам, показатели ставятся в соответствие ребрам, входящим в вершину, а затем выбирается лист, для пути к которому достигается максимум некоторой меры принадлежности (например, среднего значения весов ребер пути). Алгоритм, реализующий второй подход, выглядит следующим образом: для всех листов получить ответ классификатора для каждой рубрики на каждом уровне, просуммировать результаты оценки для всех путей по уровням и разделить на номер уровня, на котором находится лист данной ветви, и выбрать лист, на пути к которому среднее значение показателей максимально.

При этом нет необходимости применять алгоритмы поиска самого длинного пути в графе, так как при его построении все пути и их средний вес запоминаются.

К достоинствам алгоритма можно отнести то, что его результатом является взвешенный граф. Следовательно, алгоритм можно усовершенствовать, используя математический аппарат теории графов. Также он учитывает одну из особенностей УДК — листья разных ветвей могут быть очень близки по смыслу: например, код 398.541 ("Народные представления религиозного содержания. Мистерии. Райские представления. Представления, посвященные рождению Иисуса Христа. Представления о прорицании волхвов и т. д. Представления о "страстях Христовых"). Представления в д. Обераммергау") и 82-343.5 ("Легенды о святых. Жизнеописания святых в форме легенд. Жития святых"). В таком случае у ребер, инцидентных этим листьям, будут максимальные веса, даже если на верхних уровнях веса были невелики. Еще одно преимущество данного подхода — это возможность сортировки листов по среднему весу путей, ведущих к ним. Появляется возможность возвращать несколько наиболее подходящих кодов УДК, объединенных знаком "+".

Большим недостатком алгоритма является необходимость запуска всех классификаторов. Как было отмечено выше, их число может достигать нескольких сотен, а суммарное число классифицирующих алгоритмов на все дерево УДК исчисляется тысячами. Построение взвешенного графа может занимать значительное время, что не всегда соответствует требованиям, предъявляемым к программному обеспечению.

Различия двух предложенных подходов продемонстрированы на рис. 4 (см. вторую сторону обложки).

Числа, отмеченные красным, — это веса, рассчитанные по второму алгоритму; синим — по первому. Каждому листу соответствует средний вес пути от фиктивной вершины. Результатом алгоритма "быстрой" генерации является код 620.1, а результатом "медленной" генерации — код 001.

Одним из вариантов улучшения предложенных алгоритмов является введение эвристического критерия остановки спуска по дереву: например, остановить спуск, когда вероятность принадлежности текста к теме меньше 0,5.

Таким образом, можно сделать вывод о сложности оценки адекватности того или иного подхода, так как всегда существует большая вероятность того, что среди эталонной выборки текстов найдутся статьи, коды для которых могут быть правильно сгенерированы только "быстрым" алгоритмом или только "медленным" алгоритмом. Перспективное решение этой проблемы состоит в комбинации алгоритмов. Например, сначала алгоритм № 1 спускается на несколько уровней, выбирая наиболее подходящие ветви, а затем алгоритм № 2 осу-

шестьляет точный поиск правильных кодов среди подтем. При правильном выборе баланса уровней, на которых выполняет работу каждый из алгоритмов, система сможет достичь требуемой точности за сравнительно небольшое время. Выполнение предложенных алгоритмов может быть ускорено применением методов параллельных вычислений в силу того, что структура УДК соответствует ориентированному графу в виде дерева, и ветви этого графа независимы между собой. Поэтому расчет весов ребер для каждой из ветвей можно выполнять на отдельном вычислителе [23].

Заключение

Исследование основных моделей машинного обучения и результаты их тестирования позволяют прийти к выводу о том, что для построения системы автоматической классификации научных статей по УДК наиболее эффективными являются классификаторы на базе логистической регрессии (достигает до 0,76 показатель качества распознавания по всем метрикам).

В процессе исследования были разработаны графовые алгоритмы генерации кода УДК, комбинация которых отличается гибкостью и высоким потенциалом для улучшения. Одним из существенных результатов работы стал прототип программного модуля, работающий по принципу логистической регрессии и способный генерировать первую цифру кода УДК.

Следует отметить, что результаты данной работы являются необходимой базой для проведения дальнейших исследований, в ходе выполнения которых предполагается рассмотрение обширной проблематики, которая включает:

- проблемы баланса скорости и эффективности при использовании графовых алгоритмов генерации кода УДК;
- перспективные архитектуры нейронных сетей для рассматриваемой проблематики и возможности применения методов искусственного интеллекта к управлению построением системы классификаторов;
- возможности рекурсивного автоматического построения системы классификаторов для всего дерева УДК;
- возможности применения метода параллельных вычислений для ускорения алгоритмов классификации и генерации УДК;
- применение семантических моделей текста, а также проведение экспериментов по комбинированию их со статистическими методами;
- возможности генерации специальных определителей УДК (места, даты, этноса и т. д.);
- адаптация результатов исследований к другим задачам обработки текстов на естественных языках.

Итогом этих исследований станет создание программного комплекса, который сможет самостоятельно загружать текстовые документы требуемого

типа из сети Интернет, обрабатывать их, выбирать модель и ее гиперпараметры в целях построения системы классификаторов со структурой, способной наиболее точно описывать предметную область документа.

Список литературы

1. **Partalas I. et al.** LSHTC: A Benchmark for Large-Scale Text Classification // ArXiv preprint. 2015. ArXiv № 1503.08581v1.
2. **Sebastiani F.** Classification of text, automatic // The Encyclopedia of Language and Linguistics. Second Edition. Oxford: Elsevier, 2006. Vol. 2. P. 457–462.
3. **Булгаков М. В., Гридина Е. Г., Иванников А. Д., Старых В. А.** Федеральная система информационно-образовательных ресурсов // Информационные ресурсы России. 2009. № 2. С. 25–27.
4. **Gabrilovich E., Markovitch S.** Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5 // Proc. of the Twenty-first international conference on Machine learning. 2004. P. 41–48.
5. **Johnson R., Zhang T.** Effective Use of Word Order for Text Categorization with Convolutional Neural Networks // Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL. Denver: Association for Computational Linguistics, 2015. P. 103–112.
6. **Ruiz M. E., Srinivasan P.** Hierarchical text categorization using neural networks // Information Retrieval. 2002. Vol. 5, N. 1. P. 87–118.
7. **Епрев А. С.** Автоматическая классификация текстовых документов // Математические структуры и моделирование. 2010. № 1 (21). С. 65–82.
8. **Проноза Е. В., Ягунова Е. В.** Аспектный анализ отзывов о ресторанах для рекомендательных систем е-туризма // Сборник научных статей XVIII Объединенной конференции "Интернет и современное общество". IMS—2015, Санкт-Петербург, 2015. С. 23–25.
9. **Romanov A. Yu., Lomotin K. E., Kozlova E. S., Kolesnichenko A. L.** Research of Neural Networks Application Efficiency in Automatic Scientific Articles Classification According to UDC // IEEE 2016 International Siberian Conference on Control and Communications (SIBCON): Proceedings. Moscow: NRU HSE, 2016. Article Number: 543fu4t.
10. **ГОСТ 7.74-96—1997.** Система стандартов по информации, библиотечному и издательскому делу. Информационно-поисковые языки. Термины и определения. М.: Издательство стандартов, 1997. 38 с.
11. **ГОСТ 7.90—2007.** Система стандартов по информации, библиотечному и издательскому делу. Универсальная десятичная классификация. Структура, правила введения и индексирования. М.: Стандартинформ, 2010. 28 с.
12. **Общая методика применения универсальной десятичной классификации.** URL: <http://systemling.narod.ru/udc/Short-Guide.htm> (дата обращения: 25.12.2016).
13. **Заболеева-Зотова А. В., Орлова Ю. А.** Автоматизация семантического анализа документации технического задания // Вестник компьютерных и информационных технологий. 2008. № 9. С. 26–34.
14. **Ломотин К. Е., Козлова Е. С., Колесниченко А. Л., Романов А. Ю.** Применение современных методов классификации к рубрикации научных статей по УДК // Инновационные, информационные и коммуникационные технологии: сборник трудов XIII Международной научно-практической конференции. М.: Ассоциация выпускников и сотрудников ВВИА им. проф. Жуковского, 2016. С. 92–95.
15. **Терехин А. Т.** Нейросетевое моделирование когнитивных функций мозга: обзор основных идей // Психологические исследования. 2009. № 2 (4). URL: <http://psystudy.ru> (дата обращения: 25.12.2016).
16. **Мешкова Е. В.** Методика построения классификатора текста на основе гибридной нейросетевой модели // Известия Южного федерального университета. Технические науки. 2008. Т. 81, № 4. С. 212–215.
17. **Gardner M. W., Dorling S. R.** Artificial neural networks (the multilayer perceptron) — a review of applications in the atmospheric sciences // Atmospheric environment. 1998. Vol. 32, N. 14. P. 2627–2636.

18. **LeCun Y.** Gradient-based learning applied to document recognition // *Proceedings of the IEEE*. 1998. Vol. 86. N. 11. P. 2278–2324.
 19. **Kingma D., Ba J.** Adam: A method for stochastic optimization // *ICLR 2015, arXiv preprint*. San Diego, 2015. ArXiv № 1412.6980v2.
 20. **Rätsch G., Onoda T.** Soft margins for AdaBoost // *Machine learning*. 2001. Vol. 42. N. 3. P. 287–320.
 21. **Донской В. И.** Алгоритмы обучения, основанные на построении решающих деревьев // *Журнал вычислительной математики и математической физики*. 1982. Т. 22. № 4. С. 963–974.

22. **Жилина Е. В.** Использование бинарной логистической регрессии для оценки качества адаптивного теста // *Вестник Томского государственного университета*. Томск: НИ ТГУ, 2010. № 334. С. 106–109.
 23. **Пескишева Т. А., Котельников Е. В.** Параллельная реализация алгоритма обучения системы текстовой классификации // *Вестник Уфимского государственного авиационного технического университета*. 2011. Т. 15, № 5 (45). С. 130–136.

A. Yu. Romanov, PhD, Senior Lecturer, e-mail: a.romanov@hse.ru,
 National Research University Higher School of Economics,
K. E. Lomotin, Student, e-mail: ke.lomotin@gmail.com,
E. S. Kozlova, Student, e-mail: hse.kozlovaes@gmail.com

Machine Learning Methods Application to Automatic Scientific Articles Classification According to UDC

The paper deals with the applicability of modern machine learning methods to the problem of automatic generation of UDC for scientific articles. As the classifiers, such models as artificial neural networks, logistic regression and boosting are considered. Graph algorithms and a prototype software module to generate UDC are designed.

Keywords: text classification, machine learning, artificial neural network, logistic regression, boosting, UDC

References

1. **Partalas I., et al.** LSHTC: A Benchmark for Large-Scale Text Classification, *arXiv preprint*, 2015, arXiv no.: 1503.08581v1.
 2. **Sebastiani F.** Classification of text, automatic, *The Encyclopedia of Language and Linguistics. Second Edition*, Oxford: Elsevier, 2006, vol. 2, pp. 457–462.
 3. **Bulgakov M. V., Gridina E. G., Ivannikov A. D., Staryh V. A.** Federal'naja sistema informacionno-obrazovatel'nyh resursov (The federal system of informational and educational resources), *Informacionnye resursy Rossii*, 2009, no. 2, pp. 25–27.
 4. **Gabrilovich E., Markovitch S.** Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5, *Proceedings of the twenty-first international conference on Machine learning*, 2004, pp. 41–48.
 5. **Johnson R., Zhang T.** Effective Use of Word Order for Text Categorization with Convolutional Neural Networks, *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, Denver: Association for Computational Linguistics, 2015, pp. 103–112.
 6. **Ruiz M. E., Srinivasan P.** Hierarchical text categorization using neural networks, *Information Retrieval*, 2002, vol. 5, no. 1, pp. 87–118.
 7. **Eprez A. S.** Avtomaticheskaja klassifikacija tekstovyh dokumentov (Automatic classification of text documents), *Matematicheskie struktury i modelirovanie*, 2010, no. 1 (21), pp. 65–82.
 8. **Pronoza E. V., Jagunova E. V.** Aspektnyj analiz otzyvov o restoranah dlja rekomendatel'nyh sistem e-turizma (Aspect analysis of restaurant reviews for e-tourism recommendation systems), *Sbornik nauchnyh statej XVIII Ob'edinennoj konferencii Internet i sovremennoe obshhestvo (IMS-2015)*, Sankt-Peterburg, 2015, pp. 23–25.
 9. **Romanov A. Yu., Lomotin K. E., Kozlova E. S., Kolesnichenko A. L.** Research of Neural Networks Application Efficiency in Automatic Scientific Articles Classification According to UDC, *IEEE 2016 International Siberian Conference on Control and Communications (SIBCON). Proceedings*, Moscow: NRU HSE, 2016, article number: 543fu4t.
 10. **GOST 7.74-96—1997.** Sistema standartov po informacii, biblioteknomu i izdatel'skomu delu. Informacionno-poiskovye jazyki. Terminy i opredelenija (System of standards on information, librarianship and publishing. Information retrieval languages. Terms and Definitions), M.: IPK Izdatel'stvo standartov, 1997, 38 p.
 11. **GOST 7.90—2007.** Sistema standartov po informacii, biblioteknomu i izdatel'skomu delu. Universal'naja desjaticnaja klassifikacija. Struktura, pravila vvedenija i indeksirovanija (System of standards on information, librarianship and publishing. Universal Decimal Classification. Structure, rules of administration and indexing), Moscow, Standartinform, 2010, 28 p.

12. **Obshhaja metodika primeneniya universal'noj desjaticnojj klassifikacii** (General procedure of application of Universal Decimal Classification), available at: <http://systemling.narod.ru/udc/Short-Guide.htm> (date of access: 25.12.2016).
 13. **Zaboleeva-Zotova A. V., Orlova Ju. A.** Avtomatizacija procedur semanticheskogo analiza teksta tehničeskogo zadanija (Automation of semantic analysis of technical specifications), *Izvestija Volgogradskogo gosudarstvennogo tehničeskogo universiteta*, 2007, vol. 9, no. 3, pp. 52–55.
 14. **Lomotin K. E., Kozlova E. S., Kolesnichenko A. L., Romanov A. Yu.** Primenenie sovremennyh metodov klassifikacii k rubrikacii nauchnyh statej po UDK (Application of modern classification methods to scientific articles categorization by UDC), *Innovacionnye, informacionnye i kommunikacionnye tehnologii: sbornik trudov XIII Mezhdunarodnoj nauchno-praktičeskoj konferencii. Moscow, Associacija vypusnikov i sotrudnikov VVIA im. prof. Zhukovskogo*, 2016, pp. 92–95.
 15. **Terehin A. T.** Nejrosetevoe modelirovanie kognitivnyh funkcij mozga: obzor osnovnyh idej (Neural network modeling of cognitive brain functions: an overview of the basic ideas), *Psihologičeskie issledovanija*, 2009, no. 2 (4), available at: <http://psystudy.ru> (date of access: 25.12.2016).
 16. **Meshkova E. V.** Metodika postroenija klassifikatora teksta na osnove gibridnoj nejrosetevoj modeli (Technique of text classifier construction based on of the hybrid neural network model), *Izvestija Juzhnogo federal'nogo universiteta. Tehničeskie nauki*, 2008, vol. 81, no. 4, pp. 212–215.
 17. **Gardner M. W., Dorling S. R.** Artificial neural networks (the multilayer perceptron) — a review of applications in the atmospheric sciences, *Atmospheric environment*, 1998, vol. 32, no. 14, pp. 2627–2636.
 18. **LeCun Y.** Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 1998, vol. 86, no. 11, pp. 2278–2324.
 19. **Kingma D., Ba J.** Adam: A method for stochastic optimization, *ICLR 2015, arXiv preprint*, San Diego, 2015, arXiv no.: 1412.6980v2.
 20. **Rätsch G., Onoda T.** Soft margins for AdaBoost, *Machine learning*, 2001, vol. 42, no. 3, pp. 287–320.
 21. **Donskoj V. I.** Algoritmy obuchenija, osnovannye na postroenii reshajushhijh derev'ev (Machine learning algorithms, based on the construction of decision trees), *Zhurnal vychislitel'noj matematiki i matematičeskoj fiziki*, 1982, vol. 22, no. 4, pp. 963–974.
 22. **Zhilina E. V.** Ispol'zovanie binarnoj logičeskojj regreessii dlja ocenki kachestva adaptivnogo testa (Application of binary logistic regression to the quality estimation of the adaptive test), *Vestnik Tomskogo gosudarstvennogo universiteta*, Tomsk, NI TGU, 2010, no. 334, pp. 106–109.
 23. **Peskisheva T. A., Kotel'nikov E. V.** Parallel'naja realizacija algoritma obuchenija sistemy tekstovoj klassifikacii (Parallel implementation of learning algorithm of text classification system), *Vestnik Ufimskogo gosudarstvennogo aviacionnogo tehničeskogo universiteta*, Ufa: UGATU, 2011, vol. 15, no. 5 (45), pp. 130–136.

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 629.7

Э. Э. Акимкина, аспирант, elakimkina@gmail.com

Московский государственный областной технологический университет, г. Королев

Инструментальный подход к организации сбора данных в хранилище систем поддержки принятия решений

Проанализированы проблемы современных средств интеграции данных в хранилище систем поддержки принятия решений; разработана шестиуровневая архитектура, которая отличается от широко применяемой трехуровневой архитектуры тем, что введенные уровни учитывают проблемы извлечения данных с помощью новейших информационных технологий от различных поставщиков информационных ресурсов и позволяют детализировать операции обработки данных внутри каждого уровня. Определены аналитические зависимости для различных методов и сценариев извлечения, сбора и загрузки данных; сформулированы требования к программному функционалу, реализующему шестиуровневую сервис-ориентированную архитектуру хранилища данных.

Ключевые слова: интеграция данных, методы сбора данных, уровни детальности, сервис-ориентированные решения

Введение

На современном этапе развития систем поддержки принятия решений (СППР) использование хранилищ данных (ХД) позволяет обеспечить быстрый доступ к редко используемым или ненормализованным данным и осуществить анализ больших объемов данных. Проблемы современных средств интеграции данных в хранилище СППР заключаются в том, что эти средства недостаточно поддерживают процесс извлечения данных, не учитываются особенности взаимодействия с поставщиками информационных ресурсов и целевые концепции хранения; не согласованы уровни детальности. Необходимо однозначно определенные методы сбора данных и согласованности уровней детальности для унификации обработки в ХД электронных источников различных форматов.

Описание сервис-ориентированной архитектуры ХД

Хранилище данных (ХД) — предметно-ориентированная информационная база данных (БД), специально разработанная и предназначенная для подготовки отчетов и бизнес-анализа в целях поддержки принятия решений в организации. Строится на базе систем управления и СППР.

Задачи интеллектуального анализа интегрированных данных различаются по уровню сложности. Обобщенная концептуальная схема ХД, представленная на рис. 1 [1], имеет трехуровневую архитектуру (источники данных — хранилище данных —

отчеты) [2] и иллюстрирует зависимость архитектуры ХД от доступности и структуры основной БД. На третьем уровне (визуализация, отчеты) могут также располагаться предметно-ориентированные витрины данных (ВД). СППР может иметь независимые витрины данных, не входящие в архитектуру ХД [3].

В целях повышения оперативности обработки данных допускаются ненормализованные отношения, трехуровневая архитектура ХД представляется гиперкубом данных [4].

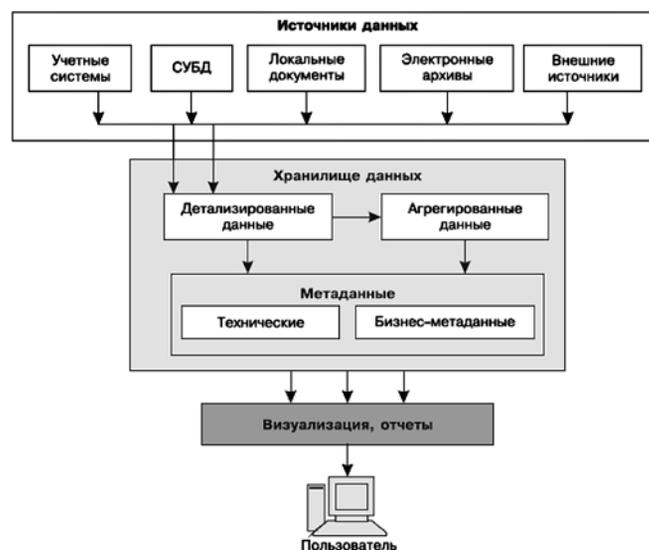


Рис. 1. Типовая конфигурация ХД

Широко распространенные трехуровневые архитектуры ХД и гиперкубы, основанные на трехуровневой архитектуре, не могут описывать задачи современных ХД, характеризующихся большим объемом обрабатываемых данных разных форматов (до миллиона запросов в день). Архитектура ХД должна отражать средства: для интеграции данных с другими источниками; для повышения производительности; трансформации данных; для фиксации истории получения данных; для очистки данных. Широко используемые при обработке данных XML-преобразования, Web-сервисы, спецификации JDBC и JMS определяют архитектуру ХД как сервисно-ориентированную [5].

Для отображения новейших сервисов интеграции, хранения, доставки, предоставления данных автором разработана сервис-ориентированная шестиуровневая архитектура ХД, структура которой представлена на рис. 2.

Передача данных из первого уровня во второй уровень осуществляется на основе подхода ETL (от англ. *extract, transformation, load* — извлечение, трансформация, загрузка), включающего этапы. На втором уровне находятся двумерные ASCII-файлы, файлы во внутреннем формате системы или в виде временных промежуточных таблиц в БД (снимки или реплицированные из других источников таблицы). Данные, находящиеся на втором уровне, для анализа не доступны. На втором уровне может

| | |
|--|-------------------------------|
| Первый уровень «Источники данных» | Документы MS Office |
| | Унаследованные системы |
| | Транзакционные системы |
| | Учетные системы |
| | Файлы |
| | Архивы |
| | Системы управления БД |
| Второй уровень «Подготовка данных для хранения» | Извлечение и сбор данных |
| | Трансформация данных |
| | Загрузка данных |
| Третий уровень «Хранение данных» | Ведение метаданных |
| | Центральное хранилище данных |
| | Оперативный склад данных |
| | Зоны временного хранения |
| | Режимы хранения |
| Четвертый уровень «Обработка данных» | Выборка |
| | Реструктуризация |
| | Доставка |
| Пятый уровень «Предоставление данных» | Тематическая витрина данных |
| | Региональная витрина данных |
| | Витрина данных подразделения |
| | Прикладная витрина данных |
| | Функциональная витрина данных |
| Шестой уровень «Приложения данных» | Сценарный анализ |
| | Статистический анализ |
| | Многомерный анализ |
| | Отчетность |
| | Планирование |

Рис. 2. Шестиуровневая архитектура ХД

быть введен промежуточный слой прикладного программного обеспечения (ПО), которое берет на себя функции нивелирования технических решений разных автоматизированных систем путем реализации новых и поддержки некоторых старых API-интерфейсов (*application programming interface* — интерфейс прикладного программирования) [6].

При хранении данных на третьем уровне обеспечивается распределенный доступ к ХД, резервирование информации, защита от сбоев программного и аппаратного обеспечения.

При выборке, реструктуризации и доставке данных на четвертом уровне определяется частота обновления данных в ХД и процедуры обработки данных. В соответствии с заданной частотой и процедурами формируются требования к пропускной способности каналов связи и алгоритмам управления трафиком и распределения нагрузки.

Цели визуализации данных на пятом уровне должны быть четко определены [7]. Для визуализации результатов анализа, отражающих особенности конструкций технических изделий, элементы математической статистики, данные о фактических расходах на проектирование и эксплуатацию, целесообразно применять технологии N -мерного моделирования [8]. На пятом уровне также должны быть предусмотрены средства обеспечения восстановления данных.

Использование элементов математической статистики, сценарного и многомерного анализа, планирования для контроля и последующей оптимизации фактических данных расхода информационных, энергетических, финансовых, материальных и трудовых ресурсов повышает эффективность управления с помощью СППР [9—11].

В данной работе рассматриваются методы и сценарии сбора, извлечения и загрузки данных на основе унификации регламентов взаимодействий с поставщиками информационных ресурсов.

Методы и сценарии извлечения, сбора и загрузки данных

Операционные метаданные загрузки задают множество поставщиков:

$$U^S = \{S_1, S_2, \dots, S_{|U^S|}\}, \quad (1)$$

где S_i ($i = 1 \dots |U^S|$) — поставщик.

На практике одна организация может поставлять данные для нескольких моделей хранилища. Поэтому каждый поставщик определяет множество логических блоков данных:

$$S = \{DB_1, DB_2, \dots, DB_{|S|}\}, \quad (2)$$

где DB_{ij} ($j = 1 \dots |S_i|$) — блок данных поставщика S_i .

Блок данных является основным понятием и определяется тройкой

$$DB = (t, Ex, Ld), \quad (3)$$

где t — тип поставщика; Ex — метаданные уровня извлечения; Ld — метаданные уровня загрузки.

Для каждого типа поставщика разработан свой метод извлечения данных. В случае, когда поставщиком является закрытая система, метаданные уровня извлечения не проектируются ($Ex = \otimes$), так как метод извлечения данных и способы их предоставления определяются самим поставщиком.

Метод автоматизации сбора агрегированных данных поставщика-клиента обеспечивает проектирование средств массового сбора статистической отчетности. На основе типизации структур аналитических моделей хранилища, ориентированных на многомерный анализ ведомственной и корпоративной статистики, разработан унифицированный шаблон средства сбора, определяющий метаданные уровня извлечения ($Ex = EX^{SC1}$). Метод включает последовательное построение трех основных разделов шаблона: "Объект мониторинга", "Показатель деятельности" и "Период отчетности", соответствующих базовым типам измерений аналитической модели.

При проектировании раздела "Объект мониторинга" используется соответствующая навигационная модель хранилища, поддерживающая движение по уровням иерархии исследуемой системы. Навигационная модель данных характеризуется тем, что в узлах иерархического механизма хранилища данных достаточно расположить только атрибут, связанный лишь с одним адресным указателем, на месторасположение данных (в виде значения смещения в байтах от начала файла), записанных последними по порядку в первичную структуру хранения информации. Такой иерархический механизм поиска может быть целиком размещен в оперативной памяти компьютера.

Иерархия механизма хранилища данных определяется объектами мониторинга и составляет метаописание уровня извлечения:

$$EX^{SC1} = \{O_1, O_2, \dots, O_{|EX^{SC1}|}\}. \quad (4)$$

Объект мониторинга задается кортежем

$$O = (st, \bar{A}^{D_k}, U^{A^{1:1}}, U^F),$$

где st — вид поиска экземпляра объекта; \bar{A}^{D_k} — корневой атрибут дерева навигационной модели хранилища; $U^{A^{1:1}}$ — множество атрибутов, используемых для поиска необходимого экземпляра объекта; U^F — множество форм ввода.

Множество U^F определяет шаблоны форм, которые формируются из показателей аналитической модели при проектировании раздела "Показатель деятельности". Шаблон формы $F \in U^F$ определяется кортежем

$$F = (P, M, X^1, X^2, X^3),$$

где P — период отчетности; M — множество показателей аналитической модели; X^1 — синтаксические ограничения, накладываемые на вводимые значения показателей множества M ; X^2 — внутренние семантические ограничения, накладываемые на вводимые значения показателей множества M ; X^3 — межуровневые семантические ограничения, накладываемые на предоставленные значения показателей множества M .

Проектирование раздела "Период отчетности" выполняется по временным измерениям аналитической модели хранилища (типовые элементы — "год", "квартал", "месяц" и др. [6]).

Метод автоматизации сбора детализированных данных поставщика-клиента обеспечивает сбор регистрационных сведений в процессе ведомственного учета. Для этого используют формы, спроектированные на основе навигационных метаданных хранилища. Метод включает выбор интересующей навигационной модели, построение иерархии объектов ведомственного учета и прототипов соответствующих форм сбора, задание семантических ограничений и визуальных параметров элементов управления форм.

В основе проектирования лежит навигационная модель хранилища. С ее помощью выполняется построение метаданных уровня извлечения ($Ex = EX^{SC2}$), которые представляют собой иерархию объектов ведомственного учета:

$$EX^{SC2} = \{O_1, O_2, \dots, O_{|EX^{SC2}|}\}, \quad (5)$$

где O_i ($1 \leq i \leq |EX^{SC2}|$) — объект учета.

Объект в иерархии определяет сущность предметной области и задается кортежем

$$O = (parent, rt, st, \bar{A}^{D_k}, U^{A^{1:1}}, X, C, V), \quad (6)$$

где $parent$ — родительский объект в иерархии; rt — тип связи с родительским объектом, который может принимать значения из множества $\{1: M, M:1, M: N\}$; st — тип отображения формы объекта на форму объекта $parent$; \bar{A}^{D_k} — корневой атрибут дерева навигационной модели; $U^{A^{1:1}}$ — множество атрибутов объекта; X — семантические ограничения на значения атрибутов объекта; C — множество дочерних объектов; V — визуальные параметры отображения элементов ввода атрибутов объекта.

Для каждого объекта $O \in EX^{SC2}$ автоматически выполняется построение прототипа пользовательской формы ввода. Она представляет собой набор элементов управления, ассоциированных с соответствующими атрибутами объекта. При этом анализируются свойства, влияющие на построение шаблона формы: является ли атрибут ссылкой с типом связи $1:M$ или $M:N$ для формирования подчиненной формы; ссылается ли атрибут на справочник

($M:1$) для формирования соответствующего списка выбора и др.

Метод автоматического извлечения данных поставщиков открытых систем использует подход виртуальной интеграции, развиваемый учеными Дж. Ульманом, Д. Хопкрофтом, М. Лензерини [13,14]. Основой систем виртуальной интеграции данных является концепция медиатора (от англ. *mediator* — посредник), выполняющего оптимизацию гетерогенных запросов к разнородным системам хранения данных. Разработан специализированный посредник, названный "агентом". Он поддерживает основные языки манипулирования данными для соответствующего переформулирования унифицированного запроса хранилища в форму, используемую в целевой системе хранения поставщика.

Метод обеспечивает проектирование унифицированного запроса на основе навигационной модели блока данных поставщика U^A , представленной графом атрибутов, наделенных определенными свойствами (тип связи с вышестоящим атрибутом; признак — является ли атрибут потенциально корневым; название целевой таблицы и поля в базе данных и др.). Модель U^A , поддерживаемая на уровне метаданных хранилища, определяет концептуальное представление структуры данных системы хранения поставщика. С помощью этого метода выбирается корневой атрибут, определяются интересующие атрибуты выборки, задаются условия ограничения результирующего набора данных и функций их преобразования. Спроектированный запрос определяет метаданные уровня извлечения ($Ex = EX^{OP}$) и представляется кортежем

$$EX^{OP} = (\bar{A}^{D_k}, U^{\hat{A}}, RE, FN^{\hat{A}}, TIME), \quad (7)$$

где \bar{A}^{D_k} — выбранный корневой атрибут дерева D_k ,

относительно которого строится запрос; $U^{\hat{A}}$ — множество атрибутов дерева D_k , по которым запрашиваются данные; RE — условия ограничения результирующей выборки; $FN^{\hat{A}}$ — множество функций преобразования данных результата; $TIME$ — регламент выполнения запроса на выборку данных.

На стадии извлечения спроектированный запрос преобразуется в форму языкового средства манипулирования данными целевой системы хранения для обеспечения выборки необходимой СППР информации.

Метод загрузки данных из разнородных электронных источников применяется для обработки информации, поступающей с уровня извлечения. В основу метода положены концепция потокового представления электронного документа, событийный механизм обработки данных, теория семантических меток, языки запросов по оцениваемым и

неоцениваемым деревьям, предложенные К. Кохом, М. Бенедиктом, Л. Либкиным [15, 16].

Электронный источник представляется набором элементов данных, снабженных структурными метками, называемых "трассами значений". Трассы однозначно определяют функции элементов, что позволяет установить соответствие данных источника атрибутам целевых сущностей хранилища. Метод обеспечивает проектирование соответствий на основе унифицированного представления схемы данных источника и включает операции трассировки, уточнения и привязки, в результате чего формируются метаданные уровня загрузки.

Метаданные уровня загрузки Ld описывают процесс интеграции данных электронного источника в целевые структуры хранения последовательностью шагов трансформации:

$$Ld = (f_1, M_1, f_2, M_2, \dots, M_n), \quad (8)$$

где f_i — схема данных электронного источника для i -го шага трансформации; для первого шага ($i = 1$) схема данных f_1 определяет электронный источник, полученный на уровне извлечения; M_i — карта привязки, определяющая правила преобразования данных электронного источника f_i в реляционные структуры хранения; n — число шагов трансформации.

Схема f задается с помощью унифицированного формата описания структуры электронного источника на основе модели иерархии классов:

$$f = \{Elm_1, Elm_2, \dots, Elm_{|f|}\}, \quad (9)$$

где Elm — элемент структуры источника,

$$Elm = (parent, t^S, n^S, A^S, cond^S, C), \quad (10)$$

здесь $parent$ — родительский элемент (элементы, имеющие неопределенный родительский элемент, есть корневые элементы иерархий); t^S — семантическое название элемента, используемое при проектировании; n^S — структурная метка элемента; A^S — множество атрибутов элемента (может быть пустым); $cond^S$ — ограничения на значения атрибутов; C — множество дочерних элементов.

Карта привязки M определяет функциональные соответствия структур данных источника атрибутам сущностей хранилища:

$$M = (L, U^{\tilde{E}}), \quad (11)$$

где L — множество соответствий структур источника

целевым атрибутам сущностей хранилища; $U^{\tilde{E}}$ — промежуточная область, используемая для выполнения трансформации.

Множество соответствий $L = \{l_1, l_2, \dots, l_{|L|}\}$ определяет структурные привязки в виде:

$$l = (p, A^p, FN^{A^p}), \quad (12)$$

где p — структура данных источника, определяющая элементарное значение (задается шаблоном трассы); A^p — множество атрибутов сущностей хранилища, которые соответствуют заданной структуре данных источника (может содержать атрибуты разных сущностей); FN^{A^p} — множество функций преобразования значений для целевых атрибутов множества A^p .

Построение множества L карты M выполняется на основе элементарных структур электронного источника (листьев сформированных деревьев множества f). Листьями деревьев являются атрибуты элементов. Для установления соответствия атрибутов целевым структурам хранения множество интересующих проектировщика листьев $L_f \subseteq U^{A^S}$, где

$$U^{A^S} = \bigcup_{i=1}^{|f|} A_i^S, \text{ преобразуется в структурные метки.}$$

Такие метки получили название "шаблоны трасс" элементарных структур.

На основе деревьев f автоматически формируются соответствующие шаблоны трасс. Такая операция получила название "трассировка" схемы данных источника:

$$\text{trace} : L^f \rightarrow P, \quad (13)$$

где P — множество шаблонов трасс элементарных структур источника. Функция trace преобразует атрибуты $a^S \in L^f$ в трассы-шаблоны $p \in P$ следующего вида:

$$p = (Elm_1, Elm_2, \dots, Elm_n, a^S), \quad (14)$$

где Elm_z — элемент иерархии, являющийся z -м предком для атрибута a^S , при $1 \leq z \leq n$; n — длина шаблона трассы.

Для каждого элемента шаблона возможно определение условных ограничений cond на значения атрибутов с помощью операции уточнения:

$$\text{adjust} : P \rightarrow P'. \quad (15)$$

Ограничения задаются И/ИЛИ деревом условий, в которых используются атрибуты элементов трассы. При этом в множество P добавляется новая трасса p'_k , включающая исходную p_k , с заданными ограничениями соответствующих элементов. Использование ограничений повышает функциональность предлагаемого метода.

На основе уточненного множества P' задаются функциональные соответствия шаблонов трасс целевым структурам хранилища. Данная операция получила название "привязка" шаблонов трасс. Набор используемых для привязки функций унифицирован и включает функцию приведения типов, функцию форматирования, функцию слияния, функцию согласования. Процедура привязки вы-

полняется в два этапа. На первом этапе формируется множество L карты M :

$$\text{link}_1 : P' \times U^{A^E} \rightarrow L. \quad (16)$$

При этом явно указываются соответствия шаблонов трасс атрибутам сущностей хранилища U^{A^E} . Для элементов L возможно задание функций обработки данных (FN^{A^p}). На основе результатов выполнения первого этапа привязки формируется множество сущностей $U^{\tilde{E}} \subseteq U^E$, атрибуты которых были связаны с шаблонами трасс, где U^E — множество всех сущностей хранилища, поддерживаемое на уровне концептуальных метаданных. Множество $U^{\tilde{E}}$ определяет сущности хранилища, в которые будет происходить загрузка данных по карте M . На втором этапе выполняется формирование промежуточной области $U^{\tilde{E}} = \{\tilde{E}_1, \tilde{E}_2, \dots, \tilde{E}_{|U^{\tilde{E}}|}\}$ как проекции множества $U^{\tilde{E}}$:

$$\text{link}_2 : U^{\tilde{E}} \rightarrow U^{\tilde{E}}. \quad (17)$$

Элементом промежуточной области служит табличный кэш

$$\tilde{E} = (E, p^{\tilde{E}}, u), \quad (18)$$

где $E \in U^{\tilde{E}}$ — целевая сущность хранилища; $p^{\tilde{E}}$ — структура данных источника, описывающая логическую запись (задается шаблоном трасс); u — режим добавления записи исходных данных источника в таблицу целевой сущности.

При проектировании табличного кэша задается шаблон трассы $p^{\tilde{E}}$ для определения условий переноса данных из промежуточной области в таблицу целевой сущности E и режим добавления данных u (безусловная вставка, условная вставка, условное обновление, вставка-обновление) для определения способа слияния данных.

Осуществление гибкого масштабирования среды информационного наполнения ХД СППР реализуется посредством предложенной структуры инструментальной системы, показанной на рис. 3.

Как видно из рис. 3, в ХД применяются WORM (от англ. *write once, read many* — носители информации, допускающие однократную запись и многократное чтение) для хранения архивных копий и защиты от случайного/преднамеренного изменения или удаления.

Дедупликация данных создает образы виртуальных дисков для хранилищ, которые будут содержать только уникальные данные записываемых на них файлов.

Для каждого инструментального средства сбора данных в хранилище СППР можно подсчитать показатели эффективности, например, снижение временных и финансовых затрат на извлечение данных благодаря организации информационных взаимодействий с учетом различных типов поставщиков.

Выводы

Описанный подход позволяет решить проблему интеграции данных различных поставщиков информационных ресурсов для поддержки принятия решений в СППР. Использование разработанных методов и средств обеспечивает повышение эффективности извлечения данных и загрузки их в хранилище посредством унификации обработки электронных источников различных форматов независимо от их структуры и размера.

Разработанный программный функционал позволяет реализовать инструментальный подход к организации сбора данных в хранилище СППР на основе новейших информационных технологий.

Предложенные аналитические соотношения для различных методов сбора данных описывают процесс интеграции данных электронного источника в целевые структуры хранения.

Список литературы

1. Орешков В. И., Паклин Н. Б. Бизнес-аналитика: от данных к знаниям: учеб. пособие. 2-е изд. СПб.: Питер, 2013. 704 с.
2. Кузьмин А. Н. Трехуровневая архитектура хранилища данных с интерфейсом запросов // Социально-экономические и технические системы: исследование, проектирование, оптимизация. 2006. № 2. С. 3.
3. Егоров С. Л. Разработка архитектуры информационного хранилища данных в процессе проектирования системы поддержки принятия решений // Известия высших учебных заведений. Уральский регион. 2010. № 4. С. 18–21.
4. Григорович А. Г. Архитектура хранилищ данных с ненормализованными отношениями (Data Warehouse Architecture with unnormalized relations) // Технические науки — от теории к практике. 2013. № 22. С. 6–10.
5. Белошицкий Д. А. Интеграция данных в информационных системах // Молодежный научно-технический вестник. 2013. № 8. С. 32.
6. Илюшин Г. Я., Соколов И. А. Организация управляемого доступа пользователей к разнородным ведомственным информационным ресурсам // Информатика и ее применения. 2010. Т. 4, № 1. С. 24–40.

| Программный функционал | | | |
|---|--|---------------------------------------|---|
| Надежность хранения | Оптимизация хранения | Оптимизация производительности | Управление данными |
| Локальная репликация данных | Динамическое управление размером томов | Балансировка производительности ХД | Виртуализация сторонних ХД |
| Репликация данных на другое устройство | Дедупликация данных | Виртуальные копии томов | Программа-агент для манипулирования данными |
| Мгновенные снимки | Компрессия (сжатие) и уплотнение данных | Балансировка хранения по типам дисков | Интеграция с антивирусом |
| Быстрое восстановление из мгновенных снимков | | | |
| Поддержка WORM | Виртуальная интеграция данных | Потоковое представление документов | Аутентификация и авторизация |
| Создание копий прикладного программного обеспечения | Режимы хранения в виде многомерных кубов | «Трассировка» схемы данных источника | Шифрование |

Рис. 3. Состав программного функционала для информационного наполнения ХД

7. Соханевич С. В. Хранилище данных как базовый элемент построения инструментальных систем поддержки принятия управленческих решений // Известия ЮФУ. Технические науки. 2011. № 11 (124). С. 205–208.

8. Акимкина Э. Э. Развитие и адаптация имитационного и компьютерного моделирования в системах поддержки принятия решений // Современные информационные технологии. Сборник трудов по материалам II межвузовской научно-технической конференции, 14 сентября 2016 года, г. о. Королев, "МГОТУ" / Под общей науч. ред. В. М. Артюшенко. М.: Научный консультант, 2016. С. 112–121.

9. Аббасов Э. М., Акимкина Э. Э. Достижение максимальной производительности при работе с крупными хранилищами данных // Информационные технологии. Радиоэлектроника. Телекоммуникации (ITRT-2016): сб. статей VI международной заочной научно-технической конференции. 24–25.03.2016. Ч. 1 / Поволжский гос. ун-т сервиса. Тольятти: Изд-во: ПВГУС, 2016. С. 7–12.

10. Аббасов А. Э., Аббасов Т. Э. Оценка качества программного обеспечения для современных систем обработки информации // Информационно-технологический вестник. 2015. № 3 (05). С. 15–27.

11. Аббасова Т. С. Совмещение управляющих и измерительных функций при интерактивном управлении телекоммуникационными системами // Информационно-технологический Вестник. 2015. № 2 (04). С. 14–38.

12. Акимкина Э. Э., Самаров К. Л., Васильев Н. А., Белошенко И. М. Оценка эффективности обслуживания клиентов в контексте системного анализа // Вопросы региональной экономики. 2016. № 2 (27). С. 123–134.

13. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы. М.: Вильямс, 2001. 384 с.

14. Batini C., Lenzerini M. A. Methodology for data schema integration in the entity relationship model // IEEE Trans. Softw. Eng. SE-10, 1984. 6 (Nov.). P. 650–663.

15. Gottlob G., Koch C. Logic-based Web Information Extraction // SIGMOD Record. June 2004. Vol. 33, N. 2. P. 87–94.

16. Benedikt M., Libkin L., Neven F. Logical definability and query languages over ranked and unranked trees // ACM Transactions on Computational Logic (TOCL). 2007. N. 8 (2). P. 1–62.

Instrument Approach to the Organization of Data Collection Warehouse Decision Support Systems

Analyzed the problems of modern means of integration in the storage decision support systems A six-level architecture is developed that differs from the widely used three-level architecture in that the levels introduced take into account the problems of extracting data using the latest information technologies from various information resource providers and enable detailed data processing operations within each level. Analytical dependencies for various methods and scenarios of extraction, collection and loading of data are determined; Requirements for the software functionality implementing the six-level service-oriented data storage architecture are formulated.

The approach to organizing data collection in the DDS repository is to select methods and scenarios for collecting, retrieving and loading data based on the unification of the rules of interaction with information resource providers.

This approach allows to solve the problem of data integration of various information resource providers to support decision-making in the Decision Support System (DSS). The use of the developed methods and tools provides an increase in the efficiency of extracting data and loading it into the repository by unifying the processing of electronic sources of various formats regardless of their structure and size.

The developed software functionality allows implementing an instrumental approach to organizing data collection in the DSS repository based on the latest information technologies.

The proposed analytical relationships for various methods of data collection describe the process of integrating the data of the electronic source into the target storage structures.

Keywords: data integration, data collection methods, levels of detail, service-oriented solutions

References

1. **Oreshkov V. I., Paklin N. B.** *Biznes-analitika: ot dannyh k znanijam* [Business Intelligence: from data to knowledge], Uchebnoe posobie (tutorial). 2-e izdanie. Sankt-Peterburg, Piter, 2013. 704 p. (in Russian).
2. **Kuz'min A. N.** Trjohurovnevaja arhitektura hranilishha dannyh s interfejsom zaprosov [Three-tier architecture of data warehouse query interface], *Social'no-jekonomicheskie i tehniczeskie sistemy: issledovanie, proektirovanie, optimizacija*, 2006, no. 2, pp. 3 (in Russian).
3. **Egorov S. L.** Razrabotka arhitektury informacionnogo hranilishha dannyh v processe proektirovanija sistemy podderzhki prinjatija reshenij [Architecting the data warehouse information in the process of designing a decision support system], *Izvestija vysshih uchebnyh zavedenij. Ural'skij region*, 2010, no. 4, pp. 18–21 (in Russian).
4. **Grigorovich A. G.** Arhitektura hranilishh dannyh s nenormalizovannymi otnoshenijami [Data Warehouse Architecture with unnormalized relations] *Tehniczeskie nauki — ot teorii k praktike* [Engineering — From Theory to Practice], 2013, no. 22, pp. 6–10 (in Russian).
5. **Beloshickij D. A.** Integracija dannyh v informacionnyh sistemah [Integration of data in information systems], *Molodezhnyj nauchno-tehniczeskij vestnik* [Youth Science and Technology Gazette], 2013, no. 8, pp. 32 (in Russian).
6. **Iljushin G. Ja.** Organizacija upravljajemogo dostupa pol'zovatelej k raznorodnym vedomstvennym informacionnym resursam [Organization of managed user access to heterogeneous departmental informational resources], *Informatika i ee primenenija* [Computing and its Applications], 2010, vol. 4, no. 1, pp. 24–40 (in Russian).
7. **Sohanevich S. V.** Hranilishhe dannyh kak bazovyj jelement postroenija instrumental'nyh sistem podderzhki prinjatija upravlencheskih reshenij [Data Warehouse as a basic element of construction tools support management decision-making systems], *Izvestija JuFU. Tehniczeskie nauki* [Proceedings of SFU. Technical science], 2011, no. 11 (124), pp. 205–208 (in Russian).
8. **Akimkina E. E.** Razvitie i adaptacija imitacionnogo i komp'juternogo modelirovanija v sistemah podderzhki prinjatija reshenij [The development and adaptation of simulation and computer modeling in the decision support systems], *Sovremennye informacionnye tehnologii, sbornik trudov po materialam II mezhhuzovskoj nauchno-tehniczeskoj konferencii, 14 sentjabrja 2016 g., g. Korolev, "MGOTU"*, Moscow, Nauchnyj konsul'tant, 2016, pp. 112–121 (in Russian).
9. **Abbasov E. M., Akimkina E. E.** Dostizhenie maksimal'noj proizvoditel'nosti pri rabote s krupnymi hranilishhami dannyh [Achieving peak performance when working with large data stores], *Informacionnye tehnologii. Radiojelektronika. Telekommunikacii (ITRT-2016): sb. statej VI mezhdunarodnoj zaochnoj nauchno-tehniczeskoj konferencii. Ch. 1, Povolzhskij gos. un-t servisa*, 24–25.03.2016, Tol'jatti: Izd-vo: PVGUS, pp. 7–12 (in Russian).
10. **Abbasov A. E., Abbasov T. E.** Ocenka kachestva programmnogo obespechenija dlja sovremennyh sistem obrabotki informacii [Evaluation of software quality for modern data processing systems], *Informacionno-tehniczeskij Vestnik*, 2015, no. 3 (05), pp. 15–27 (in Russian).
11. **Abbasova T. S.** Sovmeshhenie upravljajushih i izmeritel'nyh funkcij pri interaktivnom upravlenii telekommunikacionnymi sistemami [The combination of control and measurement functions for the interactive management of telecommunication systems], *Informacionno-tehniczeskij Vestnik*, 2015, no. 2 (04), pp. 14–38 (in Russian).
12. **Akimkina E. E., Samarov K. L., Vasil'ev N. A., Beljuchenko I. M.** Ocenka jeffektivnosti obsluzhivanija klientov v kontekste sistemnogo analiza [Evaluating the effectiveness of customer service in the context of the system analysis], *Voprosy regional'noj jekonomiki*, 2016, no. 2 (27), pp. 123–134 (in Russian).
13. **Aho A., Hopcroft Dzh., Ul'man Dzh.** *Struktury dannyh i algoritmy*, Moscow, Willams, 2001, 384 p. (in Russian).
14. **Batini C., Lenzerini M. A.** Methodology for data schema integration in the entity relationship model, *IEEE Trans. Softw. Eng. SE-10*, 1984, no. 6 (Nov.), pp. 650–663.
15. **Gottlob G., Koch C.** Logic-based Web Information Extraction [Text] / G. Gottlob, *SIGMOD Record*, 2004, vol. 33, no. 2, pp. 87–94.
16. **Benedikt M., Libkin L., Neven F.** Logical definability and query languages over ranked and unranked trees, *ACM Transactions on Computational Logic (TOCL)*, 2007, no. 8 (2), pp. 1–62.

А. С. Акопов, д-р техн. наук, доц., проф., e-mail: aakopov@hse.ru,

А. Л. Бекларян, канд. техн. наук, преподаватель, e-mail: abeklaryan@hse.ru,

Н. К. Хачатрян, канд. физ.-мат. наук, доц., e-mail: nkhatatryan@hse.ru,

А. В. Фомин, канд. экон. наук, доц., e-mail: afomin@hse.ru

Национальный исследовательский университет "Высшая школа экономики", г. Москва

Система прогнозирования динамики добычи нефти с использованием имитационного моделирования

Представлена оригинальная информационно-аналитическая система, разработанная с использованием методов параметрической аппроксимации и имитационного моделирования и предназначенная для сценарного прогнозирования динамики добычи нефти по скважинам. Данная система реализована в виде программного комплекса, ядром которого является разработанная в среде AnyLogic имитационная модель, интегрированная с базой данных системы и описывающая динамику добычи на каждой скважине с учетом реализованных и плановых геолого-технических мероприятий (ГТМ). Прототип системы был успешно внедрен в крупнейшей российской нефтяной компании и используется для прогнозирования дополнительной добычи нефти по всему пулу имеющихся скважин (более 100 000 скважин на десятилетнем временном интервале), а также для оценки потенциального эффекта от ГТМ в целях последующего перераспределения ресурсов между скважинами.

Ключевые слова: имитационное моделирование, параметрическая аппроксимация, прогнозирование добычи нефти, сервисно-ориентированное решение, AnyLogic

Введение

В настоящее время наблюдается усложнение требований, предъявляемых к системам прогнозирования добычи нефти по скважинам, в связи с существенным ростом информационной базы, обусловленным необходимостью управления сверхбольшим пулом производственных объектов (десятки тысяч скважин со своими характеристиками) с учетом влияния множественных мероприятий, в частности, различных геолого-технических мероприятий, остановок, ремонтов, запусков и т. д.

Добыча нефти на нефтяной скважине — динамический процесс с нелинейными обратными связями, множественными технологическими и финансовыми ограничениями. Несмотря на то что фундаментальные методы прогнозирования добычи нефтяной скважины известны в течение длительного времени, начиная с уравнения Дюпюи и закона Дарси [1], проблема оптимального выбора геолого-технических мероприятий (ГТМ) на нефтяных скважинах по-прежнему очень важна для нефтяных компаний, имеющих крупные фонды нефтяных скважин и дефицит ресурсов. Иногда посредством повышения эффективности работы отдельной нефтяной скважины можно возместить ущерб от времени простоя нескольких скважин. Специфические особенности таких нефтяных компаний обуславливают потребность комплексного динамического моделирования их деятельности с учетом взаимосвязанных производственных и инвестиционных процессов [2—6].

Нужно отметить, что задача выбора оптимального набора ГТМ на нефтяной скважине обуслов-

ливает необходимость прогнозирования дополнительной нефтедобычи как разницы между двумя прогнозами до и после вмешательства для каждой нефтяной скважины соответственно.

Можно использовать различные методы для прогнозирования добычи нефтяной скважины — как известные статистические методы [7], так и физические методы гидродинамического моделирования [8], а также методы имитационного моделирования, в частности, системной динамики и дискретно-событийного моделирования [2—4, 9] для учета влияния ГТМ на динамику добычи. Нужно отметить, что использование исключительно статистических методов без производственного моделирования нефтяной скважины существенно ограничивает применимость подобных моделей прогнозом базовой добычи (т.е. добычи до ГТМ).

В условиях неоднородного вмешательства в производственный процесс посредством ГТМ объединение статистических и гидродинамических методов для прогноза нефтедобычи более предпочтительно. Очевидно различные ГТМ, такие как общее производительное обслуживание (ТРМ), увеличение нефтеотдачи (ОП), добыча нефти вторичным методом (EOR) оказывают различное влияние на увеличение нефтедобычи. В то же время каждая из крупнейших нефтяных компаний, таких как *Saudi Aramco*, Китайская национальная нефтегазовая корпорация, *Exxon Mobil*, Роснефть и другие, управляет более чем ста тысячами нефтяных скважин. Поэтому лицам, принимающим решения (ЛПР), необходимо оценивать множество альтернатив, максимизирующих производство нефтяной скважины и минимизирующих требуемые ресурсы.

Для поиска оптимальных решений о ресурсах, перераспределяемых между нефтяными скважинами могут быть применены различные алгоритмы оптимизации нулевого порядка, в частности генетические оптимизационные алгоритмы [10].

Цель данной статьи — разработка системы, предназначенной для прогнозирования базовой и дополнительной добычи нефти на скважинах с учетом различных ГТМ и с использованием методов имитационного моделирования. Предлагаемый подход позволяет, в частности, объединить различные методы прогнозирования добычи в рамках единой информационно-аналитической системы в целях последующего определения наилучших альтернатив (ГТМ) на скважинах при имеющихся ограничениях, а также визуализировать все существующие алгоритмы расчета и этапы вычислительной процедуры и интегрировать расчетные алгоритмы с базой данных системы.

Для реализации программного комплекса используется система имитационного моделирования *AnyLogic* [9], интегрированная с базой данных системы и веб-сервисом, обеспечивающим программное управление всеми вычислительными процедурами в параллельном режиме для обеспечения множественных расчетов в режиме реального времени.

1. Имитационная модель

Разработанная модель основана на вычислении накопленной добычи нефтяной скважины до и после ГТМ.

Для этого необходимо выполнить прогнозирование базовой добычи нефтяной скважины в течение всего периода перед любой ГТМ для нефтяной скважины, в частности с помощью метода параметрической аппроксимации.

Введем следующие обозначения:

$w \in \{1, 2, \dots, W(t)\}$ — индекс скважин; $i_w \in \{1, 2, \dots, I_w(t)\}$ — индекс точек входных данных w -й скважины до ГТМ (например, данные дневной добычи); $t \in \{t_0, t_0 + 1, \dots, T + t_0\}$ — время по дням; t_0 — начальное модельное время, T — горизонт прогнозирования;

$\{x_{i_w}(t); y_{i_w}(t)\}_{i_w=1}^{I_w(t)}$ — координаты значений фактических данных, которые должны быть аппроксимированы;

$$f_w(x_w(t)) = a_w(t)x_w(t) + b_w(t)y_w(t) + c_w(t). \quad (1)$$

Допустим, все входные точки имеют веса ($v_{i_w}(t)$). Тогда можно вычислить отклонение аппроксимирующей прямой от фактических данных $\{x_{i_w}(t); y_{i_w}(t)\}$ с помощью простой разности:

$$D_w(t) = \sum_{i_w=1}^{I_w(t)} ((y_{i_w}(t) - a_{0,w}(t) - a_{1,w}(t)x_{i_w}(t)) \cdot v_{i_w}(t))^2, \quad (2)$$

где

$$a_{0,w}(t) = -c_w(t)/b_w(t); \quad (3)$$

$$a_{1,w}(t) = -a_w(t)/b_w(t). \quad (4)$$

В результате решения оптимизационной задачи

$$\min_{a_{0,w}(t), a_{1,w}(t)} \{D_w(t)\}, \quad (5)$$

мы получим

$$a_{1,w}(t) = \frac{N_{Y,w}(t) \cdot M_{V,w}(t) - M_{X,w}(t) \cdot M_{Y,w}(t)}{N_{V,w}(t) \cdot N_{X,w}(t) - M_{X,w}^2(t)}; \quad (6)$$

$$a_{0,w}(t) = \frac{M_{Y,w}(t) - a_{1,w}(t) \cdot M_{X,w}(t)}{N_{W,w}(t)}, \quad (7)$$

где

$$M_{X,i_w}(t) = \sum_{i_w=1}^{I_w(t)} x_{i_w}(t) \cdot v_{i_w}^2(t); \quad (8)$$

$$M_{Y,i_w}(t) = \sum_{i_w=1}^{I_w(t)} y_{i_w}(t) \cdot v_{i_w}^2(t); \quad (9)$$

$$N_{X,i_w}(t) = \sum_{i_w=1}^{I_w(t)} x_{i_w}^2(t) \cdot v_{i_w}^2(t). \quad (10)$$

Динамика экспоненциально аппроксимируемой базовой добычи w -й скважины

$$Q_w(t) = e^{(a_{0,w}(t) + a_{1,w}(t)t)}, \quad w \in \{1, 2, \dots, W(t)\}. \quad (11)$$

Аналогично можно вычислить динамику добычи нефти после ГТМ — $\tilde{Q}_w(t)$. При этом в качестве входных данных берется фактический дневной дебит скважины до ГТМ с учетом ожидаемого производственного эффекта в виде прироста начального дебита скважины.

Дополнительная добыча вычисляется на интервале между каждыми соседними ГТМ:

$$\sum_{w=1}^{W(t)} \sum_{j_w}^{J_w(t)} (\tilde{Q}_{w,j_w}(t) - Q_{w,j_w}(t)); \quad (12)$$

$$Q_{w,j_w}(t) = \tilde{Q}_{w,j_w-1}(t), \quad (13)$$

где $j_w \in \{1, 2, \dots, J_w(t)\}$ — индекс ГТМ, осуществляемых на w -й скважине.

Таким образом, можно сформулировать следующую оптимизационную задачу для сегмента "нефтедобыча".

Задача. Требуется максимизировать дополнительную добычу нефти на скважинах за счет выбора наилучшего набора ГТМ:

$$\max_{\delta_{j_w}} \left[\sum_{w=1}^{W(t)} \sum_{j_w} (\tilde{Q}_{w,j_w}(t) - Q_{w,j_w}(t)) \right]; \quad (14)$$

$$\sum_{w=1}^{W(t)} \sum_{j_w} (I_{w,j_w}(t)) \leq CAPEX, \quad (15)$$

где $\delta_{j_w} \in \{0, 1\}$ — управляющий параметр модели; $\delta_{j_w} = 0$ — отказ от проведения j_w -го ГТМ на скважине; $\delta_{j_w} = 1$ — принятие решения о проведении ГТМ; $I_{w,j_w}(t)$ — инвестиционные расходы, требуемые для проведения ГТМ; $CAPEX$ — лимит инвестиционных расходов.

Отметим, что в задаче (14) могут быть включены и другие ограничения, например, лимит операционных расходов, минимально необходимый уровень дополнительной добычи, ограничение на материально-техническую базу, необходимую для проведения ГТМ и др. Для решения задачи типа (14) целесообразно использовать генетические оптимизационные алгоритмы [9], например, ГА с угасающей селекцией [10], что обосновано сверхбольшой размерностью подобной задачи (сотни тысяч скважин с десятками ГТМ на каждой скважине в течение 10–15 лет), а также значимостью выбора последовательности проведения ГТМ и наличием внутренних нелинейных зависимостей, в частности, связанных с использованием нелинейных методов аппрокси-

мации для ГТМ сложного типа (например, при гидроразрыве пласта).

Отметим, что при наличии данных о направлениях поставок добываемой нефти, транспортных и прочих издержек, структуре экспорта и других, можно вычислить NPV (чистую приведенную стоимость) сегмента "нефтедобыча", как это было ранее показано в работах [2–5, 11] в рамках задачи максимизации акционерной стоимости ВИНК (вертикально-интегрированной нефтяной компании).

2. Описание программного комплекса

Ядром разработанной системы является имитационная модель прогнозирования добычи, реализованная в системе *AnyLogic*, обеспечивающей необходимый пользовательский интерфейс (рис. 1).

Особенностью разработанного интерфейса является возможность выбора метода прогнозирования добычи нефти в сценарном режиме (например, аппроксимация методом МНК, падение с заданным темпом и т. д.). При этом имеется возможность сохранения всех сценарных расчетов в базе данных системы в целях дальнейшего анализа полученных результатов и выбора наилучшего метода прогнозирования добычи на основе оценки предикативного качества модели для ГТМ, относящихся к определенным типам.

Важным преимуществом используемой системы *AnyLogic* является возможность визуального проектирования всех расчетных алгоритмов с использованием диаграмм действий и методологии UML (*Unified Modeling Language*). Пример программной реализации модели с использованием такого подхода представлен на рис. 2, 3.

При этом важным отличием *AnyLogic* от других инструментов, поддерживающих UML, является реализация полноценной функциональности *Java*-кода на уровне всех элементов UML-диаграммы (например, переменных, управляющих конструкций, циклов, блоков кода и др.). Таким образом, визуально проектируемая диаграмма является исполнимой вычислительной процедурой, написанной на языке программирования *Java* и интегрируемой с другими элементами имитационной модели, например, с системами дифференциальных уравнений или конечно-разностных уравнений, использующих результаты подобной (встраиваемой) вычислительной процедуры.

На рис. 3 представлен фрагмент реализации метода наименьших квадратов (МНК) в *AnyLogic*. Как видно из рис. 3, программная реализация диаграммы действий осуществляется непо-

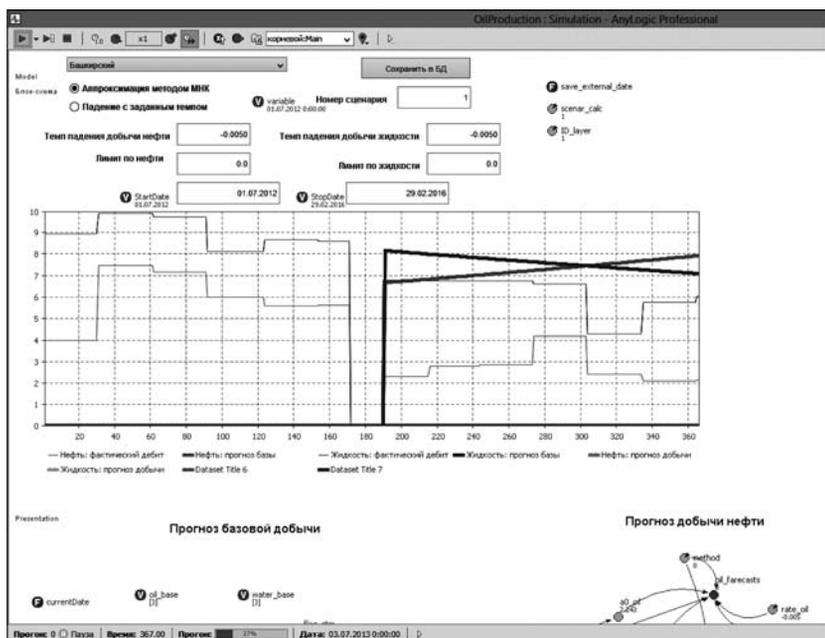


Рис. 1. Имитационная модель добычи нефти в *AnyLogic*

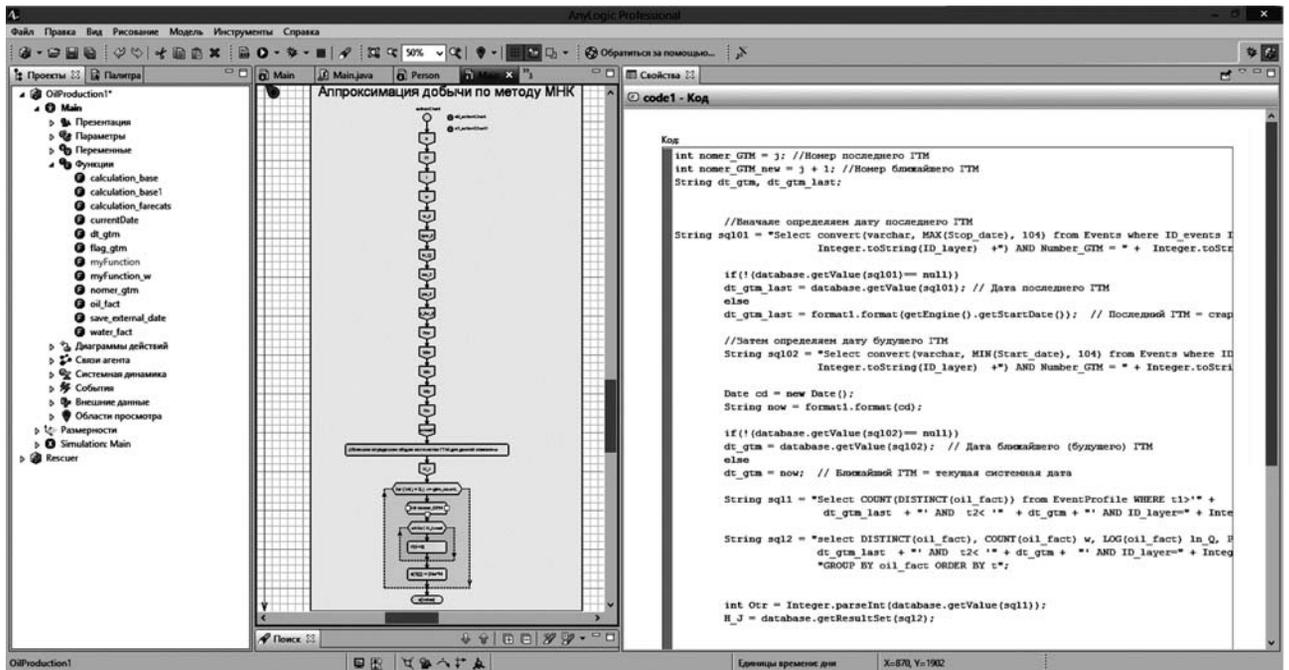


Рис. 2. Программная реализация имитационной модели в AnyLogic

средственно в элементах UML-диаграммы. Подобный подход обеспечивает прозрачность всех расчетных алгоритмов, а также существенно облегчает масштабируемость реализуемых методов прогнозирования и оптимизации.

Другой важной особенностью AnyLogic является интегрируемость с различными базами данных посредством технологии JDBC.

Разработанная система включает спроектированную базу данных на платформе MS SQL Server (рис. 4), интегрированную с имитационной моделью.

База данных системы состоит из ряда таблиц, основными из которых являются таблица Profile, содержащая данные о характеристиках скважины (наименование пласта, номер скважины и др.); таблица Forecasts, содержащая результаты прогнозирования добычи нефти и жидкости (по дням); таблица Profile-Event, содержащая номера ГТМ, относящихся к конкретным скважинам; таблица Events, содержащая характеристики соответствующих ГТМ, в частности, время начала и окончания мероприятий; таблица EventProfile, содержащая данные о фактической (исторической) добыче, прогноз базовой добычи нефти и жидкости с учетом ГТМ и др.

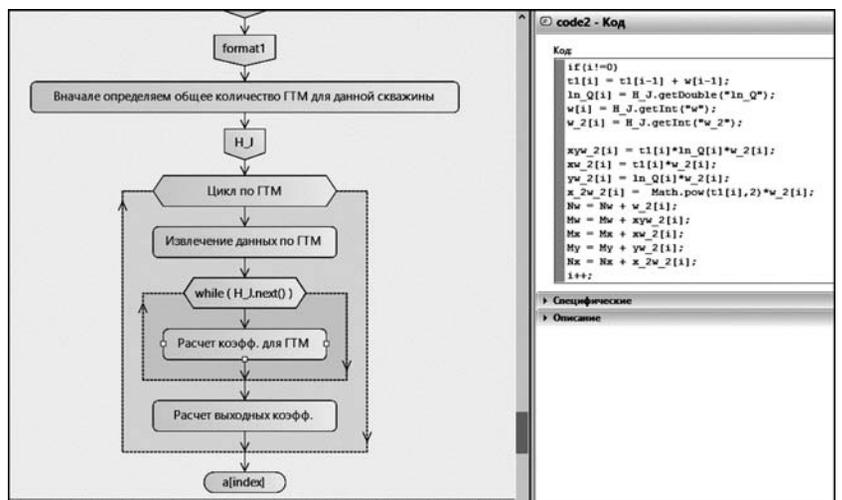


Рис. 3. Фрагмент реализации метода МНК в AnyLogic

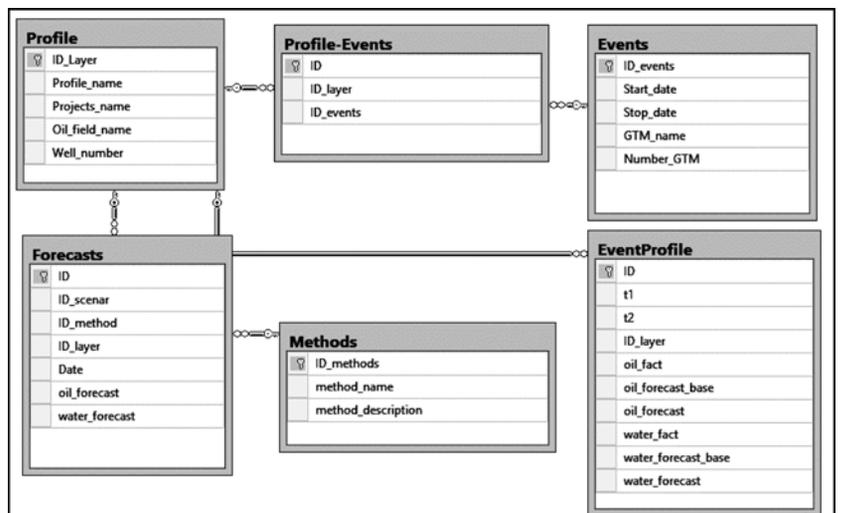


Рис. 4. Фрагмент базы данных системы

Далее с использованием технологий веб-сервисов (SOA), в частности JAX-WS (Java API for XML Web Services), был разработан веб-интерфейс системы прогнозирования добычи (рис. 5).

Веб-сервис обеспечивает возможность программного управления запуском и выполнением всех вычислительных процедур для выбранной пользователем скважины, выбранного метода прогнозирования со своими параметрами и на заданном временном интервале. Ключевой фрагмент Java-кода разработанного веб-сервиса, обеспечивающего программный запуск модели *AnyLogic*, представлен ниже.

```
Simulation s = new oilproduction.Simulation();
Main m = new Main(s.getEngine(), null, null);
m.setParametersToDefaultValues();
s.getEngine().start(m);
//Устанавливаем значения управляющих параметров модели
m.method = method;// Выбранный метод прогнозирования добычи
m.ID_layer = ID_layer; // Выбранный пласт (скважина)
m.scenar_calc = scenar_calc; // Номер сценария
m.rate_oil = rate_oil; // Темп падения добычи нефти
m.rate_water = rate_water; // Темп падения добычи жидкости
m.limit_oil = limit_oil; // Предельный уровень падения добычи нефти
m.limit_oil = limit_water; // Предельный уровень падения добычи жидкости
//Запуск модели на выбранном диапазоне расчета
while(s.getEngine().date().before(s.getEngine().getStopDate()))
{
s.getEngine().step(); //Запускаем новый поток (пошаговый прогон модели)
}
// Сохранение в базу данных результатов расчетов в AnyLogic
```

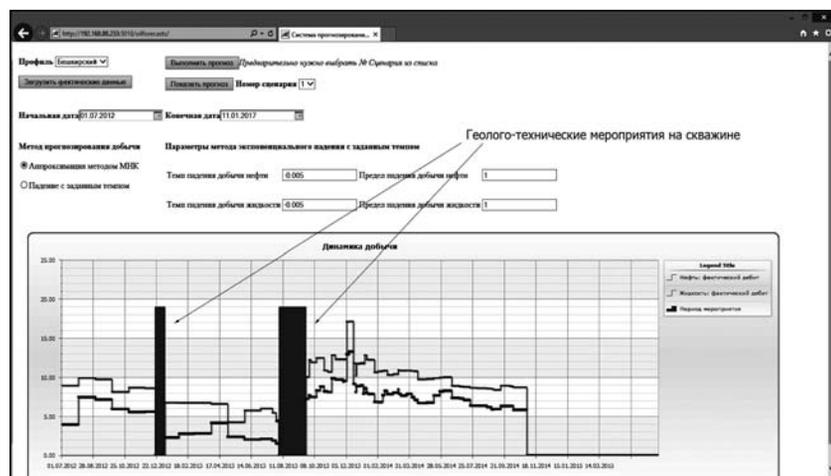


Рис. 5. Веб-интерфейс системы

```
m.save_external_date(ID_layer, (method + 1), scenar_calc); //Данный метод в AnyLogic был объявлен как public
s.getEngine().stop(); // Уничтожение модели (освобождение памяти)
return result;
```

Рассматриваемый веб-сервис, обеспечивающий программный запуск имитационной модели и управление ходом вычислительной процедуры, периодически взаимодействует с другим веб-сервисом, обеспечивающим мониторинг результатов прогнозирования и формирующим внешнее управление для основного сервиса (через интерфейс администратора). Например, при определенном значении одного из имеющихся управляющих параметров системы выполнение прогнозов по данной скважине приостанавливается (с высвобождением соответствующих вычислительных ресурсов).

Подобное веб-приложение, основанное на асинхронных веб-сервисах, может быть развернуто на различных веб-серверах, в том числе *Apache Tomcat*, *IBM WebSphere*, *JBoss* и других, и может использовать кластерную архитектуру для организации прогнозирования добычи нефти по скважинам в режиме параллельных вычислений с возможностью внешнего управления ходом выполнения подобных вычислительных процедур (например, остановка и возобновление расчетов, инициация автоматического запуска прогнозной модели из хранимых процедур и триггеров базы данных системы, приоритизация расчетов при высокой нагрузке на сервер и др.).

Заключение

В настоящей работе представлен оригинальный программный комплекс, разработанный с использованием методов имитационного моделирования, технологий баз данных, веб-сервисов (SOA) и предназначенный для сценарного прогнозирования динамики добычи нефти по скважинам. Прототип системы был успешно внедрен в крупнейшей российской нефтяной компании и используется для прогнозирования дополнительной добычи нефти по всему пулу имеющихся скважин (более 100 000 скважин на десятилетнем временном интервале), а также при оценке потенциального перераспределения ресурсов между скважинами.

Важным преимуществом предлагаемого подхода является использование системы имитационного моделирования *AnyLogic* для реализации и визуализации сложных вычислительных процедур, в частности методов параметри-

ческой аппроксимации, обеспечение бесшовной интеграции прогнозной модели с базой данных системы, использование технологии асинхронных веб-сервисов для обеспечения эффективного управления ходом выполнения множественных расчетов в режиме реального времени.

Разработанная система может быть масштабирована за счет включения новых методов прогнозирования и оптимизации добычи и расширения пользовательского веб-интерфейса.

Список литературы

1. Dupuit J. Etudes Théoriques et Pratiques sur le mouvement des Eaux dans les canaux découverts et á travers les terrains perméables. Paris: Dunod, 1863. 304 p.
2. Акопов А. С. Проблемы управления субъектом ТЭК в современных условиях. М.: ЦЭМИ РАН, 2004.
3. Акопов А. С. Использование средств динамического имитационного моделирования для подготовки управленческих решений в ТЭК // Системы управления и информационные технологии. 2004. № 2. С. 72—77.

4. Акопов А. С., Бекларян Г. Л. Интеллектуальные гибридные системы управления вертикально-интегрированными организационными структурами / ЦЭМИ РАН. Серия ISBN 978-5-8211-0526-4 Препринт #WP/2009/267. 2009.

5. Акопов А. С. К вопросу проектирования интеллектуальных систем управления сложными организационными структурами. Ч. 2. Программная реализация системы управления инвестиционной деятельностью вертикально-интегрированной нефтяной компании // Проблемы управления. 2011. № 1. С. 47—54.

6. Алексеев Ю. В., Николаенко В. А. Динамическое имитационное моделирование в стратегическом и оперативном планировании добычи // Газовая промышленность. 2005. № 5. С. 51—53.

7. Лоран П. Ж. Аппроксимация и оптимизация. М.: Мир, 1975. 496 с.

8. Дейк Л. П. Основы разработки нефтяных и газовых месторождений. М.: Премиум Инжиниринг, 2009. 576 с.

9. Акопов А. С. Имитационное моделирование: учебник и практикум для академического бакалавриата. М.: Юрайт, 2014. 389 с.

10. Акопов А. S. Parallel genetic algorithm with fading selection // International Journal of Computer Applications in Technology. 2014. Vol. 49. N. 3—4. P. 325—331.

11. Акопов А. S. Designing of integrated system-dynamics models for an oil company // International Journal of Computer Applications in Technology. 2012. Vol. 45, N 4. P. 220—230.

A. S. Akopov, Professor, e-mail: aakopov@hse.ru, A. L. Beklaryan, Lecturer, e-mail: abeklaryan@hse.ru,
N. K. Khachatryan, Associate Professor, e-mail: nkachatryan@hse.ru,
A. V. Fomin, Associate Professor, e-mail: afomin@hse.ru

National Research University Higher School of Economics, Department of Business Analytics, Moscow

The Forecasting System Dynamics of Oil Production using Simulation Modeling

This article presents an original information-analytical system developed using parametric approximation and simulation techniques and designed for scenario forecasting the dynamics of oil production in wells. This system is implemented as a software, the core of which is a simulation model developed in an IDE AnyLogic, integrated with a database and describes the dynamics of production at each well, taking into account implemented and planned geological and technical measures (GTM). A prototype system has been successfully implemented in Russia's largest oil company and used to predict the incremental oil production around the pool of all existing wells (more than 100,000 wells in the ten-year time frame), as well as assessing the potential effect of GTM for subsequent reallocation of resources between wells.

Keywords: simulation modeling, parametric approximation, oil production forecast, service-oriented solution, AnyLogic

References

1. Dupuit J. Etudes Théoriques et Pratiques sur le mouvement des Eaux dans les canaux découverts et á travers les terrains perméables. Paris: Dunod, 1863. 304 p.
2. Акопов А. С. *Problemy upravleniya sub#ektom TJeK v sovremennyh usloviyah* (Energy management problems in modern conditions), CEMI RAN, Moscow, 2004.
3. Акопов А. S. Ispol'zovanie sredstv dinamicheskogo imitacionnogo modelirovaniya dlja podgotovki upravlencheskih reshenij v TJeK (Using of Dynamic Simulation Models for Decision-Making Support in Oil and Gas Industry), *Sistemy upravleniya i informacionnye tehnologii*, 2004, no. 2, pp. 72—77.
4. Акопов А. S., Beklaryan G. L. *Intellektual'nye gibridnye sistemy upravleniya vertikal'no-integrirovannymi organizacionnymi strukturami (Intelligent hybrid control system for vertically integrated organizational structures)* / CEMI RAN. Seriya ISBN 978-5-8211-0526-4 "Preprint #WP/2009/267". 2009.
5. Акопов А. S. K voprosu proektirovaniya intellektual'nyh sistem upravleniya slozhnyimi organizacionnymi strukturami. Ch2. Programmnaja realizacija sistemy upravleniya investicionnoj dejatel'nost'ju vertikal'no-integrirovannoj nefjtanoj kompanii (On the issue of developing of intelligent control systems of complex organizational

structures (ii). Software support for control system of the vertically integrated oil company investment activities), *Problemy upravleniya*, 2011, no. 1, pp. 47—54.

6. Алексеев Ю. В., Николаенко В. А. Dinamicheskoe imitacionnoe modelirovanie v strategicheskom i operativnom planirovanii dobychi (Dynamic simulation in strategic and operational planning of production), *Gazovaja promyshlennost'*, 2005, no. 5, pp. 51—53.

7. Лоран П. Ш. *Approksimacija i optimizacija* (Approximation and Optimization), Moscow, Mir, 1975, 496 p.

8. Дейк Л. П. *Osnovy razrabotki nefjtjanyh i gazovyh mestorozhdenij* (Fundamentals of Reservoir Engineering). Moscow, Premium Inzhiniring, 2009, 576 p.

9. Акопов А. S. *Imitacionnoe modelirovanie: uchebnik i praktikum dlja akademicheskogo bakalavriata* (Simulation modeling: a textbook and a workshop for the academic Bachelor). Moscow, Jurajt, 2014, 389 p.

10. Акопов А. S. Parallel genetic algorithm with fading selection, *International Journal of Computer Applications in Technology*, 2014, vol. 49, no. 3—4, pp. 325—331.

11. Акопов А. S. Designing of integrated system-dynamics models for an oil company, *International Journal of Computer Applications in Technology*, 2012, vol. 45, no. 4, pp. 220—230.

Г. Г. Булычев, д-р физ.-мат. наук, проф., e-mail: geo-bulychev@mail.ru,
Московский технологический университет (МИРЭА)

Моделирование динамики полого цилиндра методом пространственных характеристик

Предлагается новая методика построения алгоритмов для характеристической формы уравнений динамики деформируемых твердых тел. Эта методика используется для построения вычислительных схем динамики полого изотропного однородного линейно-упругого цилиндра в трехмерной постановке. Полученная таким образом программа проверяется на работоспособность и тестируется. Анализируются полученные результаты и возможности дальнейшего совершенствования и применения предлагаемой методики.

Ключевые слова: метод пространственных характеристик, численное моделирование, динамика конструкций, алгоритмы

Введение

Динамике цилиндрических оболочек посвящено большое число статей и монографий, достаточно подробный обзор которых можно найти в [1]. С точки зрения используемых методов их можно разделить на три группы: статьи, в которых используются численные методы; статьи, в которых используются численно-аналитические методы, и, наконец, статьи, в которых используется в основном аналитический аппарат.

В последние 30 лет в связи в бурным ростом вычислительной техники для решения вышеуказанных задач в подавляющем большинстве используются численные методы: методы конечных разностей (МКР) [2], методы конечного элемента (МКЭ) [3] и наиболее популярные в настоящее время методы граничного элемента (МГЭ) [4]. Достоинствами этих методов является их универсальность, а недостатки их различны. К недостаткам МКР можно отнести то, что при решении гиперболических уравнений и сеточные, и сеточно-характеристические схемы дают "вычислительную рябь" [2], затрудняющую вычисления и часто приводящую к неустойчивости счета. К недостаткам МКЭ и МГЭ [3, 4], применительно к задачам гиперболического типа, следует отнести: повышенные требования к граничным условиям (не при всех типах граничных условий построение алгоритмов возможно, граничные условия должны быть гладкими и т. д.), огромный объем вычислений, связанный с вычислением напряженно-деформированного состояния во всех точках пространственной сетки, формирование и решение алгебраических уравнений очень высокого порядка (для МКЭ) или решение интегральных уравнений Фредгольма второго рода (для МГЭ).

Численно-аналитические методы используются в основном для уменьшения количества вычислений, главным образом для уменьшения размер-

ности задачи. Например, трехмерную неосесимметричную задачу динамики цилиндров путем разложения решения в конечные синус или косинус ряды Фурье по азимутальной координате можно свести к N двумерным задачам, зависящим от номера гармоники n ($1 \leq n \leq N$), интегральное преобразование Лапласа по времени сводит задачу динамики к задаче статики с параметром и т. д. Подобных преобразований существует множество, но все они не универсальны и далеко не всегда ведут к реальному упрощению задачи.

Аналитические решения неосесимметричных задач динамики полых цилиндров были получены в основном тогда, когда возможности вычислительных машин были весьма ограничены и вычислительные схемы годились лишь для грубых расчетов (конечные разности на грубых сетках, суммирование рядов, определение собственных значений трансцендентных уравнений).

Достаточно подробный обзор аналитических методов решения динамических задач и использование этих методов содержится в монографии [5].

Одна из подобных задач была решена автором совместно с С. Г. Пшеничным [6]. В ней рассматривалась волновая динамика полого, трехслойного, линейно-упругого цилиндра, бесконечного в осевом направлении, при неосесимметричной нагрузке. Для решения задачи использовался целый комплекс аналитических методов: приведение к векторному и скалярному потенциалам, Фурье-разложение по азимутальной координате, разделение переменных, преобразование Лапласа по времени, преобразование Бесселя по радиальной координате, решение ряда систем трансцендентных уравнений (естественно, численное), преобразование Меллина и суммирование двойных рядов (тоже численное).

Использованный в данной работе метод пространственных характеристик принадлежит, вообще говоря, к численно-аналитическим методам, поскольку характеристики могут быть сведены к

производным по направлению и размерность задачи для каждого характеристического уравнения может быть понижена на единицу. Поскольку характеристическая форма уравнений динамики была получена автором для весьма широкого класса сред и в любой ортогональной криволинейной системе координат [7], в данной работе будут рассматриваться только численные аспекты применения этого метода к трехмерной задаче динамики полого, изотропного линейно-упругого цилиндра.

Подробно достоинства и недостатки метода пространственных характеристик рассматривались в работе [8]. В частности, там указывалось, что 1) дифференциальные части характеристических уравнений имеют дивергентный вид и 2) построение вычислительных шаблонов оказывается затруднительным для тел, геометрия которых предполагает криволинейную сетку. Эти два положения и обсуждаются в данной статье.

Разработка алгоритмов

Рассмотрим классический способ построения конечно-разностных соотношений по шаблонам на примере двумерной задачи динамики с переменными t , x_1 и x_2 . Покажем, как строятся конечно-разностные уравнения для расчета напряжений и скоростей частиц в продольной волне, распространяющейся в обе стороны вдоль координатных осей x_1 и x_2 . Пусть рассматриваемое тело изотропно, однородно, а его материал линейно упругий. Соответствующие характеристические уравнения и их вывод можно посмотреть, например, в работе [9], где рассматривается процесс разрушения тонкой пластинки. После приведения к безразмерному виду эти уравнения приобретают следующий вид:

$$\left(\frac{\partial}{\partial \tau} \pm \frac{\partial}{\partial \bar{x}_1}\right) (\bar{\sigma}_{11} \mp \bar{V}_1) \pm \frac{\partial}{\partial \bar{x}_2} (\bar{\sigma}_{12} \mp v \bar{V}_2) = 0; \quad (1)$$

$$\left(\frac{\partial}{\partial \tau} \pm \frac{\partial}{\partial \bar{x}_2}\right) (\bar{\sigma}_{22} \mp \bar{V}_2) \pm \frac{\partial}{\partial \bar{x}_1} (\bar{\sigma}_{12} \mp v \bar{V}_1) = 0;$$

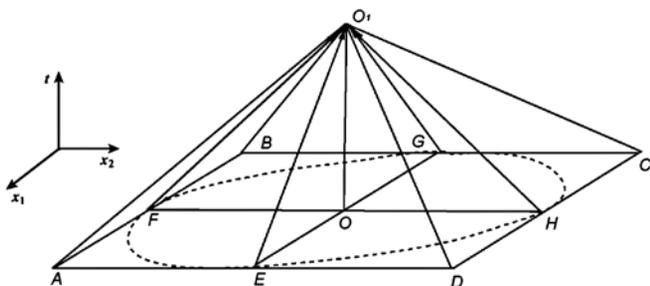


Рис. 1

где $\tau = c_0 t/x_0$, $\bar{x}_i = x_i/x_0$; $\bar{\sigma}_{ij} = \sigma_{ij}/(\rho c_0^2)$; $\bar{V}_i = V_i/c_0$; $i, j = 1, 2$; σ_{ij} — напряжения; V_i — скорости частиц; v — коэффициент Пуассона; ρ — плотность материала тела; c_0 — скорость продольных волн; x_0 — характерный размер.

Шаблон для построения конечно-разностных соотношений имеет вид, изображенный на рис. 1.

Здесь конус $EFGHO_1$ — область, из которой волны приходят в рассматриваемую точку O_1 , пирамида $ABCD O_1$ аппроксимирует этот конус и является основой для построения конечно-разностных соотношений. Отрезки $O_1 E$, $O_1 F$, $O_1 G$ и $O_1 H$, по которым конус и пирамида соприкасаются друг с другом, соответствуют характеристикам уравнений (1). Действительно, характеристические операторы этих уравнений можно представить в виде: $\frac{\partial}{\partial \tau} \pm \frac{\partial}{\partial x_i} = \sqrt{2} \frac{\partial}{\partial l_{1,2}^{(i)}}$,

где $\frac{\partial}{\partial l}$ — производные по направлению от основания к вершине конуса, $i = 1, 2$; A, B, C, D — узлы пространственной сетки.

Аналогично строятся шаблоны для поперечных волн, для неподвижных разрывов вырожденной характеристикой является высота OO_1 . Все аппроксимации проводятся относительно узлов A, B, C, D пространственной сетки.

Указанное построение показывает, что для изотропных тел пространственная сетка должна быть квадратной, а для анизотропных — прямоугольной (в основании конуса будет лежать эллипс). Что же будет в случае криволинейной координатной сетки? Рассмотрим рис. 2, на котором показан этот случай.

Здесь круги — это основания конусов продольных волн с центрами O, O_1 и O_2 , $AA_3 B_3 B$ — часть кругового сектора, в которую требуется вписать эти круги. Видно, что в $ABB_1 A_1$ круг вписать не удастся, в $A_1 B_1 B_2 A_2$ — удастся, а часть сектора $A_2 B_2 B_3 A_3$ — слишком широка, для того чтобы вписать в нее круг. При моделировании оказывается, что в первом случае алгоритмы демонстрируют неустойчивость: степенное нарастание, пропорциональное a^n ,

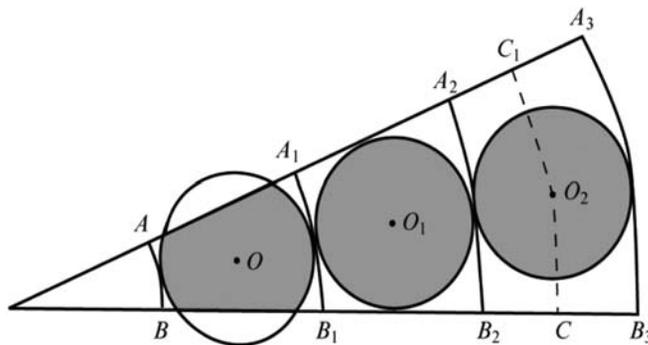


Рис. 2

где a — отношение площадей круга и сектора, n — номер шага по времени; в третьем случае происходит затухание волн по тому же закону и только во втором случае моделирование по описанным выше шаблонам оказывается удачным.

Поэтому в настоящее время для полых цилиндров метод пространственных характеристик используется только тогда, когда $\Delta r/r_0 \leq 0,01$, где Δr — толщина цилиндра, r_0 — его внутренний радиус, т. е. когда ячейки пространственной сетки почти квадратные.

Использование формул Гаусса—Остроградского

При указанном выше подходе не учитывается дивергентный вид построенных уравнений. А для аппроксимации таких уравнений в трехмерном случае можно, а в случае криволинейной пространственной сетки и необходимо, использовать интегральную теорему Гаусса — Остроградского, согласно которой

$$\text{div} A dV = \oint\!\!\!\oint (A \cdot n) dS, \quad (2)$$

где A — некоторая векторная функция, изменяющаяся в объеме V , ограниченном поверхностью S . В словесной формулировке это означает, что изменение векторной функции в замкнутом объеме равно потоку этой функции через границы объема. С точки зрения нашей задачи это означает изменение вычислительных шаблонов согласно рис. 3, где отрезок OO соответствует координатной линии CC_1 на рис. 2, все отрезки OO_1 равны OO_1 на рис. 1, а наклонные AO_1 и BO_1 равны и одинаково направлены с отрезками GO_1 и EO_1 на рис. 1. Очевидно, что треугольник AO_1B в центре рисунка соответствует конусу волн, изображенному на рис. 1 (ось x_1 совпадает с выбранным фрагментом оси $r\varphi$), отрезки AO_1 и BO_1 , направленные к границам ячейки, соответствуют потокам, идущим к этим границам.

Таким образом, вычислительная схема, представленная формулой (2), предполагает вычисление потоков всех видов волн (продольных и поперечных) через границы ячеек, образованных производной по направлению (или осью t для вы-

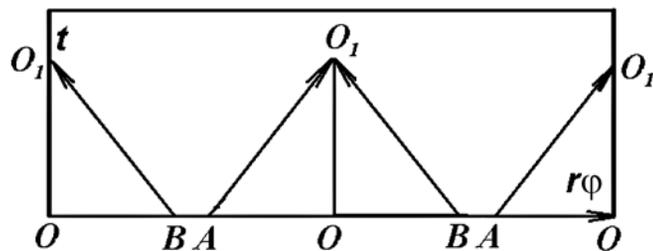


Рис. 3

рожденных характеристик) и оставшимися одной (для двумерных задач) или двумя (для трехмерных задач) пространственными переменными. Очевидно, что в случае $OO = 2OO_1$, вычислительные схемы соответствующие формулам (1) и (2), совпадают с точностью до тех геометрических членов, которые входят в состав дивергенции. Это различие обусловлено учетом (как в данной схеме) или не учетом (как при использовании сопутствующей декартовой системы координат) кривизны координатных линий при построении конечно-разностных соотношений.

Моделирование динамики полого цилиндра

Проведем моделирование динамики изотропного линейно-упругого полого цилиндра, используя уравнения (2) для построения алгоритмов. Исходная математическая модель в цилиндрических координатах $\{r, \theta, z\}$ имеет следующий вид:

Уравнения движения

$$\begin{aligned} \frac{\partial \sigma_{rr}}{\partial r} + \frac{1}{r} \frac{\partial \sigma_{r\theta}}{\partial \theta} + \frac{\partial \sigma_{rz}}{\partial z} + \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r} &= \rho \frac{\partial V_r}{\partial t}, \\ \frac{\partial \sigma_{r\theta}}{\partial r} + \frac{1}{r} \frac{\partial \sigma_{\theta\theta}}{\partial \theta} + \frac{\partial \sigma_{\theta z}}{\partial z} + \frac{2}{r} \sigma_{r\theta} &= \rho \frac{\partial V_\theta}{\partial t}, \\ \frac{\partial \sigma_{rz}}{\partial r} + \frac{1}{r} \frac{\partial \sigma_{\theta z}}{\partial \theta} + \frac{\partial \sigma_{zz}}{\partial z} + \frac{\sigma_{rz}}{r} &= \rho \frac{\partial V_z}{\partial t}. \end{aligned} \quad (3)$$

Закон Гука

$$\begin{aligned} \sigma_{rr} &= 2\mu \varepsilon_{rr} + \lambda e, \quad \sigma_{\theta\theta} = 2\mu \varepsilon_{\theta\theta} + \lambda e, \quad \sigma_{zz} = 2\mu \varepsilon_{zz} + \lambda e, \\ \sigma_{r\theta} &= 2\mu \varepsilon_{r\theta}, \quad \sigma_{\theta z} = 2\mu \varepsilon_{\theta z}, \quad \sigma_{rz} = 2\mu \varepsilon_{rz}, \\ e &= \varepsilon_{rr} + \varepsilon_{\theta\theta} + \varepsilon_{zz}, \end{aligned} \quad (4)$$

где μ и λ — параметры Ламе.

Соотношения между деформациями ε_{ij} и перемещениями u_i

$$\begin{aligned} \varepsilon_{rr} &= \frac{\partial u_r}{\partial r}, \quad \varepsilon_{\theta\theta} = \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r}, \quad \varepsilon_{zz} = \frac{\partial u_z}{\partial z}, \\ \varepsilon_{rz} &= \frac{1}{2} \left(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z} \right), \quad \varepsilon_{r\theta} = \frac{1}{2} \left(\frac{1}{r} \frac{\partial u_r}{\partial \theta} + \frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r} \right), \\ \varepsilon_{\theta z} &= \frac{1}{2} \left(\frac{1}{r} \frac{\partial u_z}{\partial \theta} + \frac{\partial u_\theta}{\partial z} \right). \end{aligned} \quad (5)$$

Выражения для скоростей частиц в лагранжевой системе координат

$$V_r = \frac{\partial u_r}{\partial t}, \quad V_\theta = \frac{\partial u_\theta}{\partial t}, \quad V_z = \frac{\partial u_z}{\partial t}. \quad (6)$$

Уравнения (3)—(6) позволяют создать две группы уравнений: уравнения движения и определяющие уравнения относительно двух групп неизвестных — напряжений и скоростей частиц тела.

Приведение этих групп уравнений к единой (матричной) форме и построение ее характеристической формы с помощью тождественных преобразований было сделано в работе [7] для любых анизотропных упруговязкопластических тел с симметричной матрицей жесткости и в произвольной криволинейной ортогональной системе координат и для любого заданного направления.

В частности, для нашей системы после приведения к безразмерному виду (в качестве параметра x_0 выбирается внутренний радиус цилиндра r_0) характеристические уравнения в направлении координатной линии $r\theta$ будут иметь следующий вид:

$$\begin{aligned} & \left(\frac{\partial}{\partial \tau} - \frac{n_\theta}{\bar{r}} \frac{\partial}{\partial \eta} \right) V_\theta + n_\theta \sigma_{\theta\theta} = \\ & = \frac{\partial}{\partial \bar{z}} \sigma_{\theta z} + n_\theta v_1 V_z + \frac{\partial}{\partial \bar{r}} \sigma_{\theta r} + n_\theta v_1 V_r + \frac{2\sigma_{r\theta} + n_\theta V_r}{\bar{r}}, \\ & \left(\frac{\partial}{\partial \tau} - n_\theta \frac{\gamma}{\bar{r}} \frac{\partial}{\partial \eta} \right) V_z + n_\theta \sigma_{\theta z} / \gamma = \\ & = \frac{\partial}{\partial \bar{z}} \sigma_{zz} + n_\theta \gamma V_\theta + \frac{\partial \sigma_{rz}}{\partial \bar{r}} + \frac{\sigma_{rz}}{\bar{r}}, \\ & \left(\frac{\partial}{\partial \tau} - n_\theta \frac{\gamma}{\bar{r}} \frac{\partial}{\partial \eta} \right) V_r + n_\theta \sigma_{\theta r} / \gamma = \\ & = \frac{\partial \sigma_{rz}}{\partial \bar{z}} + \frac{\partial}{\partial \bar{r}} (\sigma_{rr} + n_\theta \gamma V_\theta) + \frac{\sigma_{rr} - \sigma_{\theta\theta} - n_\theta \gamma V_\theta}{\bar{r}}, \\ & \frac{\partial}{\partial \tau} \sigma_{rr} - v_1 \sigma_{\theta\theta} = v_3 \left(\frac{\partial V_r}{\partial \bar{r}} + v \frac{\partial V_z}{\partial \bar{z}} \right), \\ & \frac{\partial}{\partial \tau} \sigma_{zz} - v_1 \sigma_{\theta\theta} = v_3 \left(\frac{\partial V_z}{\partial \bar{z}} + v \frac{\partial V_r}{\partial \bar{r}} \right), \\ & \frac{\partial \sigma_{rz}}{\partial \tau} = v_4 \left(\frac{\partial V_z}{\partial \bar{r}} + v \frac{\partial V_r}{\partial \bar{z}} \right), \end{aligned} \quad (7)$$

где $\eta = r_0 \theta$, $\bar{r} = r/r_0$, $\bar{z} = z/r_0$, $v_1 = v/(1-v)$, $v_3 = (1-2v)/(1-v)^2$, $v_4 = 0,5(1-2v)/(1-v)$, $\gamma = c_\perp/c_0 = \sqrt{(1-v_1)/2}$, c_\perp — скорость поперечных волн в цилиндре, $n_\theta = +1$, когда направление движения волн совпадает с направлением возрастания θ координаты, и $n_\theta = -1$ для волны противоположного направления.

Анализ уравнений (7) показывает, что три последних уравнения соответствуют неподвижным разрывам и в коррекции не нуждаются. В первых трех уравнениях заменим характеристики производными по направлениям

$$\begin{aligned} & \left(\frac{\partial}{\partial \tau} - \frac{n_\theta}{\bar{r}} \frac{\partial}{\partial \eta} \right) = \sqrt{1 + \bar{r}^{-2}} \frac{\partial}{\partial l_{1,2}}, \\ & \left(\frac{\partial}{\partial \tau} - n_\theta \frac{\gamma}{\bar{r}} \frac{\partial}{\partial \eta} \right) = \sqrt{1 + \gamma^2 \bar{r}^{-2}} \frac{\partial}{\partial l_{3,4}}, \end{aligned} \quad (8)$$

где четные индексы у l соответствуют $n_\theta = +1$, а нечетные — $n_\theta = -1$, и запишем первые три уравнения (7) в виде

$$\begin{aligned} & \frac{\partial}{\partial l_{1,2}} \{ \sqrt{1 + \bar{r}^{-2}} V_\theta + n_\theta \sigma_{\theta\theta} \} - \frac{\partial}{\partial \bar{z}} \sigma_{\theta z} + n_\theta v_1 V_z - \\ & - \frac{1}{\bar{r}} \frac{\partial}{\partial \bar{r}} \bar{r} \sigma_{\theta r} + n_\theta v_1 V_r = \frac{\sigma_{r\theta} + n_\theta v_2 V_r}{\bar{r}}, \quad v_2 = 1 - v_1, \\ & \frac{\partial}{\partial l_{3,4}} \{ \sqrt{1 + \gamma^2 \bar{r}^{-2}} V_z + n_\theta \sigma_{\theta z} / \gamma \} - \\ & - \frac{\partial}{\partial \bar{z}} \sigma_{zz} + n_\theta \gamma V_\theta - \frac{1}{\bar{r}} \frac{\partial \bar{r} \sigma_{rz}}{\partial \bar{r}} = 0, \\ & \frac{\partial}{\partial l_{3,4}} \{ \sqrt{1 + \gamma^2 \bar{r}^{-2}} V_r + n_\theta \sigma_{\theta r} / \gamma \} - \frac{\partial \sigma_{rz}}{\partial \bar{z}} - \\ & - \frac{1}{\bar{r}} \frac{\partial}{\partial \bar{r}} \bar{r} \sigma_{rr} + n_\theta \gamma V_\theta = - \frac{\sigma_{\theta\theta} + 2n_\theta \gamma V_\theta}{\bar{r}}, \end{aligned} \quad (9)$$

где в левых частях записана дивергенция векторов $\Psi_{1,2}^{(i)}$, где $i = 1, 2, 3$, в координатах $\{l_{1,2}, \bar{z}, \bar{r}\}$ и $\{l_{3,4}, \bar{z}, \bar{r}\}$ соответственно. Эти векторы имеют следующие компоненты:

$$\begin{aligned} & \Psi_{1,2}^{(1)} = \{ \sqrt{1 + \bar{r}^{-2}} (V_\theta + n_\theta \sigma_{\theta\theta}), \\ & - (\sigma_{\theta z} + n_\theta v_1 V_z), -(\sigma_{\theta r} + n_\theta v_1 V_r) \}; \\ & \Psi_{1,2}^{(2)} = \{ \sqrt{1 + \gamma^2 \bar{r}^{-2}} (V_z + n_\theta \sigma_{\theta z} / \gamma), \\ & - (\sigma_{zz} + n_\theta \gamma V_\theta), -\sigma_{rz} \}; \\ & \Psi_{1,2}^{(3)} = \{ \sqrt{1 + \gamma^2 \bar{r}^{-2}} (V_r + n_\theta \sigma_{\theta r} / \gamma), \\ & - \sigma_{rz} - (\sigma_{rr} + n_\theta \gamma V_\theta) \}. \end{aligned} \quad (10)$$

В правых частях уравнений (9) находятся так называемые "геометрические" члены, содержащие определяемые переменные, деленные на радиус и не вошедшие в дивергенции. Поэтому построение вычислительных алгоритмов должно включать как использование формулы (2) для вычисления левых частей, так и итеративный процесс (схема предиктор-корректор) для вычисления правых частей. Для аппроксимации производных по направлению используются односторонние разности, для остальных производных — центральные; при этом (при первом порядке аппроксимации) остаточные члены дают уравнения параболического типа и погрешности счета со временем затухают.

Тестирование вычислительных схем

Предлагаемая методика использовалась для построения вычислительных схем в программе моделирования трехмерной динамики толстого ($\Delta r/r_0 = 1$) цилиндра, что при использовании старого метода

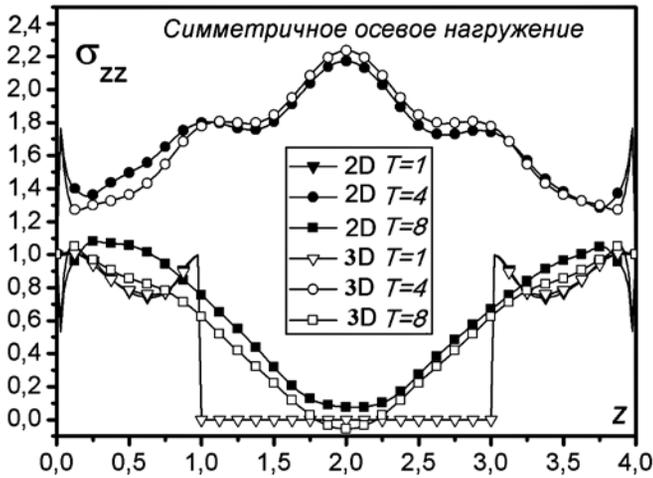


Рис. 4

аппроксимации давало экспоненциальную расходимость решения. Для сравнения использовались построенные ранее и хорошо зарекомендовавшие себя двумерные $\{t, z, r\}$ характеристические схемы для полых цилиндров. Нагрузки подбирались таким образом, чтобы решение задачи не зависело от окружной координаты. Время моделирования менялось в широких пределах: от единиц до сотни пробегов продольной волны по толщине цилиндра. При больших временах проверялось поведение алгоритмов в переходных процессах и выход решения на статику.

Тест 1. Растягивающая нагрузка приложена симметрично к торцевым граням цилиндра. Величина нагрузки равна единице. Остальные поверхности цилиндра свободны от напряжений. Нагрузка имеет вид единичного скачка напряжения, приложенного в начальный момент, и не меняется в процессе моделирования. Длина цилиндра $z_m = 4$, внутренний радиус цилиндра $r_0 = 1$, толщина цилиндра $\Delta r = 1$, коэффициент Пуассона $\nu = 0,3$, время моделирования — до 100 времен пробегов по толщине цилиндра. Деление цилиндра по толщине: 20 ячеек. Результаты моделирования показаны на рис. 4.

Здесь по оси абсцисс отложены осевые координаты цилиндра, по оси ординат — осевые напряжения, определяемые на линии (z, r_c) , где $r_c = r_0 + \Delta r/2$ для двумерной задачи и на линии (z, r_c, π) — для трехмерной. Графики с черными маркерами получены при использовании тестовых (2D) алгоритмов, а со светлыми — тестируемых (3D) для различных моментов времени. Соответствие графиков моментам времени указано в табличках на рисунке. Всюду наблюдается хорошее совпадение. Любопытно, что при больших временах пробег $T = 100$ исследуемые алгоритмы стремятся к статическому решению $\sigma_{zz} = 1$ даже быстрее, чем контрольные.

Тест 2. Нагружение цилиндра внутренней сжимающей нагрузкой. Нагрузка имеет вид единичного

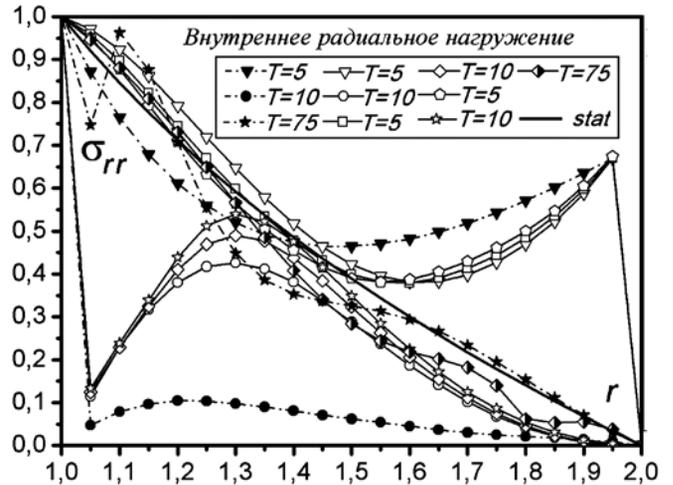


Рис. 5

скачка нормального напряжения, одинакового по всей нагружаемой поверхности и не меняется в процессе моделирования. Остальные поверхности цилиндра свободны от напряжений. Размеры цилиндра и коэффициент Пуассона те же, что и в тесте 1. Результаты тестирования приведены на рис. 5.

На нем показано распределение радиальных напряжений по толщине цилиндра в указанные на рисунке моменты времени. Как и ранее, графики с черными маркерами построены с использованием тестовых алгоритмов, а со светлыми — с помощью исследуемых. Кроме того, сплошной линией на рисунке показано статическое решение той же задачи, полученное аналитически, в предположении, что длина цилиндра бесконечно большая. Графики со светлыми маркерами разбиты на три группы, соответствующие делению цилиндра по радиусу на 20, 40 и 60 ячеек. Это позволяет оценить сходимость алгоритмов при уменьшении шага сетки. График с ромбовидным черно-светлым маркером показывает сходимость алгоритмов трехмерной задачи к статическому решению. При моделировании выбиралось $z = z_{\max}/2$, $\theta = \pi/2$. Анализ рис. 5 показывает:

- хорошее соответствие двумерных и трехмерных решений для всех рассмотренных моментов времени, в том числе и для переходного процесса ($T = 75$);
- исследуемые алгоритмы устойчивы к изменению шага сетки, что особенно важно при таких больших объемах вычислений;
- при описании переходных процессов исследуемые трехмерные алгоритмы сходятся к статическому решению даже быстрее, чем двумерные.

Таким образом проведенный анализ подтверждает правомерность существования и использования предлагаемой методики и работоспособность построенных на ее основе вычислительных схем.

Разработана и использована новая методика построения вычислительных схем на базе характеристической формы уравнений динамики упруговязкопластических деформируемых твердых тел. Построены соответствующие вычислительные схемы; они использованы при моделировании динамики линейно-упругого изотропного однородного полого цилиндра. Тестирование полученных схем показало их работоспособность при больших толщинах цилиндра. Поскольку проверен лишь простейший случай моделирования, представляют интерес дальнейшие исследования поведения предлагаемой методики, в том числе: исследования поведения цилиндрических арок и дуг, цилиндров с полостями и включениями, кусочно-однородных цилиндров, анизотропных и неоднородных цилиндров, тел с другими видами кривизны. Все это представляется предметом дальнейших исследований.

1. Амосов А. А. Приближенная трехмерная теория нетонких упругих оболочек и плит. Докторская дисс., Ташкент, 1990. 336 с. URL: <http://tekhnosfera.com/view/239223/d#?page=1>
2. Самарский А. А. Теория разностных схем: учеб. пособие. М.: Наука, 1983. 616 с.
3. Зенкевич О., Морган К. Конечные элементы и аппроксимация / Пер. с англ. М.: Мир, 1986.
4. Кацикаделис Джон Т. Граничные элементы. Теория и приложения. М.: Изд-во АСВ, 2007. 366 с.
5. Поручиков В. Б. Методы динамической теории упругости. М.: Наука, Гл. ред. физ.-мат. лит., 1986. 328 с.
6. Булычев Г. Г., Пшеничнов С. Г. Распространение упругих волн в слоистом цилиндре // ДАН СССР. 1988. Т. 303, № 5. С. 1074—1078.
7. Булычев Г. Г. Построение матричной характеристической формы уравнений динамики анизотропных упруговязкопластических сред. // Известия АН России. Механика твердого тела. 1995. № 1. С. 91—95.
8. Булычев Г. Г. Метод пространственных характеристик в механике деформируемого твердого тела. Часть 1 // Информационные технологии. 2017. Т. 23, № 1. С. 21—27.
9. Булычев Г. Г. Динамическое разрушение изотропной пластинки // Строительная механика и расчет сооружений. 2008. № 3. С. 47—54.

G. G. Bulychev, DSc., Professor, e-mail: geo-bulychev@mail.ru
Moscow Technological University, Moscow

Simulation of Dynamics of the Hollow Cylinder by the Method of Spatial Characteristics

The method of spatial characteristics is a numerical and analytical technique in which the finite difference relations built after bringing the original model of the dynamics of deformed solid body to the characteristic form. From a programmer's point of view, this method can be classified as direct explicit grid-characteristic method of the first or second order approximation. The first order approximation of the remainder term has a parabolic shape, so spurious vibrations caused by the hyperbolicity of the original equations, damped. When considering the advantages and disadvantages of this method mentioned earlier that one of the main obstacles to its wider dissemination, is the difficulty of constructing algorithms for objects with curved surfaces. For such bodies spatial grid on which modeling check ditsya is curved and contains a different size of the cell that is not valid for the finite approximation templates. In this article, on the example of the dynamics of a hollow cylinder, it is shown that this difficulty can be overcome, but with some limitations and not for any classes of problems using divergence of the differential equations characteristic parts. To a first order approximation it was constructed numerically, then was realized and tested the corresponding algorithms. Algorithms showed good accuracy and stability in a large time interval. Method applied here can be used to study the dynamics and destruction of various dynamic bodies with curved surfaces.

Keywords: a method of spatial characteristics, numerical simulation, dynamics of constructions, algorithms

References

1. Амосов А. А. *Priblizhennaya trekhmernaya teoriya netonkikh uprugikh obolochek i plit*. Doktorskaya diss., Tashkent. 1990. 336 p. URL: <http://tekhnosfera.com/view/239223/d#?page=1> (in Russian).
2. Самарский А. А. *Teoriya raznostnykh skhem*. Ucheb. posobie. Moscow, Nauka, 1983. 616 p. (in Russian).
3. Зенкевич О., Морган К. *Konechniye ehlementi i approksimatsiya*. Per. s angl. Moscow, Mir, 1986. (in Russian).
4. Кацикаделис Джон Т. *Granichniye ehlementi. Teoriya i prilozheniya*. Moscow, Izd-vo ASV, 2007. 366 p. (in Russian).
5. Поручиков В. Б. *Metodiy dinamicheskoy teorii uprugosti*. Moscow, Nauka, Gl. red. Fiz.-mat. lit., 1986. 328 p. (in Russian).
6. Булычев Г. Г., Пшеничнов С. Г. Распространение упругих волн в слоистом цилиндре, *DAN SSSR*, 1988, vol. 303, no. 5, pp. 1074—1078 (in Russian).
7. Булычев Г. Г. Построение матричной характеристической формы уравнений динамики анизотропных упруговязкопластических сред, *Izvestiya AN Rossii. Zh. Mekhanika tverdogo tela*, 1995, no. 1, pp. 91—95 (in Russian).
8. Булычев Г. Г. Метод пространственных характеристик в механике деформируемого твердого тела. Часть 1, *Informacionniye tekhnologii*, 2017, vol. 23, no. 1, pp. 21—27. (in Russian).
9. Булычев Г. Г. Динамическое разрушение изотропной пластинки, *Stroitel'naya mekhanika i raschet sooruzheniy*, 2008, no. 3, pp. 47—54 (in Russian).

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ CRYPTOSAFETY INFORMATION

УДК 004.056.57

И. В. Машкина, д-р техн. наук, проф., e-mail: mashkina.vtzi@gmail.com,

А. О. Куприянов, магистрант, e-mail: anton_mail@ro.ru,

ФГБОУ ВПО "Уфимский государственный авиационный технический университет", г. Уфа

Классификационная схема и модели современных вирусных программ

Предложена классификационная схема современных вирусных программ. Классификационная схема является многоуровневой, к первому уровню которой относятся вирусы: статические, динамические, комбинированные — с дальнейшей детализацией. При разработке моделей вирусов на основе предложенной классификации использовались EPC-диаграммы, которые позволяют подробно описать архитектуру вируса, показывая логику его работы, функции, события и ресурсы.

Ключевые слова: вредоносная программа, вирус, классификационная схема вирусов, модели вирусов, статический вирус, динамический вирус, комбинированный вирус

Введение

Несмотря на значительные усилия специалистов в области разработки программного обеспечения (ПО) для обнаружения вредоносных кодов, задача обеспечения безопасности информационной системы от воздействия вирусов остается на сегодняшний день не решенной.

Вирус — это вредоносная программа, способная к самовоспроизведению, предназначенная для получения несанкционированного доступа к информации, обрабатываемой на ЭВМ, в целях причинения ущерба собственнику или владельцу.

Первым известным вирусом считается Creeper [1], обнаруженный в 1971 г. Теоретические основы самовоспроизводящихся программ были представлены в работе Veith Risak в 1972 г. [2]. В 1984 г. профессор Fredrick Cohen ввел понятие компьютерного вируса [3]. В 1985 г. были обнаружены первые широко распространенные статические исполняемые коды, которые не использовали технологии самозащиты.

В 2000 г. было положено начало значительным переменам в сфере ПО компьютерных систем: был написан скриптовый вирус, концепция которого была неизвестна [4]. Он активировался, как только пользователь открывал вложение к электронному письму. После запуска вирус рассылал себя по всем адресам, найденным в контактах почтового клиента пользователя, задействовав большой объем ресурсов. Это было началом новой эры вирусов, которые стали более сложными и опасными.

Хотя основные свойства вирусов остаются общими, сегодняшние образцы получили значительное развитие по сравнению с первыми вредонос-

ными кодами, они в силах причинить ущерб общемирового масштаба.

Современные вирусы создаются для извлечения коммерческой выгоды. Это может быть кража банковских данных в целях вывода средств со счета либо шифрование файлов пользователя, с последующим вымогательством денежных средств за их расшифровку. За разработкой вируса может стоять как один программист, так и группа программистов в зависимости от сложности вируса.

В настоящее время панацеи от вирусов не существует. Каждый новый вирус легко поражает свои цели, пока не найден способ борьбы с ним. Если обновление антивирусного ПО становится доступным всего через несколько часов после начала распространения вируса, оно не поможет, если система уже заражена.

Сегодня множество антивирусных компаний работают над задачей обеспечения защиты пользовательских компьютеров от вирусов, однако с развитием антивирусных технологий также развивается и вредоносное ПО.

Классификационная схема

При любом способе заражения вирусом целью антивируса является обнаружение внедрения вируса на ранней стадии. Успех создания антивирусного продукта в значительной мере зависит от наличия классификации вредоносных программ, позволяющей разработать модели вирусов. Эта информация необходима для построения адекватной модели антивирусного ПО.

Классификационная схема вирусов позволяет представить информацию о различных видах вредоносных программ в организованном виде.

Ранее предпринимались попытки классифицировать компьютерные вирусы, однако эти схемы были неполные или устаревали ввиду неработоспособности некоторых видов вирусов в современных операционных системах [5—7]. Кроме того, существующее многообразие различных классификаций современных вирусных программ не позволяет построить на их основе модели работы вирусов.

В то же время создание моделей вредоносных программ на основе какой-либо классификации по существу является единственным методом достаточно полного исследования потенциально возможных деструктивных воздействий на информационную систему.

Предлагается классификационная схема современных вирусных программ, на основании которой можно строить их модели (рис. 1).

Вирусы бывают трех типов: статические, динамические и комбинированные. Первые два типа являются основными, третий представляет смесь свойств статических и динамических вирусов.

Статические вирусы имеют постоянное тело и не могут сами его модифицировать; как следствие, такой код имеет сигнатуру и достаточно легко определяется антивирусными программами.

Динамические вирусы не имеют постоянного тела, способны сами модифицировать свой код, применяя при этом несколько техник (шифрование, обфускация) для затруднения автоматического анализа антивирусными программами.

На четвертом уровне классификации приведены названия вредоносных вирусных программ: Rootkit, Trojan, Worm.

Вирус Rootkit может быть реализован как в незащищенном исполнении, так и в защищенном, т. е. с шифрованием и/или с обфускацией. То же самое касается вредоносной программы Trojan. Однако отличие между моделью Rootkit и Trojan заключается в алгоритме работы модуля "полезная нагрузка". В первом случае работа модуля заключается в перехвате системных вызовов в целях сокрытия вредоносного кода в системе. Во втором случае модуль "полезная нагрузка" выполняет только деструктивные действия. Возможные варианты деструктивных действий Trojan приведены на пятом уровне классификации:

- Agent включает в себя подтипы: Downloader — скачивает из сети Интернет вредоносную программу и запускает ее на выполнение, Dropper — выделяет из своего файла остальные компоненты, записывает их на диск и запускает их на выполнение, Notifier — сообщает злоумыш-

леннику о том, что зараженный компьютер находится сейчас "на связи", Clicker — обращается к Интернет-ресурсам, сведения о которых указаны в теле вируса;

- Backdoor включает в себя подтипы: Spy — обеспечивает удаленный доступ злоумышленника к экрану, веб-камере, клавиатуре и файловой системе компьютера, DDoS — атакует компьютер жертвы, посылая многочисленные запросы с зараженных компьютеров, Proxy — осуществляет анонимный доступ к различным Интернет-ресурсам через компьютер-жертву;
 - Stealer включает в себя: PSW — отправляет хранящиеся на компьютере пользователя пароли злоумышленнику, Banker — отправляет учетные данные систем Интернет-банкинга, Mailfinder — собирает адреса электронной почты на компьютере пользователя и отправляет злоумышленнику, IM — отправляет учетные данные программ мгновенного обмена сообщениями;
 - Ransom — ограничивает возможность для пользователя работы на его компьютере, включая блокирование, в целях вымогательства денег;
 - Logic Bomb — искажает или удаляет данные с зараженного компьютера по временной метке.
- Червь Worm может быть реализован в незащищенном исполнении, зашифрованном виде, с му-

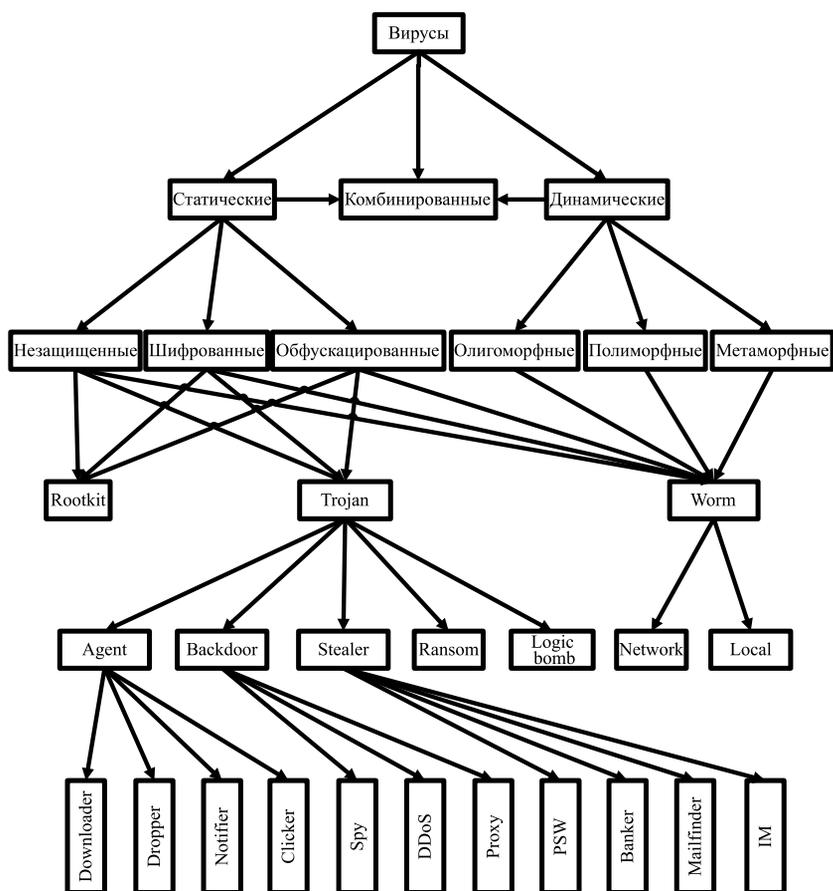


Рис. 1. Классификационная схема вирусов

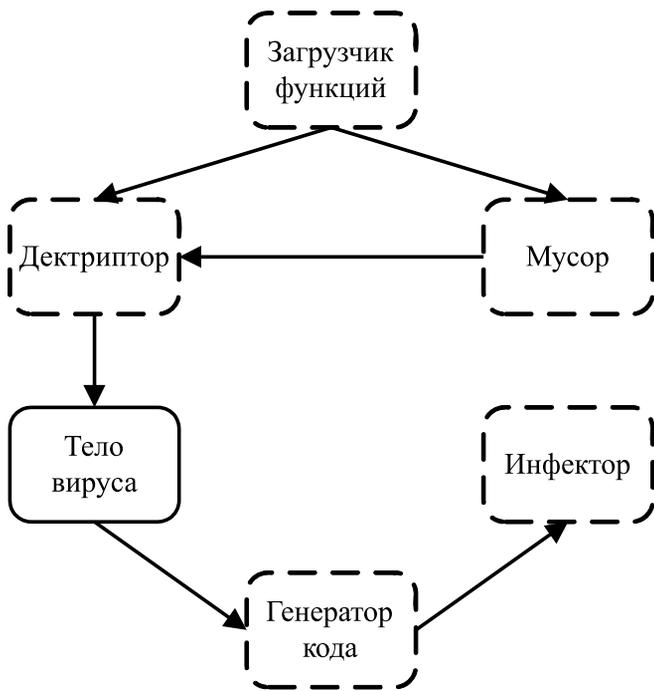


Рис. 2. Обобщенная модель состава вредоносного ПО

сорными инструкциями, а также как динамический вирус, т. е. в виде олигоморфного, полиморфного либо метаморфного. Разработанные модели олигоморфного, полиморфного и метаморфного вирусов характерны только для вируса Worm. Отличие вируса Worm с обфускацией или шифрованием от вирусов Rootkit, Trojan с шифрованием и обфускацией заключаются в алгоритмах "полезной нагрузки".

Названия Network и Local отображают варианты его распространения (сеть, система).

На рис. 2 отражены модули, которые может содержать вирус в зависимости от его типа.

В модели состава вредоносного ПО представлены модули, которые могут содержать вирус: сплошной линией обозначен модуль, обязательно присутствующий в составе вредоносной программы; штриховой линией обозначены модули, входящие в состав в зависимости от типа вируса. Стрелками показано возможное взаимодействие между модулями.

К основным функциям вирусов относятся способность к созданию собственных копий, к распространению в автоматическом режиме, инфицированию файловых объектов, а также деструктивные функции.

Разработка моделей вредоносного ПО

Эффективным инструментом исследования угроз нарушения безопасности информации, в том числе сложных самовоспроизводящихся алгоритмов, является моделирование.

Известны различные методы моделирования угроз. Построение EPC-диаграмм [8] — рациональное решение для разработки моделей вирусов, поскольку позволяет подробно описать архитектуру вируса, показывая логику его работы, функции, события и ресурсы.

Данное исследование посвящено разработке и анализу EPC-моделей вирусов. Разработаны шесть базовых моделей вредоносных программ. Рассмотрим EPC-модели вредоносного ПО, соответствующие третьему уровню классификационной схемы.

На рис. 3 представлена модель работы незащищенного вируса. После запуска он проецирует свой код в память, через таблицу импорта получает адреса функций Windows API, выполняет деструктивные действия, после чего вирус завершается. Исходя из данной модели можно сделать вывод о том, что данный вирус легко поддается анализу, поэтому в большинстве случаев антивирус находит вредоносный код без сигнатуры, определяя логику его работы, с помощью эвристического модуля.

На рис. 4 представлена модель работы шифрованного вируса. После запуска он проецирует свой код в память. С помощью ключа, жестко зашитого в теле вируса, расшифровываются служебные данные для поиска адресов функций Windows API. Затем осуществляется сам поиск. Далее расшифровывается основной код вируса, который выполняет деструктивные действия, после чего завершается действие вируса. Исходя из данной модели можно сделать вывод о том, что данный вирус является более устойчивым, чем предыдущий тип, поскольку нацелен на затруднение анализа с помощью автоматических средств антивируса, так как не содержит открытых данных об используемых функциях операционной системы и исполняемом коде.

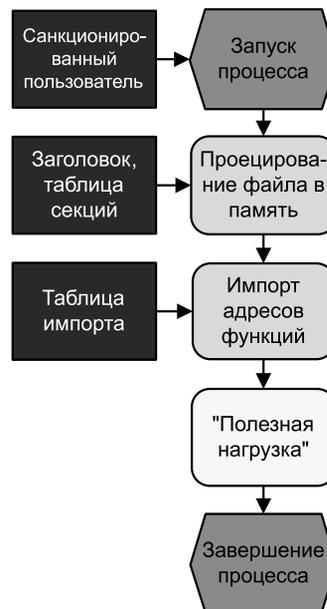


Рис. 3. Модель работы статического незащищенного вируса

На рис. 5 представлена модель работы обфусцированного вируса. После запуска он проецирует свой код в память, через таблицу импорта получает адреса функций Windows API, выполняет мусорный код и деструктивные действия, после чего вирус завершается. Исходя из данной модели можно сделать вывод о том, что данный вирус, как и предыдущий, нацелен на затруднение анализа автоматическими средствами антивируса, поскольку имеет в своем теле бессмысленный код, полноценный анализ которого невозможен из-за ограничения ресурсов системы, выделяемых на проверку.

На рис. 6 представлена модель работы динамического олигоморфного вируса. После запуска он проецирует свой код в память, через таблицу импорта получает адреса функций Windows API, выполняет деструктивные действия, в зависимости от способа распространения (система, сеть, съемные носители) размещает свои копии. При этом он модифицирует часть кода в каждой своей новой копии. Исходя из данной модели можно сделать вывод о том, что данный вирус нацелен на сбивание сигнатурного анализа антивируса.

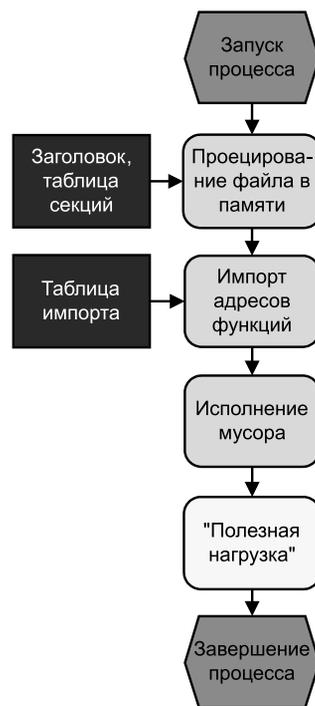


Рис. 5. Модель работы статического обфусцированного вируса

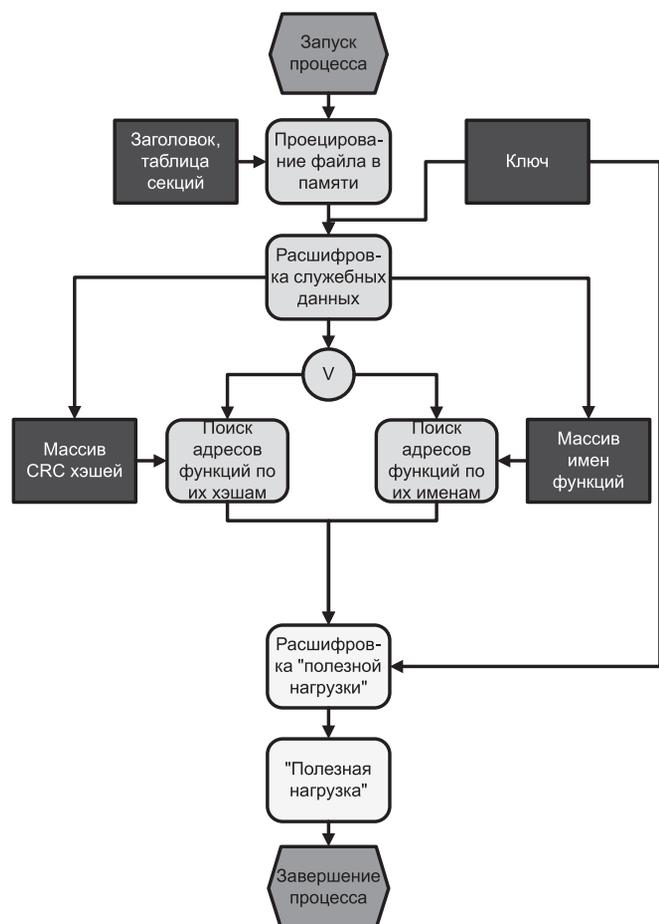


Рис. 4. Модель работы статического шифрованного вируса

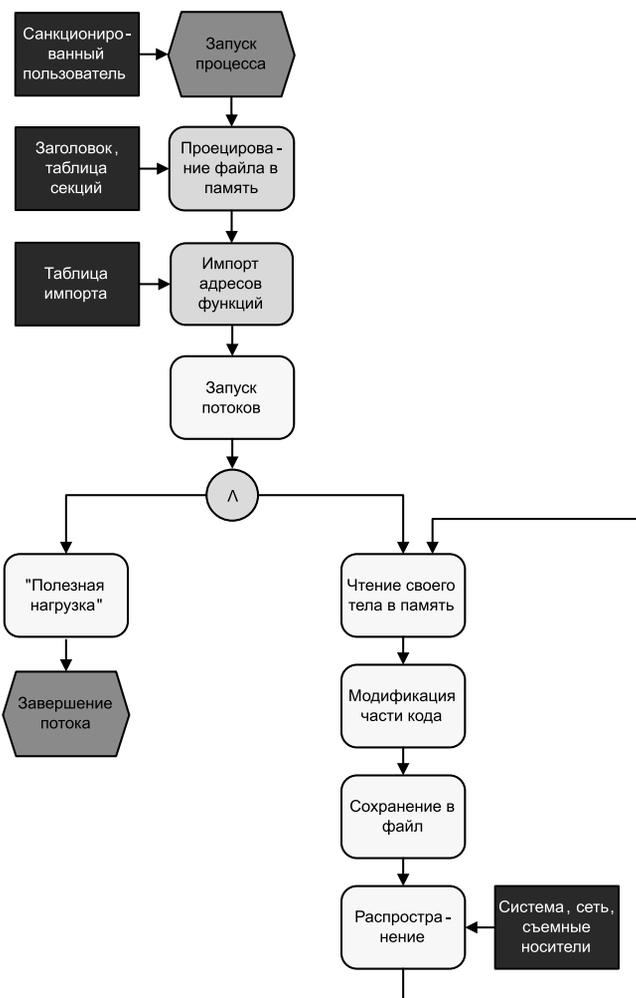


Рис. 6. Модель работы динамического олигоморфного вируса

На рис. 7 представлена модель работы динамического полиморфного вируса. После запуска он проецирует свой код в память. С помощью ключа, жестко зашитого в теле вируса, вирус расшифровывает служебные данные для поиска адресов функций Windows API, выполняет деструктивные действия, в зависимости от способа распространения (система, сеть, съемные носители) размещает свои копии. При этом тело вируса перешифровывается новым ключом, а иногда и другим алгоритмом, создается динамический декриптор, который не имеет постоянной сигнатуры. Исходя из данной модели можно сделать вывод о том, что данный вирус нацелен на сбивание сигнатурного анализа антивируса.

На рис. 8 представлена модель работы динамического метаморфного вируса. После запуска он проецирует свой код в память, через таблицу им-

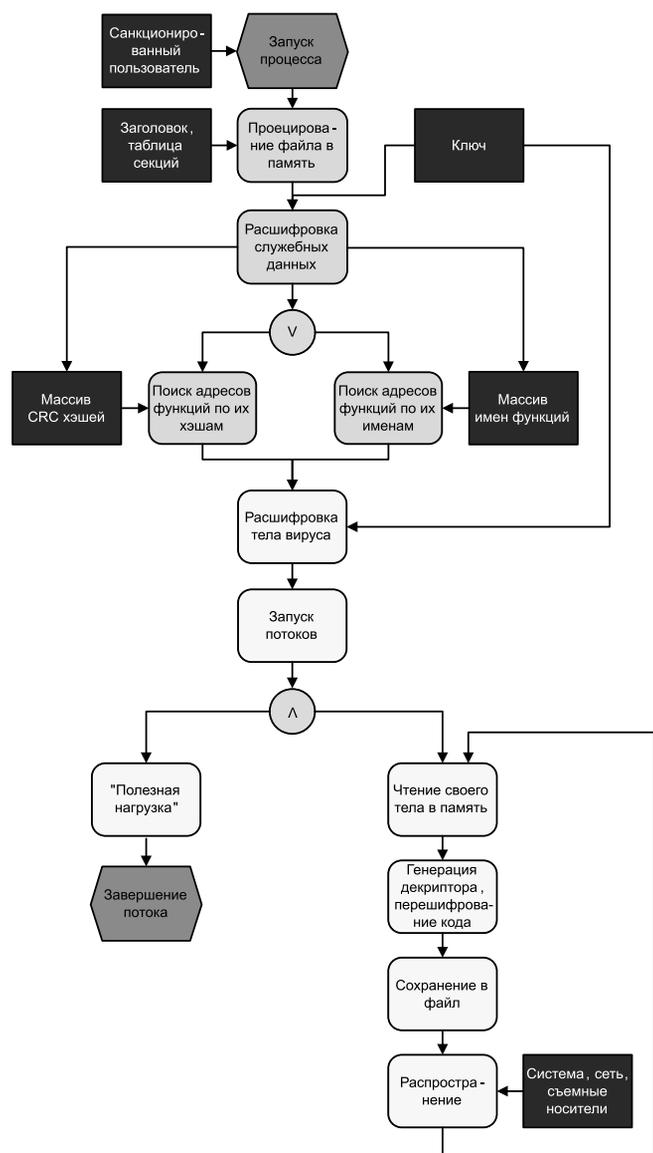


Рис. 7. Модель работы динамического полиморфного вируса

порта получает адреса функций Windows API, выполняет деструктивные действия, в зависимости от способа распространения (система, сеть, съемные носители) размещает свои копии. При этом тело вируса изменяется таким образом, чтобы не нарушалась логика работы и не было постоянной сигнатуры. Исходя из данной модели можно сделать вывод о том, что данный вирус нацелен на сбивание сигнатурного анализа антивируса.

Все описанные выше типы вирусов, за исключением "незащищенного" изначально были спроектированы таким образом, чтобы быть "невидимыми" для антивирусных программ. Поэтому без соответствующих записей в базе данных антивируса подобные новые образцы не обнаруживаются в автоматическом режиме эвристическим модулем, что является серьезной угрозой безопасности.

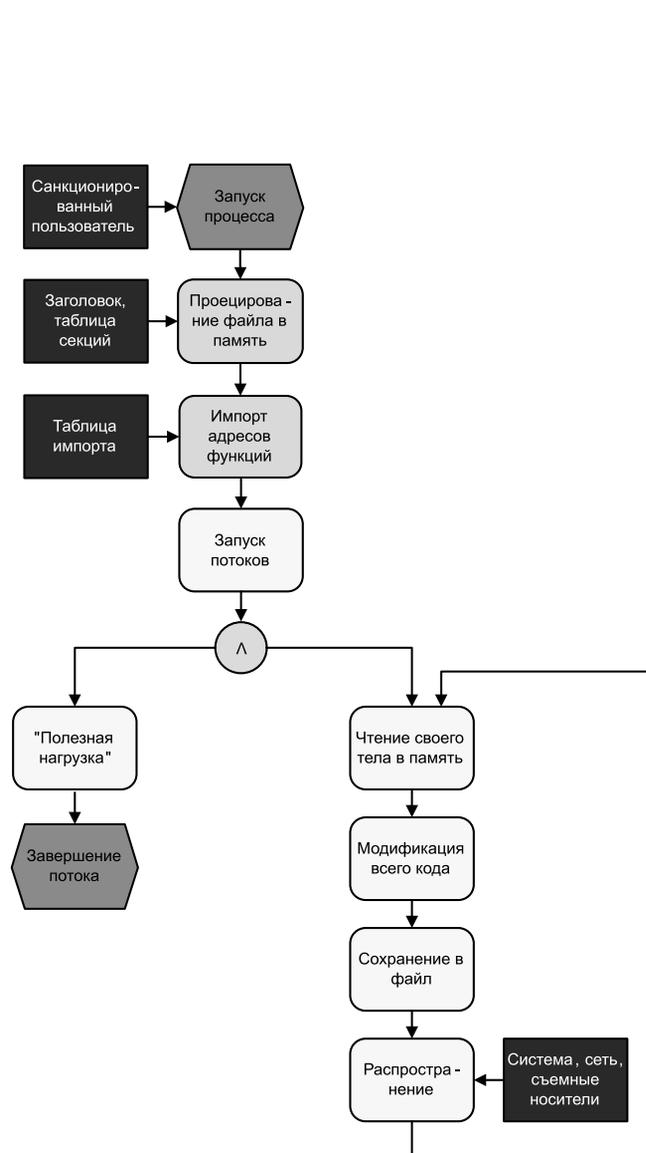


Рис. 8. Модель работы динамического метаморфного вируса

Заклучение

Предложенные классификационная схема и модели современных вирусных программ необходимы для построения адекватной модели антивирусного ПО. На основе идентификации образцов вредоносного ПО могут быть определены соответствующие меры пресечения заражения компьютерной системы вирусом.

Список литературы

1. **Chen T.** The Evolution of Viruses and Worms. URL: <http://vxheaven.org/lib/atc01.html> (дата обращения 06.02.2017).
2. **Risak V.** Selbstreproduzierende Automaten mit minimaler Informationsübertragung, Zeitschrift für Maschinenbau und Elektro-

technik. URL: <http://www.cosy.sbg.ac.at/~risak/bilder/selbstrep.html> (дата обращения 09.03.2011).

3. **Cohen F.** Computer virus. URL: https://en.wikipedia.org/wiki/Fred_Cohen (дата обращения 06.02.2017).
4. **Блазуккая Е. Ю., Шарафутдинов А. Г.** Вирусы нового поколения и антивирусы. *NOVAINFO.RU*. 2015. № 35. С. 92—94.
5. **Рычков А. В.** Классификация компьютерных вирусов // Ученые записки Российского государственного социального университета. 2012. № 3. С. 208—211.
6. **Трубачев Е. С.** Троянские программы: механизмы проникновения и заражения // Вестник Волжского университета им. В. Н. Татищева. 2011. № 18. С. 130—134.
7. **Андреев Д. А., Котрахов В. В., Остапенко А. Г.** Компьютерные вирусы: классификация и статический анализ // Информация и безопасность. 2010. № 2. С. 295—296.
8. **Шеер А. В.** ARIS — моделирование бизнес-процессов. М.: Вильямс, 2000. 175 с.

I. V. Mashkina, Professor, e-mail: mashkina.vtzi@gmail.com,
A. O. Kupritanov, Undergraduate, e-mail: anton_mail@ro.ru
Ufa State Aviation Technical University (UGATU), Ufa, Russia

The Classification Scheme and Modern Viruses Models

The classification scheme of the modern-day viruses software is proposed in this paper. The proposed scheme is multi-layer, the following bases of classification are used in the first layer: static, dynamic and combined. Static virus has a permanent body and can't modify it, such code has a signature and is easily detected by antivirus programs. Dynamic virus don't have a permanent body it can modify its code itself, using several techniques (encryption, obfuscation) to make more difficult automatic analysis by antivirus programs. Event-driven process chains (EPC) are used when developing the virus models based on the suggest classification. These models appropriate to describe the virus architecture in detail and show action logic, functions, events, and resources of virus. The proposed classification scheme and models of modern viruses are required to design an adequate mode of antivirus software. Applying of the models of malware samples is appropriate when designing antiviruses to prevent infection of the computer systems.

Keywords: malware, virus, virus classification scheme, viruses model, static virus, dynamic virus, combined virus

References

1. **Chen T.** The Evolution of Viruses and Worms. URL: <http://vxheaven.org/lib/atc01.html> (data obrashhenija 06.02.2017).
2. **Risak V.** Selbstreproduzierende Automaten mit minimaler Informationsübertragung, Zeitschrift für Maschinenbau und Elektrotechnik. URL: <http://www.cosy.sbg.ac.at/~risak/bilder/selbstrep.html> (data obrashhenija 09.03.2011).
3. **Cohen F.** Computer virus. URL: https://en.wikipedia.org/wiki/Fred_Cohen (data obrashhenija 06.02.2017).
4. **Blazuckaja E. Ju., Sharafutdinov A. G.** Virusy novogo pokolenija i antivirussy (A new generation of viruses and antivirus). *NOVAINFO.RU*, 2015, no. 35, pp. 92—94 (in Russian).

5. **Rychkov A. V.** Klassifikacija komp'juternyh virusov (Classification of computer viruses). *Uchenye zapiski Rossijskogo gosudarstvennogo social'nogo universiteta*, 2012, no. 3, pp. 208—211 (in Russian).
6. **Trubachev E. S.** Trojanskije programmy: mehanizmy proniknovenija i zarazhenija (Trojan: mechanisms of infiltration and infection). *Vestnik volzhskogo universiteta im. V. N. Tatishheva*, 2011, no. 18, pp. 130—134 (in Russian).
7. **Andreev D. A., Kotrahov V. V., Ostapenko A. G.** Komp'juternye virusy: klassifikacija i staticheskij analiz (Computer Viruses: classification and static analysis), *Informacija i bezopasnost'*, 2010, no. 2, pp. 295—296 (in Russian).
8. **Sheer A. V.** ARIS — modelirovanie biznes-processov (Business Process Modeling). Moscow, Vil'jams, 2000, 175 p. (in Russian).

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ INFORMATION TECHNOLOGIES IN EDUCATION

УДК 004.05 004.3-6

Р. Ш. Загидуллин, канд. техн. наук, доц., e-mail: zag@bmstu.ru,

А. С. Черников, канд. техн. наук, начальник учебно-научной лаб.

"Компьютерные информационные системы и технологии", e-mail: chernik@bmstu.ru,

А. А. Чибисов, программист научно-образовательного центра "Электронный Университет",
e-mail: chibisov.alexandr.25@gmail.com,

Московский государственный технический университет им. Н. Э. Баумана

Решение проблемы объединения сред MOODLE и Электронный университет МГТУ им. Н. Э. Баумана

Рассматривается методика интегрирования сред MOODLE (свободно распространяемой среды) и Электронного Университета (среда, разработанная в МГТУ им. Н. Э. Баумана). Если первая представляет собой удобный инструмент в руках преподавателя, то вторая среда — инструмент поддержки принятия и реализации управленческих решений на уровне ректората и деканатов. До сих пор их взаимодействие осуществлялось за счет ручного переноса данных MOODLE в среду Электронный университет со всеми недостатками, присущими ручному сопровождению переноса данных. Авторами рассматривается реализация интегрированной работы двух сред в целях улучшения качества организации и управления учебным процессом.

Ключевые слова: информационная система управления, MOODLE, базы данных, Электронный университет, студент, структура курса, класс, человеческий фактор

Введение

В большинстве университетов учебный процесс организуется и контролируется с помощью специальных информационно-управляющих систем. Одна часть университетов использует в качестве таких систем универсальные платформы, адаптированные под их задачи компаниями — поставщиками услуг [1—3], другая — создает свои системы самостоятельно [4—8]. В МГТУ им. Н. Э. Баумана создана собственными силами и в течение ряда лет функционирует система Электронный университет (ЭУ), которая используется не только как информационный ресурс, но и как инструмент поддержки принятия и реализации управленческих решений на уровне ректората и деканатов [9]. ЭУ

хранит всю необходимую информацию об учебных планах, студентах, изучаемых ими дисциплинах, полученных отметках, позволяет проводить статистическую обработку данных по целому ряду параметров и предоставляет администрации объективную информацию для управления учебным процессом. Данный ресурс доступен из внутренней сети МГТУ им. Н. Э. Баумана (рис. 1).

Преподаватели же Университета в своей повседневной работе используют очень популярную, свободно распространяемую модульную объектно-ориентированную динамическую учебную среду MOODLE, ориентированную, прежде всего, на организацию взаимодействия между преподавателем и студентами как в процессе очного обучения, так и с применением дистанционных методов [10—12].

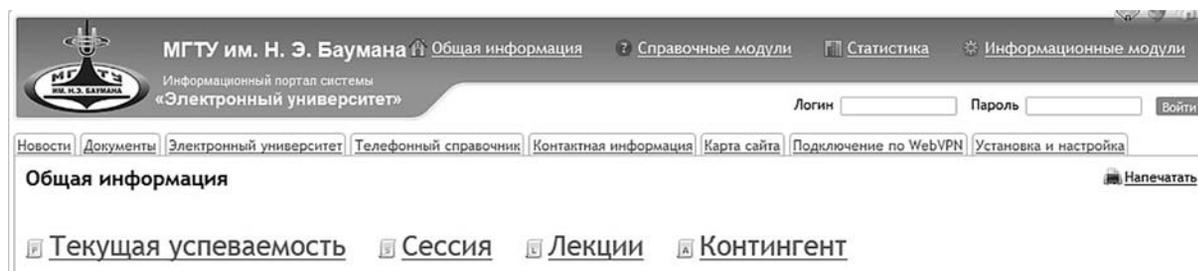


Рис. 1



Рис. 2

В среде MOODLE легко создавать и хранить различные учебные материалы, проводить контрольные мероприятия и хранить их результаты.

Доступ к среде при этом осуществляется через ресурс специализированной Электронной Образовательной Системы МГПУ им. Н. Э. Баумана (ресурс http://e-learning.bmstu.ru/new_face/) (рис. 2).

К сожалению, системы MOODLE и ЭУ из-за невозможности прямого взаимодействия функционируют отдельно, что снижает практическую ценность и эффективность их использования.

Действительно, преподаватель работает со студентами в среде MOODLE. Для этого ему требуется вводить в систему все данные о студентах, читаемых дисциплинах, отметках и пр. Значительная часть этих данных должна быть введена и в систему ЭУ. Ввод данных в обе системы осуществляется преподавателем вручную. Таким образом, на этапе обмена информацией между MOODLE и ЭУ включается "человеческий фактор". Это и обуславливает снижение быстродействия, появление ошибок передачи данных и уменьшает актуальность переданных данных, поскольку преподаватель вынужден загружать вручную практически одни и те же данные в две разные системы, что неизбежно повышает вероятность появления ошибок. Нерационально используется рабочее время преподавателя, возникает проблема неоднозначности и синхронизации данных в системах MOODLE и ЭУ. Временная задержка ввода снижает не только актуальность данных, но и эффективность принятых административных решений.

Для уменьшения влияния "человеческого фактора", обеспечения однозначного представления и синхронизации данных целесообразно осуществить интеграцию административной информационно-управляющей системы ЭУ и учебной среды MOODLE.

Постановка задачи

Для уменьшения влияния "человеческого фактора", обеспечения однозначного представления и синхронизации данных целесообразно осуществить интеграцию административной информационно-управляющей системы ЭУ и учебной среды MOODLE.

Для интеграции систем ЭУ и MOODLE потребовалось решить следующие задачи:

- формирование концепции единого представления данных в обеих системах с разработкой перечня необходимых модификаций структур данных;
- определение принципов и правил двустороннего обмена данными между системами в последовательности, отражающей реальный учебный процесс и приоритетность первичных источников информации;
- разработка, отладка и тестирование программного модуля, реализующего ранее определенные концепции и правила и автоматизирующего обмен данными между ЭУ и MOODLE.

Прежде всего, при решении задачи интеграции двух различных систем встает проблема соответствия содержания и формы представления данных, которые будут участвовать во взаимодействии интегрируемых систем.

Методика формирования единого представления данных в MOODLE и ЭУ

Для определения возможности формирования единого представления данных был проведен анализ как содержания хранимой информации, так и форматов представления данных в обеих системах.

Как известно, MOODLE используется для поддержки дистанционной и смешанной форм обучения, для создания электронных обучающих материалов, проведения консультаций. MOODLE позволяет выкладывать необходимые материалы для методического сопровождения занятий и лабораторных работ, а также организовывать общение студентов и преподавателей, в том числе с отправкой файлов на проверку (домашние задания, рефераты, отчеты по лабораторным работам). Кроме того, система обеспечивает полноценное ведение преподавателем текущей успеваемости. Таким образом, MOODLE практически является центром создания учебного материала и обеспечения интерактивного взаимодействия между участниками учебного процесса.

Контрольные мероприятия могут быть реализованы внутренними средствами системы и внедряемыми в MOODLE, такими, например, как Hot Potatoes [13]. Автоматизация проведения контрольных мероприятий позволяет получать данные сразу в виде баллов соответствующего модуля. Система может служить первичным источником информации об успеваемости студентов, в том числе и для системы ЭУ. Вместе с тем, являясь в большей мере инструментом преподавателя, MOODLE обладает некоторой избыточностью с точки зрения потребностей административной системы ЭУ, поэтому часть хранимой в ней информации не предназначена для передачи в ЭУ.

Система ЭУ, разработанная в МГПУ им. Н. Э. Баумана, является закрытой и обладает оригинальной

структурой и функциональными возможностями, поэтому никакие модификации структур и основных данных внутри нее извне проведены быть не могут [9]. Среда MOODLE, напротив, является открытой и предоставляет возможности по дополнению и частичному изменению своих структур данных в соответствии с требованиями пользователей или внешних систем. Таким образом, для обеспечения корректного взаимодействия двух сред необходимо привести формат хранения данных в среде MOODLE в соответствие с требованиями системы ЭУ.

Все объекты в системе ЭУ — студенты, дисциплины, контрольные мероприятия и т. д. — имеют уникальные идентификаторы (GUID — Globally Unique Identifier), которые позволяют их однозначно определить. Соответственно, любые данные, которые предполагается связать с этими объектами (например, оценки за хранящиеся в ЭУ контрольные мероприятия), должны оперировать этими идентификаторами. Используя GUID, можно однозначно гарантировать правильность построения цепочки данных "студент — дисциплина — контрольное мероприятие — результат". Для обеспечения взаимной однозначной совместимости двух систем по данным необходимо, чтобы и MOODLE оперировала этими идентификаторами, однако стандартная схема базы данных системы MOODLE не предусматривает такой возможности. Эту проблему можно решить, добавив в соответствующие таблицы базы данных MOODLE дополнительные поля для хранения получаемых GUID из ЭУ либо создав отдельные таблицы для хранения получаемых от ЭУ дополнительных параметров. Однако последний способ представляется нежелательным, так как создание таких дополнительных таблиц приведет к избыточному дублированию данных.

Существует несколько способов для внесения дополнительных полей для хранения информации в таблицах базы данных MOODLE — непосредственное исполнение специальных SQL-скриптов на базе данных, использование средств СУБД или web-интерфейсов [14]. В данной работе предложен и реализован вариант модификации таблиц в базе данных MOODLE в соответствии с форматом данных ЭУ с помощью специальных скриптов на языке PHP, использующих API системы MOODLE. Использование API-функций вместо прямых SQL-запросов обеспечивает безопасность и целостность данных, позволяет избежать ошибок и предоставляет разработчику возможность использовать более простые и понятные конструкции при работе с базой данных [15]. Этот способ работы с базой данных оказался наиболее эффективным и удобным при решении задач изменения структуры базы данных, так как он обеспечивает высокий уровень абстракции и безопасности данных, а также не зависит от используемой СУБД.

В результате проведенной модификации стало возможным осуществлять полноценную загрузку данных из ЭУ. Таким образом, была обеспечена работа с одинаковыми данными как на уровне административной системы ЭУ, так и непосредственно в учебном процессе в рамках среды MOODLE.

Стоит отметить, что непосредственная модификация таблиц базы данных MOODLE имеет ряд недостатков, основной из которых — невозможность сохранить эти изменения при переустановке системы MOODLE или обновлении ее до новой версии [16]. Однако эта проблема решается путем использования резервных копий базы данных и повторного запуска скриптов по модификации таблиц перед развертыванием резервной копии в обновленной системе.

Реализация обмена данными MOODLE И ЭУ

Импорт данных в MOODLE. Следующим шагом в осуществлении интеграции систем ЭУ и MOODLE является импорт данных о студентах и дисциплинах. Поскольку только ЭУ является официальным источником этих данных в МГТУ им. Н. Э. Баумана, то все программные среды, для работы которых необходимо наличие указанных данных, должны обращаться за ними к соответствующим подсистемам ЭУ.

ЭУ является закрытой системой, поэтому необходимые данные из системы можно получить, только воспользовавшись их загрузкой через специальные web-сервисы, предоставляемые соответствующими подсистемами ЭУ, при этом нужно обладать правами по доступу к запрашиваемой информации. Обработав получаемые при таких обращениях XML-файлы, можно выделить необходимые данные [17].

При решении задачи интеграции двух систем необходимо обеспечить передачу данных о списках студентов, изучаемых ими дисциплинах и составе контрольных мероприятий из среды ЭУ в среду MOODLE. Импорт данных осуществляется с помощью специальных скриптов на языке PHP, с использованием API MOODLE для создания соответствующих объектов (пользователей и курсов) в этой системе.

Импорт данных из ЭУ можно разделить на две подзадачи: импорт списков студентов и импорт дисциплин и контрольных мероприятий. При работе со списками студентов нужная информация переносится в базу данных среды MOODLE. Кроме того, обеспечивается создание пользователей MOODLE на основе полученных данных (рис. 3).

При этом логины и пароли для создаваемых учетных записей генерируются автоматически. В результате в системе MOODLE появляется набор новых пользователей, чьи параметры полностью соответствуют данным из ЭУ (рис. 4).

Задача импорта дисциплин и контрольных мероприятий несколько сложнее. Обычно дисциплины (они называются курсами в MOODLE) и их на-

| id | stud_guid | group_name | group_guid | username | password | firstname | lastname | email |
|----|--------------------------------------|------------|--------------------------------------|-------------------|----------------------|-----------|----------|------------------------|
| 72 | f8847164-dc96-11e2-ab45-005056962143 | PK6-41 | 4776c1f8-51d3-11e3-ae8d-005056960017 | 5vorobev_rk6-41 | \$2y\$10\$XEKb5fM8Z | Евгений | Воробьев | 5vorobev_rk6-41@m.ru |
| 73 | b4367b86-dcb6-11e2-bb5f-005056962143 | PK6-41 | 4776c1f8-51d3-11e3-ae8d-005056960017 | 6gorshkova_rk6-41 | \$2y\$10\$iMxVmWoaf | Елизавета | Горшкова | 6gorshkova_rk6-41@m.ru |
| 74 | 7e5426a0-dcc3-11e2-8bc4-005056962143 | PK6-41 | 4776c1f8-51d3-11e3-ae8d-005056960017 | 7gorshkova_rk6-41 | \$2y\$10\$YZ7iFDwUS | Кристина | Горшкова | 7gorshkova_rk6-41@m.ru |
| 75 | 6ab0d494-dd8c-11e2-811d-005056962143 | PK6-41 | 4776c1f8-51d3-11e3-ae8d-005056960017 | 8erofeev_rk6-41 | \$2y\$10\$rC92TSMak | Егор | Ерофеев | 8erofeev_rk6-41@m.ru |
| 76 | 0bb52086-de35-11e2-9842-005056962143 | PK6-41 | 4776c1f8-51d3-11e3-ae8d-005056960017 | 9zhukov_rk6-41 | \$2y\$10\$ihg1fY9VVs | Павел | Жуков | 9zhukov_rk6-41@m.ru |

Рис. 3

| Имя / Фамилия / Имя - фонетическая запись / Фамилия - фонетическая запись / Отчество или второе имя / Альтернативное имя | Адрес электронной почты | Город | Страна |
|--|---------------------------|--------|--------|
| Аделина Шивидова | 20shividova_rk6-41@m.ru | Москва | Россия |
| Александр Скороходов | 15skorohodov_rk6-41@m.ru | Москва | Россия |
| Александр Щербаков | 21shherbakov_rk6-41@m.ru | Москва | Россия |
| Андрей Сологуб | 17sologub_rk6-41@m.ru | Москва | Россия |
| Антон Иванов | 10ivanov_rk6-41@m.ru | Москва | Россия |
| Аршак Товмасыан | 18tovmasyan_rk6-41@m.ru | Москва | Россия |
| Борис Авдеев | 1avdeev_rk6-41@m.ru | Москва | Россия |
| Вадим Овчинников | 12ovchinnikov_rk6-41@m.ru | Москва | Россия |
| Василий Руденко | 14rudenko_rk6-41@m.ru | Москва | Россия |
| Владимир Бочаров | 4bocharov_rk6-41@m.ru | Москва | Россия |

Рис. 4

Diplom 2015

НАВИГАЦИЯ

- В начало
- Моя домашняя страница
- Страницы сайта
- Мой профиль
- Курсы

НАСТРОЙКИ

- Настройки главной страницы
- Режим редактирования
- Редактировать настройки
- Пользователи
- Фильтры
- Отчеты
- Резервное копирование
- Восстановить
- Банк вопросов

Доступные курсы

- Программирование графических приложений (Дисциплина 6) РК6-42
- Электроника РК6-42
- Базы данных РК6-42
- Основы автоматизированного проектирования РК6-42

Рис. 5

полнение в среде MOODLE создаются преподавателями, ведущими эти дисциплины. Данные по конкретной дисциплине, включая контрольные мероприятия, должны строго соответствовать учебному плану, хранящемуся в ЭУ. Поэтому при создании новой дисциплины в MOODLE преподавателю необходимо соблюдать основные параметры дисциплин и в MOODLE, и в ЭУ. Создание вручную соответствующих объектов в MOODLE для всех дисциплин и контрольных мероприятий (заданий в MOODLE) может привести к ошибкам и неточностям, что сделает невозможным передачу информации об успеваемости студентов в ЭУ.

Для устранения возможности появления ошибок возникает задача автоматической генерации в среде MOODLE списка дисциплин и контрольных мероприятий на основе данных, полученных из системы ЭУ. При этом для каждой дисциплины создается "заготовка" — начальная версия дисциплины в MOODLE с названием, соответствующим учебному плану, и с набором контрольных мероприятий (заданий), предусмотренных учебным планом (рис. 5).

После этого преподаватель может самостоятельно настроить параметры курса, разместить контрольные мероприятия в нужных разделах, добавить необходимые методические материалы. При этом привязка дисциплины и контрольных мероприятий к специальным идентификаторам, полученным

от ЭУ, остается даже при изменении названий соответствующих объектов. Так обеспечивается однозначное соответствие дисциплин и мероприятий как в учебном плане ЭУ, так и в среде MOODLE.

Особенности представления структуры дисциплин в MOODLE. При создании автоматической структуры дисциплины в MOODLE необходимо выбрать представление контрольных мероприятий внутри дисциплины. Выполнение контрольных мероприятий должно быть каким-либо образом оценено, поэтому для их представления в среде MOODLE необходимо использовать модуль, позволяющий преподавателю выставить оценку. Таким модулем является "Задание" [16]. При этом можно использовать два основных подхода к представлению мероприятий. Первый и самый простой — это однозначное соответствие модуля "Задание" в среде MOODLE и контрольного мероприятия, например, лабораторной работы. Тогда все лабораторные работы будут представлены заданиями с соответствующими именами. Такая структура подходит для случая, когда не требуется выделять отдельные этапы в выполнении контрольного мероприятия, или же оценки за все этапы выставляются преподавателем вручную.

Однако если необходимо явно выделить этапы выполнения задания (например, "выдано" — "выполнено" — "защищено" для лабораторных работ) или оценка за выполнение части задания будет проставляться автоматически (например, после прохождения, встроенного в MOODLE теста), то необходимо реализовать структуру курса с выделением этапов контрольных мероприятий (рис. 6).

При этом нельзя не отметить недостаток такого подхода — заметно увеличивается число заданий

в рамках дисциплины. Например, при наличии в дисциплине восьми лабораторных работ при стандартном подходе в MOODLE будет представлено восемь заданий, при выделении трех этапов в каждой работе — 24 задания. Поэтому использование такого подхода в каждом отдельном случае должно быть методически обосновано.

Необходимо отметить, что несмотря на автоматическую генерацию дисциплин и контрольных мероприятий, преподаватель должен соблюдать неизменность составляющих, полученных на основании данных ЭУ. То есть нельзя удалять как задания из заготовки, так и в целом дисциплины, поскольку это приведет к некорректной передаче информации об успеваемости в ЭУ. При этом будет возможно добавлять свои, не передаваемые в ЭУ материалы и даже контрольные мероприятия, не предусмотренные учебным планом, изменять их положение в рамках дисциплины, менять их названия.

Экспорт данных из MOODLE. Необходимо отметить, что поскольку система MOODLE используется непосредственно в учебном процессе и обладает некоторой избыточностью с точки зрения потребностей ЭУ, то часть оценок и отметок о прохождении лабораторных работ и выполнении контрольных мероприятий могут храниться в рамках этой среды без передачи в ЭУ. При этом результаты мероприятий, входящих в соответствующее модули ЭУ, должны быть обязательно загружены в ЭУ. Таким образом, среда MOODLE будет являться первичным источником информации об успеваемости студентов для системы ЭУ.

Среда MOODLE получает от системы ЭУ данные о студентах и дисциплинах, необходимые для организации учебного процесса, и передает обратно информацию о ходе выполнения контрольных мероприятий студентами. При этом правильность передаваемой ЭУ информации гарантируется тем, что обе системы используют одинаковые исходные данные (студенты, дисциплины, контрольные мероприятия). Обмен данными как между внутренними подсистемами, так и с внешними пользователями ЭУ, осуществляется на основе технологии веб-сервисов [9]. Для представления данных используется единый формат — xml.

Таким образом, для обеспечения поступления данных о выполнении контрольных мероприятий в ЭУ среда MOODLE должна предоставлять эти данные в виде xml-файла. Эта задача решается с помощью скрипта на языке PHP, реализующего работу веб-сервиса и обеспечивающего генерацию этого xml-файла. Его структура (дерево элементов xml) определяется требованиями подсистемы "Текущая успеваемость" системы ЭУ. Так как ЭУ является

Электроника РК6-42

В начало ▶ Курсы ▶ Разное ▶ Электроника РК6-42

НАВИГАЦИЯ

В начало

- Моя домашняя страница
- Страницы сайта
- Мой профиль
- Текущий курс
 - Электроника РК6-42
 - Участники
 - Значки
 - Общее
 - Тема 1
 - Тема 2
 - Тема 3
 - Тема 4
 - Тема 5
 - Тема 6
 - Тема 7
 - Тема 8
 - Тема 9
 - Тема 10

Лабораторная работа №1 - Выдача

Лабораторная работа №1 - Выполнение

Лабораторная работа №1 - Защита

Лабораторная работа №2 - Выдача

Лабораторная работа №2 - Выполнение

Лабораторная работа №2 - Защита

Лабораторная работа №3 - Выдача

Лабораторная работа №3 - Выполнение

Лабораторная работа №3 - Защита

Лабораторная работа №4 - Выдача

Лабораторная работа №4 - Выполнение

Лабораторная работа №4 - Защита

Новостной форум

Рис. 6

закрытой системой, то невозможно осуществить загрузку какого-либо файла во внутренние структуры этой системы. Поэтому информация о результатах контрольных мероприятий генерируется по запросу от ЭУ, при этом параметром такого запроса является интервал времени, в течение которого изменялись оценки в среде MOODLE. То есть ЭУ предоставляются только те оценки, значения которых изменялись в заданный промежуток времени.

Программный модуль автоматизации обмена данными. В соответствии с методикой, описанной выше, задачи по обеспечению единого представления данных и взаимообмену информацией

между системами были решены за счет создания специальных скриптов на языке PHP, использующих технологии web-сервисов и API-функции, предоставляемые системой MOODLE. После объединения этих скриптов был сформирован программный модуль, включенный в состав локального сервера MOODLE и обеспечивающий автоматический обмен данными между системами MOODLE и ЭУ. Функциональная схема модуля приведена на рис. 7.

Функционально в модуле можно выделить два вида submodule — импорта и экспорта данных, при этом в процессе импорта данных также осуществляется подготовка базы данных MOODLE к приему данных из ЭУ.

Использование submodule "Формирование списка студентов" и "Формирование списка дисциплин (курсов)" невозможно без предварительной отработки submodule "Модификация таблиц". Кроме того, submodule "Отправка информации в ЭУ" будет функционировать только при условии, что данные в среде MOODLE представлены в формате ЭУ и привязаны к идентификаторам этой системы. Исходя из этого была обеспечена строго регламентированная последовательность запуска submodule импорта данных.

Тестирование на реальных учебных группах и дисциплинах показало работоспособность созданного программного модуля: были успешно модифицированы таблицы базы данных MOODLE, а затем импортированы списки студентов, дисциплин и контрольных мероприятий из ЭУ. Таким образом, преподаватели смогли работать с официальными списками групп и по актуальным учебным планам. Кроме того, был протестирован механизм передачи оценок от MOODLE в ЭУ, который позволил достаточно оперативно загрузить информацию об успеваемости в ЭУ без дополнительного ввода данных.



Рис. 7

Обсуждение

Разработанный программный модуль может наращивать свою функциональность по мере модернизации системы ЭУ. Одно из возможных усовершенствований программного модуля интеграции заключается в реализации функции обновления списков студентов и дисциплин. В ходе учебного процесса возможны изменения данных в системе ЭУ, поэтому необходимо обеспечить актуальность информации, загруженной ранее в среду MOODLE. Для этого следует разработать отдельный модуль, который будет сравнивать информацию о студентах и дисциплинах в системах MOODLE и ЭУ и при необходимости вносить соответствующие изменения в MOODLE.

Заключение

В процессе интеграции систем MOODLE и ЭУ удалось создать уникальный программный продукт, решающий следующие задачи:

- приведение данных в системах к единому представлению путем модификации структуры данных в системе MOODLE (а именно модификации таблиц в базе данных);
- импорт актуальных официальных данных о студентах, дисциплинах и контрольных мероприятиях в систему MOODLE;
- экспорт данных об успеваемости студентов в информационно-управляющую систему вуза в режиме реального времени с использованием технологии web-сервисов.

Интеграция систем MOODLE и ЭУ позволила получить следующие практически значимые результаты:

- устранение неполного соответствия основной информации (студенты и дисциплины) в учебной и административной системах вуза;

- избавление от несвоевременного ввода данных о выполнении контрольных мероприятий в административную систему;
- более рациональное использование времени преподавателя (благодаря устранению необходимости дублирования (двойного ввода) данных в двух параллельно функционирующих системах);
- снижение числа ошибок в системах из-за влияния "человеческого фактора";
- повышение быстроты и эффективности работы системы административного управления учебным процессом университета.

Дальнейшее поддержание и модернизация разработанного программного модуля является актуальным и перспективным направлением для развития технологий дистанционного обучения в России.

Несмотря на то, что разработанный программный модуль создавался для работы с ЭУ и не может быть напрямую использован в других организациях и вузах, разработанные в рамках проекта концепции и подходы могут быть успешно адаптированы к задачам интеграции подобных систем.

Список литературы

1. **Intranet:** Academic — система автоматизации управления учебным процессом. URL: <http://softwareinc.ru/solutions/intranet-academic/> (дата обращения 11.09.2016).
2. **Автоматизированная Система Управления деятельностью Университета.** URL: <http://www.akvadra.ru/catalog/7.html> (дата обращения 11.09.2016).
3. **University Management System UMS.** URL: <http://ampletrails.com/university-management-system-ums> (дата обращения 11.09.2016).
4. **Contribution of Integrated Management Systems to University Management: Case Study of the Federal University of Rio Grande Do Norte** // Journal of Social Sciences. 2011. Vol. 7 (3). P. 415—422.
5. **University information system MENDEL.U.** URL: <http://is.mendelu.cz/> (дата обращения 11.09.2016).
6. **Делигирова О. А.** Информационные технологии в управлении персоналом вуза // Перспективы развития информационных технологий. 2013. № 14. С. 67—73.
7. **Кочетов А. И., Крапухина Н. В., Проничкин С. В.** Разработка систем поддержки принятия решений для управления качеством деятельности вуза // Экология человека. 2009. № 9. С. 39—45.
8. **Логвинов С. И., Романов В. А.** Применение информационно-аналитических систем в образовательном процессе вуза: звено "Факультет — кафедра" // Современные проблемы науки и образования. 2014. № 1. URL: <https://science-education.ru/ru/article/view?id=12121> (дата обращения 11.09.2016).
9. **Информационная управляющая система МГТУ им. Н. Э. Баумана "Электронный университет": концепция и реализация.** // [Т. И. Агеева и др.] / Под ред. И. Б. Федорова, В. М. Черненко. М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. 376 с.
10. **Swart A. J.** Student usage of a learning management system at an open distance learning institute: A case study in electrical engineering // International Journal of Electrical Engineering Education. 2015. Vol. 52. P. 142—154.
11. **Poncela A.** A Blended Learning Approach for an Electronic Instrumentation Course // International Journal of Electrical Engineering. 2013. Vol. 50. P. 1—18.
12. **Implementation of Industrial Automation Laboratories for E-learning** // International Journal of Electrical Engineering Education. 2012. Vol. 49. P. 402—418.
13. **Vlchner A.** Moodle 3 Administration. Packt Publishing, 2016. 492 p.
14. **DeLisle M.** Mastering phpMyAdmin 2.8 for Effective MySQL Management. Third edition. Packt Publishing, 2006. 283 p.
15. **Открытая документация по системе Moodle 2.8.** URL: https://docs.moodle.org/28/en/Main_page (дата обращения 11.09.2016).
16. **Анисимов А. М.** Работа в системе дистанционного обучения Moodle. ХНАГХ, 2009. 292 с.
17. **Vaqqas M.** RESTful Web Services: A Tutorial. URL: <http://www.drdoobs.com/web-development/restful-web-services-a-tutorial/240169069> (дата обращения 11.09.2016).

R. S. Zagidullin, Dr. Associate Professor, e-mail: zag@bmstu.ru,

A. S. Chernikov, Head of Educational & Research Laboratory, e-mail: chernik@bmstu.ru,

A. A. Chibisov, Programmer of the Research & Educational Centre, e-mail: chibisov.alexandr.25@gmail.com, Moscow, 105005, BMSTU

Integration of Moodle and Electronic University Systems at BMSTU

The article considers a method of constructing integrated solutions for remote laboratory practicum. Described the remote network connections technologies; management and monitoring techniques of laboratory equipment; increase computers usage efficiency. Reviewed a sample implementation of a remote laboratory practicum on electronics for the BMSTU students.

Free platform MOODLE was integrated with closed information control system Electronic University (EU) of Bauman Moscow State Technical University which serves for support of administrative work on control of educational process. The following main problems were solved:

- creation of unified data representation in the two systems;
 - creation of students' and training courses' databases in Moodle based on data from EU.
- The Moodle data storage format was brought in line with the requirements of the EU system. Unique software was developed. Combination of Moodle with EU gives new quality — all stages of University activity join together: teaching, learning and administrative control of educational process. This concept can be used by any University to reduce "human factor" errors and to increase performance.*

Keywords: Information control system, MOODLE, database, Electronic University, student, course structure, grade, human factor

References

1. **Компания, разработывающaja и поставляющaja системы автоматизации управления учебным процессом ВУЗа в Поволжье.** URL: softwareinc.ru/solutions/intranet-academic/. Data obrashhenija 11.09.2016 (in Russian).
2. **Компания, разработывающaja и поставляющaja системы автоматизации управления учебным процессом ВУЗа в Москве.** URL: www.akvadra.ru/catalog/7.html. Data obrashhenija 11.09.2016 (in Russian).
3. **Компания, разработывающaja и поставляющaja системы автоматизации управления учебным процессом ВУЗа в Индии.** URL: ampletrails.com/university-management-system-ums. Data obrashhenija 11.09.2016 (in Russian).
4. **Lucas Ambrósio B. de Oliveira, Natalia Veloso Caldas de Vasconcelos, Fernanda Cristina Barbosa Pereira Queiroz, Jamerson Viagas Queiroz and Hélio Roberto Hékis.** Contribution of Integrated Management Systems to University Management: Case Study of the Federal University of Rio Grande Do Norte // Journal of Social Sciences, 2011, no. 7 (3), pp. 415–422.
5. **Информационная система университета Менделя (Brno).** Data obrashhenija 11.09.2016. URL: is.mendelu.cz/. (in Russian).
6. **Deligirova O. A.** Информационные технологии в управлении персоналом вуза. *Perspektivy razvitiya informacionnyh tehnologij*, 2013, no. 14, pp. 67–73. (in Russian).
7. **Kochetov A. I., Krapuhina N. V., Pronichkin S. V.** Разработка систем поддержки принятия решений для управления качеством деятельности вуза. *Jekologija cheloveka*, 2009, no. 9. URL: cyberleninka.ru/article/n/razrabotka-sistem-podderzhki-prinyatiya-resheniy-dlya-upravleniya-kachestvom-deyatelnosti-vuza. Data obrashhenija 11.09.2016 (in Russian).
8. **Logvinov S. I., Romanov V. A.** Применение информационно-аналитических систем в образовательном процессе вуза: зveno "Fakul'tet — kafedra", *Sovremennye problemy nauki i obrazovanija*, 2014, no. 1. URL: cyberleninka.ru/article/n/primenenie-informatsionno-analiticheskikh-sistem-v-obrazovatelnom-protsesse-vuza-zveno-fakul'tet-kafedra. Data obrashhenija 11.09.2016 (in Russian).
9. **Ageeva T. I., Baldin A. V., Baryshnikov V. A.** i dr.; Pod. red. I. B. Fedorova, V. M. Chernen'kogo, *Informacionnaja upravljajushhaja sistema MGTU im. N. Je. Baumana "Jelektronnyj universitet": koncepcija i realizacija.* Izd-vo MGTU im. N. Je. Baumana, 2009, 376 p.
10. **Swart A. J.** Student usage of a learning management system at an open distance learning institute: A case study in electrical engineering. *International Journal of Electrical Engineering Education*, 2015, no. 52, pp. 142–154.
11. **Poncela A.** A Blended Learning Approach for an Electronic Instrumentation Course, *International Journal of Electrical Engineering*, 2013, no. 50, pp. 1–18.
12. **Gardel A., Bravo I., Revenga P. A., Lázaro J. L. and García J.** Implementation of Industrial Automation Laboratories for E-learning, *International Journal of Electrical Engineering Education*, 2012, no. 49, pp. 402–418.
13. **Btchner A.** Moodle 3 Administration. Packt Publishing, 2016, 492 p.
14. **Delisl M.** Изучение SQL с помoshh'ju PhpMyAdmin. URL: phpmyadmin.ru/learning/. Data obrashhenija 11.09.2016 (in Russian).
15. **Otkrytaja dokumentacija po sisteme Moodle 2.8.** URL: docs.moodle.org/28/en/Main_page. Data obrashhenija 11.09.2016 (in Russian).
16. **Anisimov A. M.** *Rabota v sisteme distancionnogo obuchenija Moodle.* HNAGH, 2009, 292 p. (in Russian).
17. **Vaqqas M.** *RESTful Web Services: A Tutorial.* URL: www.drdobbs.com/web-development/restful-web-services-a-tutorial/240169069. Date views 11.09.2016

УДК 004.032.26

Н. И. Юсупова, д-р техн. наук, проф., e-mail: yussupova@ugatu.ac.ru,
О. Н. Сметанина, д-р техн. наук, проф., e-mail: smoljushka@mail.ru,
М. М. Гаянова, канд. техн. наук, доцент, e-mail: maya.gayanova@gmail.com,
Уфимский государственный авиационный технический университет, Уфа

Модели и методы обработки информации при анализе образовательных программ (на примере подготовки информатиков)

Рассмотрены вопросы обработки данных об образовательных программах, представленных на сайтах вузов. Показана проблема обработки слабоструктурированных данных, приведены результаты анализа решения указанной проблемы другими исследователями, а также предложенные авторами модели, и методы обработки такого рода данных. Предложен один из подходов к структурированию учебных планов, результаты их кластерного анализа с применением нейронных сетей на основе аналитической платформы Deductor и интерпретация результатов кластеризации.

Ключевые слова: обработка информации, структура образовательной программы, кластерный анализ, нейронные сети, слабоструктурированные данные

Введение

Реформирование системы высшего образования позволило российским университетам выйти на международный рынок образовательных услуг. Для признания зарубежными университетами образовательных программ российских учебных заведений и привлечения зарубежных студентов на

обучение в Россию необходимо разрабатывать конкурентоспособные образовательные программы (ОП) и правильно представлять их на рынке образовательных услуг [1]. Привлечение иностранных студентов для обучения в российских вузах и развитие академической мобильности является одним из мониторинговых показателей деятельности вузов, в частности, таких параметров, как доля иностран-

ных студентов в вузе и доля студентов вуза, прошедших практику, включенное обучение или стажировку за рубежом.

Разработка конкурентоспособных ОП предполагает предварительное изучение опыта ведущих зарубежных университетов, проведение сопоставительного анализа ОП. Это, в свою очередь, является нетривиальной задачей, поскольку представленные на сайтах университетов ОП имеют разную степень структурированности и разные форматы представления; опубликованы на разных языках. Схожие дисциплины, включенные в ОП, имеют разные наименования или, наоборот, дисциплины с одним наименованием различаются по содержанию, по осваиваемым компетенциям; системы измерения трудоемкости дисциплин и системы оценивания знаний существенно различаются. Если одна часть перечисленных особенностей, например, системы измерения трудоемкости/оценивания знаний, число дисциплин в учебном плане характеризуются количественными показателями, и без особого труда могут быть преобразованы с учетом коэффициентов, то другая часть, может быть оценена только качественно. Часто задача сопоставительного анализа ОП может быть сведена к задаче обработки слабоструктурированных данных.

Данная статья обобщает опыт многолетних исследований авторов в области аналитической обработки информации об образовательных программах подготовки информатиков в ведущих университетах мира, которые проводились в целях совершенствования российских образовательных программ и формирования образовательных маршрутов для академической мобильности.

1. Проблемы обработки информации об образовательных программах

Обработка информации в документах, в частности, в ОП, является частью общей проблемы обработки слабоструктурированной информации. Решение проблем поиска и обработки слабоструктурированных данных отразили в своих трудах многие исследователи как зарубежные, так и российские: A. Parameswaran, R. Kaushik [2], A. Arasu [2, 3], H. Garcia-Molina [3], M. Bernstein, J. Chen, E. Chi, R. Nairn, L. Nelson [4], K.-J. Kim [5], S. B. Cho [5, 6], С. А. Афонин, С. С. Горелов [7], В. Б. Бритков, А. В. Булычев [8], А. В. Мельников [10], В. Б. Бараннин [9, 11], А. М. Федотов [11], Н. И. Юсупова [1, 12–23], О. Н. Сметанина [1, 12–23] и др.

Авторами данной работы ведутся исследования в области сопоставительного анализа образовательных программ более 15 лет [1, 12–23]. Выявлено, что ОП университетов различных стран включают ряд документов, в том числе учебные планы, представленные на официальных сайтах университетов. Они опубликованы на разных языках и в различных форматах (например, Excel, PDF или HTML),

имеют отличающиеся структуры, которые зачастую недоопределены, являются неполными или имеют исключения, т. е. представляют собой слабоструктурированные данные. При этом в учебных планах похожие по содержанию дисциплины имеют разные наименования или, наоборот, дисциплины с одним наименованием различаются по содержанию, по осваиваемым компетенциям. Также существенно различаются системы измерения трудоемкости дисциплин и системы оценивания знаний [23]. Анализ формы и содержания этих документов позволяет сделать вывод о сложности процедуры обработки информации для принятия решений [1].

Для выявления общих свойств учебных планов различных университетов путем анализа включенных в них данных возникает необходимость собрать представительное множество этих документов, что влечет за собой потребность в автоматизации процесса анализа. Автоматизация, в свою очередь, влечет за собой необходимость разработки моделей и методов, соответствующих обрабатываемым данным.

2. Модели и методы обработки информации об образовательных программах

Многие авторы публикуют результаты проведенного сопоставительного анализа образовательных программ. При анализе такого рода данных используются разные подходы и методы — от "ручного" сравнения сроков обучения, общей трудоемкости, базового образования абитуриентов до "автоматизированного" выявления скрытых закономерностей содержательной части учебных планов.

Так авторы работ [24, 25] проводят сопоставительный анализ особенностей подготовки бакалавров направления бизнес-информатика в вузах России, США и Германии, однако как таковой семантический анализ изучаемых дисциплин в исследованиях отсутствует.

Сравнительный анализ образовательных программ [26] в области бизнес-образования в вузах США и Румынии показывает необходимость реструктурировать румынский учебный план в части сокращения трудоемкости традиционных курсов по экономике и включения дисциплин, обеспечивающих умение критически мыслить и проводить стратегический анализ, а также проведения практик в реальной бизнес-среде.

Неотъемлемой задачей сравнительных исследований, по мнению авторов [27], является сопоставление в рамках квалификаций в вузах различных стран подготовки по отдельному модулю, в частности по математике. Авторами дается глубокий анализ содержания и уровня подготовки по математике в отдельных вузах Канады, Китая, Сингапура, Великобритании. В качестве инструментария используются методы экспертных оценок.

Авторы работы [28] детально исследуют образовательные программы подготовки в области математики, физики, истории, бизнеса и др. в различных вузах европейских стран (Австрии, Бельгии, Германии, Голландии, Италии, Норвегии, Швеции и др.)

Работа [29] представляет собой сравнительное исследование, в котором используется качественный контент-анализ, сравнение различных подходов к интернационализации учебных планов высшего образования в таких странах, как Соединенные Штаты Америки, Канада, Австралия и Япония.

В работе [30] приведены результаты многоаспектного сравнительного анализа учебных планов подготовки бакалавров программ в области океанологии в вузах Санкт-Петербурга, Клайпеды и Калининграда. Отмечено сходство по ряду характеристик (трудоемкость, число дисциплин и учебные практики). В качестве метода обработки данных используется кластеризация.

В работе [31] Ф. Т. Агаев и Г. А. Мамедова представляют результаты исследований в области сопоставления ОП ИКТ-специальностей ведущих университетов мира. В качестве аппарата используется Data Mining, в частности, методы поиска ассоциативных правил.

В сравнительном исследовании [32] рассматривается взаимосвязь между результатами обучения и учебной политикой, а также развитием образования в девяти европейских странах: Германии, Ирландии, Испании, Франции, Нидерландах, Польше, Румынии, Словении и Великобритании-Шотландии.

Авторы данной работы в 90-х годах прошлого века и в начале 2000 годов обработку информации об учебных планах проводили в небольших объемах, "вручную", с использованием квалифицированных экспертов, однако это не давало возможности прояснить общую картину [23]. В ходе дальнейших исследований [12–22] была создана база учебных планов университетов, ведущих подготовку в области IT-специалистов (более 100 учебных планов из восьми стран). Для автоматизации процесса обработки информации из документов, относящихся к ОП, были сформированы базы фактов, где зафиксированы результаты сходства и различия образовательных программ, что в дальнейшем позволило снизить время поиска схожих дисциплин при использовании полученных знаний в технологии рассуждений на основе прецедентов [1].

Для обработки слабоструктурированных данных требуется специальный инструментарий, т.е. модели и методы обработки информации об ОП для представления их в компьютерной среде, и программные средства для процесса обработки этих данных. Авторами работы [33] предложена новая схема представления слабоструктурированных данных. Предполагается, что данные могут быть представлены в виде помеченных графов, и между ними за-

даются понятия соответствия, т. е. структура данных определяется в виде графа.

В работе [34] предложен подход к анализу текстовых документов, основанный на извлечении ключевых словосочетаний из текста без априорных ограничений в целях дальнейшей кластеризации.

Авторами данной статьи для обработки информации исследованы и реализованы различные модели и методы:

- оценка семантической близости ключевых слов или технология регулярных выражений, которые используют для поиска подстроки, удовлетворяющей шаблону регулярного выражения в строке; поиска и замены подстроки, удовлетворяющей шаблону регулярного выражения в строке; проверки на соответствие заданной строки шаблону; извлечения подстроки, удовлетворяющей шаблону регулярного выражения из строки [16];
- модели представления знаний в виде семантических сетей для отражения результатов сопоставительного анализа ОП, преобразованных в структуру, характерных для российских ОП [1, 16, 17];
- продукционные модели для формулирования правил логического вывода, позволяющие выдавать рекомендации в системе по результатам автоматического анализа содержания учебного плана [1];
- технологии рассуждения на основе прецедентов, позволяющие адаптировать готовые решения к новой схожей ситуации [18];
- метод кластеризации на основе нейронных сетей, который позволяет соотнести схожие ОП отдельные группы по общим признакам [21].

3. Некоторые результаты обработки информации об образовательных программах подготовки информатиков

В процессе обработки информации об ОП вузов разных стран возникает необходимость сопоставления общего списка дисциплин, количественных (трудоемкость дисциплины) параметров и качественных (содержание дисциплины) характеристик [35]. Определенный интерес представляет соотношение суммарных трудоемкостей дисциплин T блоков (характеристика) ОП различных вузов и группы вузов с близкими характеристиками ОП, для которых проще процесс разработки образовательного маршрута в рамках академической мобильности.

Авторами с использованием специально разработанной информационной системы [17] проведен анализ более 100 ОП подготовки информатиков зарубежными вузами, имеющими достаточно высокий рейтинг в данной области. Среди них можно назвать Высшую политехническую школу Лозанны, Осацкий университет, Стэнфордский университет, Технический университет Карлсруэ. Поскольку структуры ОП в российских вузах в части общих трудоемкостей блоков схожи, то авторами для срав-

нения выбран Уфимский государственный авиационный технический университет (УГАТУ) [18, 19]. Все дисциплины ОП предварительно были разделены на несколько блоков: гуманитарный, социальный и экономический (Б.1), математический и естественно-научный (Б.2), профессиональный (Б.3). Результаты с указанием доли суммарной трудоемкости в процентах частично приведены на рис. 1 (см. третью сторону обложки).

Анализ этих данных показывает, что доли трудоемкости блоков в ОП представлены в разных диапазонах: гуманитарный, социальный и экономический блок имеет разброс от 1 до 34 %; математический и естественно-научный — от 9 до 37 %; профессиональный — от 45 до 84 %. Такой разброс значений усложняет процесс разработки совместных программ двойных дипломов или выбора вуза для академической мобильности, в котором, например, дисциплинам гуманитарного, социального и экономического блока ОП отводится минимальная трудоемкость, а дисциплинам профессионального цикла — максимальная. Поэтому представляет интерес задача разбиения множества вузов на группы, в которых объединены вузы с близкими характеристиками ОП. Формально эта задача сводится к одной из задач кластерного анализа.

4. Кластеризация ОП подготовки информатиков

Задача кластеризации возникает при отсутствии достаточных сведений для формирования требуемого выходного вектора [35]. Для ее решения необходимо разбить вузы на кластеры с учетом специфики структуры ОП.

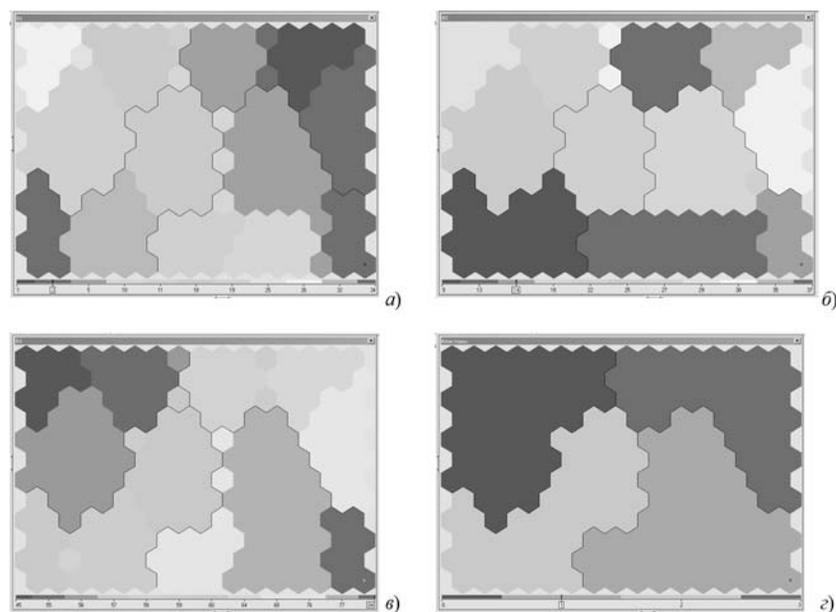


Рис. 2. Карта Кохонена: а — отображение данных Б.1; б — отображение данных Б.2; в — отображение данных Б.3; г — отображение кластеров

Для рассматриваемой задачи задано множество объектов X , представляющих различные структуры ОП, из которых выделена обучающая выборка $X^m = \{x_1, x_2, \dots, x_m\} \subset X$. Выход представлен множеством кластеров $Y = \{y_1, y_2, \dots, y_j\}$, включающих вузы со схожими структурами ОП. Задана функция расстояния между объектами $\rho(x, x')$. Требуется разбить обучающую выборку на кластеры с учетом схожести признаков объектов для одного кластера и различия признаков объектов разных кластеров. Признаками выступают доли суммарной трудоемкости блоков в учебном плане. Каждому объекту $x_i \in X^m$ приписывается номер кластера y_j . Далее каждый объект относится к одному из классов.

Для решения подобных задач разработан ряд алгоритмов и методов, в частности на основе технологий искусственного интеллекта: метод нечеткой кластеризации K -средних, генетический алгоритм и нейронная сеть Кохонена. Для проведения кластеризации авторами использована самоорганизующаяся карта Кохонена и аналитическая платформа *Deductor* [21].

Сценарий кластеризации обеспечивается выполнением следующих шагов: импорт исходных данных из файла и указание входных столбцов (столбцом 1 является значение доли T блока Б.1; столбцом 2 — Б.2; столбцом 3 — Б.3); настройка способа разделения исходного множества данных на обучающее и тестовое (соответственно, 95 и 5 %); настройка параметров карты, параметров обучения (скорость в начале обучения — 0,3 и в конце — 0,005; радиус в начале обучения — 4 и в конце — 0,1; функция соседства — ступенчатая; фиксированное число кластеров — 4) и остановки обучения (если ошибка меньше 0,05); реализация процесса обучения (число распознанных примеров и текущие значения ошибок); настройка отображений карты Кохонена (входные столбцы, кластеры).

Полученные в результате кластеризации ОП [21] отображения карты Кохонена (рис. 2) показывают, что вузы, попавшие в правую часть карты, имеют большее значение доли T профессионального блока и меньшее значение гуманитарного, социального и экономического блоков. Вузы, попавшие в верхнюю часть карты, имеют большее значение доли T математического и естественнонаучного блоков.

Для авторов было неожиданным попадание УГАТУ в один кластер с ведущими вузами США, для которых характерно, с одной стороны, то, что ОП американских университетов различаются между собой сильнее, чем российские. С другой стороны, например, в Стэнфордском университете

степень бакалавра в области информатики можно получить на нескольких направлениях, ОП которых различаются только общепрофессиональными дисциплинами. Более чем 30-летний опыт подготовки информатиков в УГАТУ позволяет говорить о том, что наши выпускники конкурентоспособны на рынке труда и образовательных услуг, в том числе и за рубежом.

Кластеры включают ОП вузов со схожими характеристиками (рис. 3, см. четвертую сторону обложки) — значениями трудоемкости T цикла: минимальное (T_{\min} : кластер 0/ цикл Б.3, 1/Б.1, 2/Б.2, 3/Б.1); максимальное (T_{\max} : 1/Б.3, 2/Б.1, 3/Б.2); среднее (T_{sr} : 3/Б.3); больше среднего ($T_{>sr}$: 1/Б.1, 1/Б.1, 3/Б.3); меньше среднего ($T_{<sr}$: 0/Б.3, 1/Б.2, 2/Б.2, 2/Б.3). Приведенные характеристики позволяют определиться с выбором принимающего вуза в рамках программ академической мобильности и разработки образовательного маршрута. Например, с T_{\max} профессионального цикла — кластер 1, или с T_{\max} математического и естественно-научного цикла — кластер 3.

5. Направления дальнейших исследований

В настоящее время, действуя в рамках государственного стандарта (ФГОС), российские вузы разрабатывают ОП с учетом определенных для конкретного направления подготовки компетенций, что дает большую свободу при формировании учебных планов. Одним из дальнейших направлений в развитии данной тематики является обработка информации об ОП для поддержки сетевого взаимодействия университетов. Опыт, накопленный при обработке информации об ОП, может быть успешно использован для выявления сходств и различий в учебных планах российских вузов. Был проведен анализ трудоемкости дисциплин нескольких университетов (УГАТУ, Уральского федерального университета (УрФУ) и Челябинского государственного университета (ЧелГУ)) по направлениям подготовки бакалавров 09.03.01 — Информатика и вычислительная техника; 09.03.04 — Программная инженерия; 02.03.03 — Математическое обеспечение и администрирование информационных систем; 38.03.05 — Бизнес-информатика.

На рис. 4 (см. четвертую сторону обложки) представлены результаты сопоставительного анализа дисциплин профессионального цикла в зачетных единицах для ОП направления 09.03.01 — Информатика и вычислительная техника.

Эти данные позволяют более тонко проводить анализ для однотипных образовательных программ на российском образовательном пространстве.

Заключение

В условиях реформирования российской высшей школы проблема сопоставимости и совершенство-

вания образовательных программ является актуальной. Вопросы аналитической обработки информации об образовательных программах связаны с проблемами обработки слабоструктурированных данных. Для решения этой проблемы используются специальные модели и методы, как правило, в основу которых положены идеи искусственного интеллекта и инженерии знаний.

Опыт авторов показывает, что модели и методы, применяемые в ходе исследования, такие как оценка семантической близости ключевых слов или технология регулярных выражений, модели представления знаний (семантические сети) для отражения результатов сопоставления и механизмы логического вывода, продукционные модели, технологии рассуждения на основе прецедентов, метод кластеризации на основе нейронных сетей эффективны при обработке информации об образовательных программах университетов разных стран и позволяют получить содержательные результаты.

Результаты, полученные в процессе обработки образовательных программ, могут быть использованы при совершенствовании образовательных программ. В условиях постоянно развивающихся информационных технологий это особенно актуально при формировании документации в случае академической мобильности (учебные соглашения, учебные планы для подготовки студентов в рамках двойных дипломов, учебные планы при сетевом взаимодействии вузов).

Исследования проводили в рамках грантов РФФИ 03-07-90242 "Интернет-комплекс поддержки выполнения проектов фундаментальных исследований сложных систем с применением интеллектуальных технологий на базе экспертных систем" (2003—2005 гг.) и 06-07-89228-а "Система поддержки коммуникативных процессов при выполнении проектов фундаментальных исследований сложных систем на основе интеллектуальных мультиагентов" (2006—2008 гг.), а также научно-исследовательской работы по теме ИФ-ВК-01-06-03 "Исследования и разработка интеллектуальных технологий поддержки принятия решений и управления на основе инженерии знаний", 09-07-00408-а "Распределенная интеллектуальная система поддержки принятия решений при выполнении проектов фундаментальных исследований сложных систем", 13-07-00273-а "Интеллектуальная поддержка принятия решений в задачах ситуационного управления сложными социально-экономическими системами".

Список литературы

1. Гузаиров М. Б., Юсупова Н. И., Сметанина О. Н., Галянова М. М. Университетские образовательные программы. Модели и методы для сопоставительного анализа. М.: Изд-во МАИ, 2006. 117 с.
2. Parameswaran A., Kaushik R., Arasu A. Efficient Parsing-based Search over Databases // CIKM '13: 22nd Int'l Conf on Information and Knowledge Management, Burlingame, USA, 2013.

3. **Arasu A., Garsia-Molina H.** Extracting Structured Data from Web Pages // *SiGMOD'03. Proceedings of the International conference on Management of data*, New York, 2003. P. 337—348.
4. **Chen J., Nairn R., Nelson L., Bernstein M., Chi E.** Short and tweet: experiments on recommending content from information streams // *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2010. P. 1185—1194.
5. **Kim K.-J., Cho S.-B.** Fuzzy integration of structure adaptive SOMs for web content mining // *Fuzzy Sets and Systems*. 2004. Vol. 148. P. 43—60. <http://sclab.yonsei.ac.kr/publications/Papers/IJ/FSS2004.pdf>
6. **Cho S.-B.** Ensemble of structure-adaptive self-organizing maps for high performance classification // *Inform. Sci.* 123, 2000. N. 1—2. P. 103—114.
7. **Афонин С. А., Хазова Е. Е., Горелов С. С.** Слабоструктурированные данные и поиск на основе онтологий // *Знания. Онтологии. Теории*. 2009. Т. 2. С. 119—123.
8. **Бритков В. Б., Бульчев А. В.** Методы анализа больших объемов слабоструктурированной информации // *Информационные технологии и вычислительные системы*. 2010. № 1. С. 36—44.
9. **Витяев Е. Е., Ковалерчук Б. К., Федотов А. М., Барахнин В. Б., Белов С. Д., Дурдин Д. С., Демин А. В.** Обнаружение закономерностей и распознавание аномальных событий в потоке данных сетевого трафика // *Вестник НГУ. Сер.: Информационные технологии*. 2008. Т. 6. Вып. 2. С. 57—68.
10. **Кровин С. Е., Мельников А. В., Кафтаников И. Л.** Моделирование семантики и прагматики документа в нотации языка XML // *Известия Челябинского научного центра УрО РАН*. 2002. № 3. С. 21—30.
11. **Шокин Ю. И., Федотов А. М., Барахнин В. Б.** Особенности организации системы управления веб-контентом сайтов информационной поддержки инновационной деятельности // *Вычислительные технологии*. 2005. Т. 10. Специальный выпуск. С. 122—128.
12. **Юсупова Н. И., Сметанина О. Н., Гаянова М. М.** Методика проведения рейтинга высших учебных заведений немецкоязычного региона, выпускающих информатиков // *Проблемы качества образования. Материалы XIII Всерос. науч.-мет. конф.*, Уфа. Изд-во Уфимс. гос. авиац. техн. ун-та, 2003. С. 81—82.
13. **Юсупова Н. И., Сметанина О. Н., Гаянова М. М.** Об особенностях подготовки информатиков в европейских университетах // *Сборник трудов "Технология и организация обучения"*. Уфа: Изд-во УГАТУ, 2004. С. 66—68.
14. **Юсупова Н. И., Сметанина О. Н., Гаянова М. М.** Технические университеты: интеграция с европейскими и мировыми системами образования: О качестве подготовки информатиков в европейских университетах. Ижевск: Изд-во ИжГТУ, 2004. С. 278—283.
15. **Юсупова Н. И., Сметанина О. Н., Гаянова М. М., Маркелова А. В.** Анализ образовательных программ с применением информационных технологий (на примере образовательных программ УГАТУ и университета СУСЕККА). Технологии и организация обучения. Уфа: Изд-во УГАТУ, 2008. С. 60—69.
16. **Юсупова Н. И., Сметанина О. Н., Маркелова А. В.** Инструментальные средства для сопоставительного анализа образовательных программ на основе регулярных грамматик // *Вестник УГАТУ*. 2010. Т. 14, № 5 (40). С. 150—156.
17. **Юсупова Н. И., Сметанина О. Н., Гаянова М. М., Маркелова А. В.** Информационная поддержка принятия решений для управления образовательным процессом на этапе разработки образовательных программ // *Интеллектуальные системы управления (коллективная монография в рамках европейской программы TEMPUS / под ред. акад. С. Н. Васильева. М.: Машиностроение, 2010. С. 447—454.*
18. **Гузайров М. Б., Юсупова Н. И., Сметанина О. Н.** Информационное и математическое обеспечение в системе поддержки принятия решений при управлении процессом разработки образовательной программы. М.: Машиностроение, 2011. 247 с.
19. **Гузайров М. Б., Юсупова Н. И., Сметанина О. Н., Козырева В. А.** Поддержка принятия решений при управлении академической мобильностью // *Системы управления и информационные технологии*. 2011. № 3.1. С. 131—136.
20. **Юсупова Н. И., Сметанина О. Н., Климова А. В.** Интеллектуальная информационная поддержка при управлении образовательным маршрутом // *Инфокоммуникационные технологии в науке, производстве и образовании: матер. V Междунар. науч.-техн. конф.*, Ч. II, Ставрополь, 2012. С. 144—148.
21. **Гузайров М. Б., Юсупова Н. И., Сметанина О. Н., Галева Н. И.** Инструментарий нейронных сетей при поддержке принятия решений по управлению образовательным маршрутом // *Нейрокомпьютеры: разработка, применение*. 2013. № 3. С. 21—26.
22. **Юсупова Н. И., Сметанина О. Н., Гаянова М. М.** Моделирование системы поддержки принятия решений при управлении образовательным маршрутом в рамках программ академической мобильности // *Информационные технологии и системы: Труды 2-й Междунар. конф.*, Банное, Россия. Челябинск: Изд-во Челябинск. гос. ун-та, 2013. С. 197—200.
23. **Юсупова Н. И., Сметанина О. Н., Гаянова М. М.** Инструменты анализа зарубежных и российских образовательных программ (на примере подготовки информатиков) // *Мавлютовские чтения: материалы Российской научно-технической конференции, посвященной 90-летию со дня рождения член-корр. РАН, д-ра техн. наук, профессора Р. Р. Мавлютова: сб. трудов в 7 т. Том 1. Проблемы образования в России при подготовке кадров для авиадвигателестроения*. Уфа: Изд-во УГАТУ, 2016. С. 74—79.
24. **Ivanova V., Kazakova E., Lezina T. A., Martyanova V. N., Saltan A. A.** Comparing bachelor Studies in business-informatics at Universities in Russia and Germany // *Вестник Российского университета дружбы народов. Серия: Информатизация образования*. 2015. № 4. С. 123—140.
25. **Ivanova V., Lezina T. A.** Evolution of Business Informatics as a Science: The European, American and Russian Perspective // *Economics and management*. 2014. № 02 (100). P. 44—50.
26. **Brătianu C., Christopher L.** A comparative analysis of the american and romanian business education programs. URL: <http://www.managementmarketing.ro/pdf/articole/10.pdf>.
27. **Bergeron L., Alcántara A.** IB mathematics comparability study: curriculum & assessment comparison // *University of Wisconsin-La Crosse report for ib global recognition*, September 2015. URL: <http://www.ibo.org/globalassets/publications/ib-research/dp/math-comparison-summary-report.pdf>.
28. **Tuning Educational Structures in Europe** // *Closing Conference*. Brussels, 31 may 2002. URL: http://www.aic.lv/ace/bologna/Prg_berl/Tun_Book.pdf.
29. **Ardakani F. B., Yarmohammadian M. H., Abari A. F., Fathid K.** Internationalization of higher education systems // *Procedia Social and Behavioral Sciences*. 15, 2011. P. 1690—1695.
30. **Gritsenko V. A., Zhurakhovskaya P. M., Yurov A. V.** Similarities and differences in curricula of a bachelor's degree in oceanology at the universities in St. Petersburg, Klaipeda and Kaliningrad // *Балтийский регион*. 2012. № 4. С. 80—89.
31. **Агаев Ф. Т., Мамедова Г. А.** Сравнительный анализ образовательных программ ИКТ-специальностей университетов мира с применением методов поиска ассоциативных правил // *Informasiya texnologiyalari problemlari*. 2014. № 2. С. 40—45. URL: <http://www.jpit.az/storage/files/article/e9a38ba38406f912b5aa05f988698e53.pdf>
32. **Learning outcomes approaches in VET curricula. A comparative analysis of nine European countries**. Luxembourg: Publications Office of the European Union, 2010. VI. 169 p. URL: <http://www.cedefop.europa.eu/>.
33. **Buneman P., Davidson S., Fernandea M., Suciu D.** Adding structure to unstructured data // *In Proceedings of ICDT*, January 1997.
34. **Барахнин В. Б., Ткачев Д. А.** Кластеризация текстовых документов на основе составных ключевых термов // *Вестник НГУ. Сер.: Информационные технологии*. 2010.
35. **Панфилова А. С.** Система тестирования интеллекта на базе факторных моделей и самоорганизующихся карт Кохонена // *Нейрокомпьютеры: разработка и применение*. 2012. № 9. С. 6—12.

N. I. Yusupova, DSc., Professor, e-mail: yussupova@ugatu.ac.ru,
O. N. Smetanina, DSc., Professor, e-mail: smoljushka@mail.ru,
M. M. Gayanova, PhD, Associate Professor, e-mail: maya.gayanova@gmail.com,
Ufa State Aviation Technical University, Ufa

Models and Methods of Information Processing in the Analysis of Educational Programs (on the Example of Training Computer Scientists)

The results of many years researching by the authors on the analysis of educational programs are presented to identify the positive experience of training IT professionals in leading foreign universities in this field (Lausanne Higher Polytechnic School, Osaka University, Stanford University, Karlsruhe Technical University, etc.). Another goal of the comparative analysis of educational programs is the possibility of forming educational routes within the framework of academic mobility.

The general features of educational programs for the training of computer scientists are reflected, both in the content aspect and in the form of presentation. It is noted that the differences in the form (structure, format (Excel, PDF, HTML), presentation language, etc.) and significant volumes of the collected documentary base make the procedure for analyzing educational programs a difficult and complex task (evaluating the semantic proximity of keywords; finding a substring that satisfies the pattern of regular expression; mapping the results of comparative analysis of educational programs in the form of knowledge representation models to provide solutions; identifying similar structures of educational programs; adaptation of ready solutions to a similar situation on the basis of reasoning based on precedents, etc.). It is revealed that some of the problems are related to the processing of weakly structured data.

The results of the educational programs analysis by other researchers are given, the models and methods used for analysis are noted. Among the analysis the following tools are revealed: methods of expert assessments, qualitative content analysis, clustering methods, methods of searching for associative rules, etc.

The results of processing educational programs information for the training of computer scientists are presented to identify their similar structures using clustering methods. The results were obtained using the analytic platform Deductor. For students in higher education institutions, whose educational programs have similar structures (they are in the same cluster), an educational route with participation in academic mobility programs can be more comfortable.

The possibility of using the results of comparative educational programs of Russian universities to support networking of universities is noted.

Thus, the issues of analytical processing of information about educational programs are related to the problems of processing poorly structured data. To solve this problem, special models and methods are used, usually based on ideas of artificial intelligence and knowledge engineering.

The possibility of using the results of the comparative analysis of educational programs of Russian universities is noted to support network interaction of universities.

Keywords: data processing, structure of the educational program, cluster analysis, neural networks, weakly structured data

References

1. **Guzairov M. B., Yusupova N. I., Smetanina O. N., Gayanova M. M.** *Universitetskie obrazovatel'nye programmy. Modeli i metody dlya sopostavitel'nogo analiza.* (Models and methods for comparative analysis). M.: MAI, 2006. 117 p.
2. **Parameswaran A., Kaushik R., Arasu A.** Efficient Parsing-based Search over Databases, *CIKM '13: 22nd Int'l Conf on Information and Knowledge Management*, Burlingame, USA, 2013.
3. **Arasu A., Garsia-Molina H.** Extracting Structured Data from Web Pages, *SiGMOD'03 Proceedings of the International Conference on Management of Data, New York, 2003*, pp. 337–348.
4. **Chen J., Nairn R., Nelson L., Bernstein M., Chi E.** Short and tweet: experiments on recommending content from information streams, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM*, 2010, pp. 1185–1194.
5. **Kim K.-J., Cho S.-B.** Fuzzy integration of structure adaptive SOMs for web content mining, *Fuzzy Sets and Systems*, 2004, vol. 148, pp. 43–60. URL: <http://sclab.yonsei.ac.kr/publications/Papers/IJ/FSS2004.pdf>
6. **Cho S.-B.** Ensemble of structure-adaptive self-organizing maps for high performance classification, *Inform. Sci.*, 2000, vol. 123, no. 1–2, pp. 103–114.
7. **Afonin S. A., Khazova E. E., Gorelov S. S.** Slabostrukturirovannye dannye i poisk na osnove ontologii, *Znaniya. Ontologii. Teorii.* (Weakly structured data and ontology-based search) Knowledge. Ontologies. Theories. 2009, vol. 2, pp. 119–123.
8. **Britkov V. B., Bulychev A. V.** Metody analiza bol'shikh ob'emov slabostrukturirovannoy informatsii, *Informatsionnye tekhnologii i vychislitel'nye sistemy.* (Methods for analyzing large volumes of weakly structured information, Information Technologies and Computing Systems. 2010, no. 1, pp. 36–44.
9. **Vityaev E. E., Kovalerchuk B. K., Fedotov A. M., Barakhnin V. B., Belov S. D., Durdin D. S., Demin A. V.** Obnaruzhenie zakonomernostey i raspoznavanie anomal'nykh sobytii v potoke dannykh setevogo trafika, *Vestnik NGU. Ser.: Informatsionnye tekhnologii.* (Detection of regularities and recognition of anomalous events in the stream of network traffic data, Vestnik NSU. Ser.: Information technology.) 2008, vol. 6, no. 2, pp. 57–68.
10. **Krovin S. E., Mel'nikov A. V., Kaftannikov I. L.** Modelirovanie semantiki i pragmatiki dokumenta v notatsii yazyka XML, *Izvestiya Chelyabinskogo nauchnogo tsentra UrO RAN.* (Modeling of semantics and pragmatics of the document in XML notation, Proceedings of the Chelyabinsk Scientific Center, Ural Branch of the Russian Academy of Sciences.) 2002, no. 3, pp. 21–30.
11. **Shokin Yu. I., Fedotov A. M., Barakhnin V. B.** Osobennosti organizatsii sistemy upravleniya veb-kontentom saytov informatsionnoy podderzhki innovatsionnoy deyatel'nosti, *Vychislitel'nye tekhnologii.* (Features of the organization a management system of web content of information support sites of innovation activity, Computational technologies.) 2005, vol. 10, special issue, pp. 122–128.
12. **Yusupova N. I., Smetanina O. N., Gayanova M. M.** Metodika provedeniya reytinga vysshikh uchebnykh zavedeniy nemetskoyazychnykh

nogo regiona, vypuskayushchikh informatikov. Problemy kachestva obrazovaniya. Materialy KhIII Vseros. nauch.-met. konf. Ufa. M.: Izd. Ufimsk. gos. aviats. tekhn. un-ta (Methodology for conducting a rating of higher educational institutions in the German-speaking region that produce computer science. *Problems of the quality of education. Materials XIII All-Russia scientific-methodical conference Ufa*, M.: Publ. Ufa State Aviation Technical University), 2003, pp. 81–82.

13. **Yusupova N. I., Smetanina O. N., Gayanova M. M.** Ob osobennostyakh podgotovki informatikov v evropeyskikh universitetakh, *Sbornik trudov "Tekhnologiya i organizatsiya obucheniya". Min. obrazovanie RF, UGATU, Ufa* (On the specifics of training informatics in European universities. Collection of works "Technology and organization of training." Min. Education of the Russian Federation, UGATU, Ufa) 2004, pp. 66–68.

14. **Yusupova N. I., Smetanina O. N., Gayanova M. M.** Tekhnicheskie universitety: integratsiya s evropeyskimi i mirovymi sistemami obrazovaniya: O kachestve podgotovki informatikov v evropeyskikh universitetakh, Izhevsk, Izd-vo IzhGTU (Technical universities: integration with European and world education systems: On the quality of training of informatics in European universities, Izhevsk, Publishing house and printing house of IzhSTU), 2004, pp. 278–283.

15. **Yusupova N. I., Smetanina O. N., Gayanova M. M., Markelova A. V.** Analiz obrazovatel'nykh programm s primeneniem informatsionnykh tekhnologiy (na primere obrazovatel'nykh programm UGATU i universiteta SUSSEKSA). Tekhnologii i organizatsiya obucheniya. Ufa, GOU VPO "Ufimskiy gosudarstvennyy aviatsionnyy tekhnicheskiy universitet". (Analysis of educational programs using information technologies (using the example of educational programs of UGATU and Sussex University). Technologies and organization of training. Ufa, Ufa State Higher Technical University "Ufa State Aviation Technical University".) 2008, pp. 60–69.

16. **Yusupova N. I., Smetanina O. N., Markelova A. V.** Instrumental'nye sredstva dlya sopostavitel'nogo analiza obrazovatel'nykh programm na osnove regul'yarnykh grammatik, *Vestnik UGATU*. nauch. zhurn. Ufimsk. gos. aviats. tekhn. un-ta. UGATU (Tools for comparative analysis of educational programs on the basis of regular grammars // Bulletin of the USATU of the Ufa State Aviation Technical University.) 2010, vol. 14, no. 5 (40), pp. 150–156.

17. **Yusupova N. I., Smetanina O. N., Gayanova M. M., Markelova A. V.** Informatsionnaya podderzhka prinyatiya resheniy dlya upravleniya obrazovatel'nym protsessom na etape razrabotki obrazovatel'nykh programm. *Intellektual'nye sistemy upravleniya (kollektivnaya monografiya v ramkakh evropeyskoy programy TEMPUS*, pod red. akad. S. N. Vasil'eva, Moscow, Mashinostroenie. (Informational support for decision-making for the management of the educational process at the stage of developing educational programs. Intelligent control systems (collective monograph in the framework of the European program TEMPUS, ed. acad. S. N. Vasilyev, Moscow: Mechanical engineering) 2010, pp. 447–454.

18. **Guzairov M. B., Yusupova N. I., Smetanina O. N.** Informatsionnoe i matematicheskoe obespechenie v sisteme podderzhki prinyatiya resheniy pri upravlenii protsessom razrabotki obrazovatel'noy programmy, Moscow, Mashinostroenie. (Informational and mathematical support in the decision support system when managing the development of the educational program.) 2011, pp. 247.

19. **Guzairov M. B., Yusupova N. I., Smetanina O. N., Kozyreva V. A.** Podderzhka prinyatiya resheniy pri upravlenii akademicheskoy mobil'nost'yu, *Sistemy upravleniya i informatsionnye tekhnologii*. (Support of decision-making in the management of academic mobility, Control Systems and Information Technology), 2011, no. 3.1, pp. 131–136.

20. **Yusupova N. I., Smetanina O. N., Klimova A. V.** Intellektual'naya informatsionnaya podderzhka pri upravlenii obrazovatel'nym marshrutom. Infokommunikatsionnye tekhnologii v nauke, proizvodstve i obrazovanii: *mater. V Mezhdunar. nauch.-tekhn. konf., Ch. II, Stavropol'*. (Intelligent information support for the management of the educational route. Infocommunication technologies in science, production and education: proceedings V International Scientific-technical Conference, Part II, Stavropol'), 2012, pp. 144–148.

21. **Guzairov M. B., Yusupova N. I., Smetanina O. N., Galeeva N. I.** Instrumentariy neyronnykh setey pri podderzhke prinyatiya resheniy po upravleniyu obrazovatel'nym marshrutom, *Neyrokomp'yutery: razrabotka, primenenie* (Instrumentation of neural networks with the

support of decision-making on the management of the educational route. The journal "Neurocomputers: development, application"), 2013, no. 3, pp. 21–26.

22. **Yusupova N. I., Smetanina O. N., Gayanova M. M.** Modelirovanie sistemy podderzhki prinyatiya resheniy pri upravlenii obrazovatel'nym marshrutom v ramkakh programm akademicheskoy mobil'nosti. Informatsionnye tekhnologii i sistemy: *Trudy 2-y Mezhdunar. konf., Bannoe, Rossiya. Chelyabinsk*: Izd-vo Chelyabinsk.gos. un-ta (Modeling the decision support system for the management of the educational route in the framework of academic mobility programs. Information Technologies and Systems: Proceedings of the 2nd International Conference, Bannoe, Russia. Chelyabinsk: Publishing house Chelyabinsk State University), 2013, pp. 197–200.

23. **Yusupova N. I., Smetanina O. N., Gayanova M. M.** Instrumenty analiza zarubezhnykh i rossiyskikh obrazovatel'nykh programm (na primere podgotovki informatikov) (Tools for analyzing foreign and Russian educational programs (on the example of training informatics)). *Mavlyutovskie chteniya: materialy Rossiyskoy nauchno-tekhnicheskoy konferentsii, posvyashchennoy 90-letiyu so dnya rozhdeniya chlen-korr. RAN, d-ra tekhn. nauk, professora R. R. Mavlyutova: sb. trudov v 7 t. Tom 1. Problemy obrazovaniya v Rossii pri podgotovke kadrov dlya aviadvigatelestroeniya*, Ufimsk. gos. aviats. tekhn. un-t. (Ufa), 2016, pp. 74–79.

24. **Ivanova V., Kazakova E., Lezina T. A., Martyanova V. N., Saltan A. A.** Comparing bachelor Studies in business-informatics at Universities in Russia and Germany, *Bulletin of the Russian University of Peoples' Friendship. Series: Informatization of education*, 2015, no. 4, pp. 123–140.

25. **Ivanova V., Lezina T. A.** Evolution of Business Informatics as a Science: The European, American and Russian Perspective, *Economics and management*, 2014, no. 02 (100), pp. 44–50.

26. **Brătianu C., Christopher L.** *A comparative analysis of the american and romanian business education programs*, URL: <http://www.managementmarketing.ro/pdf/articole/10.pdf>.

27. **Bergeron L., Alcántara A.** IB mathematics comparability study: curriculum & assessment comparison, University of Wisconsin-La Crosse report for ib global recognition, September 2015, URL: <http://www.ibo.org/globalassets/publications/ib-research/dp/maths-comparison-summary-report.pdf>.

28. **Tuning Educational Structures in Europe**, *Closing Conference. Brussels, 31 may 2002*. URL: http://www.aic.lv/ace/bologna/Prg_berl/Tun_Book.pdf.

29. **Ardakani F. B., Yarmohammadian M. H., Abari A. F., Fathid K.** Internationalization of higher education systems. *Procedia Social and Behavioral Sciences*, 2011, vol. 15, pp. 1690–1695.

30. **Gritsenko V. A., Zhurakhovskaya P. M., Yurov A. V.** Similarities and differences in curricula of a bachelor's degree in oceanology at the universities in St. Petersburg, Klaipeda and Kaliningrad, *The Baltic region*, 2012, no. 4, pp. 80–89.

31. **Agaev F. T., Mamedova G. A.** Sravnitel'nyy analiz obrazovatel'nykh programm IKT-spetsial'nostey universitetov mira s primeneniem metodov poiska assotsiativnykh pravil (Comparative analysis of educational programs of ICT specialties of universities in the world with the use of methods of searching for associative rules), *Informatsiya texnologiyalari problemlari*, 2014, no. 2, pp. 40–45. URL: <http://www.jpit.az/storage/files/article/e9a38ba38406f912b5aa05f988698e53.pdf>

32. **Learning outcomes approaches in VET curricula. A comparative analysis of nine European countries**, Luxembourg: Publications Office of the European Union, 2010, is. VI, 169 p. URL: <http://www.cedefop.europa.eu/>.

33. **Buneman P., Davidson S., Fernandea M., Suci D.** Adding structure to unstructured data, *In Proceedings of ICDT, January 1997*.

34. **Barakhnin V. B., Tkachev D. A.** Klasterizatsiya tekstovykh dokumentov na osnove sostavnykh klyuchevykh termov, *Vestnik NGU, Ser.: Informatsionnye tekhnologii*. [Clustering text documents based on composite key terms, Vestnik NSU. Ser.: Information technology.]. 2010.

35. **Panfilova A. S.** Sistema testirovaniya intellekta na baze faktornykh modeley i samoorganizuyushchikhsya kart Kokhonena, *Neyrokomp'yutery: razrabotka i primenenie*. (Intelligence testing system based on factor models and Kohonen self-organizing maps, Neurocomputers: development and application.), 2012, no. 9, pp. 6–12.

И. Л. Гаврилова, старший преподаватель кафедры электротехники и электроники, igavr@grsu.by
Учреждение образования "Гродненский государственный университет имени Янки Купалы",
г. Гродно, Республика Беларусь

Опыт разработки и внедрения в учебный процесс вуза электронного учебно-методического комплекса

Представлен опыт разработки и внедрения в учебный процесс электронного учебно-методического комплекса "Антенно-фидерные устройства". Описаны основные элементы комплекса, представлена его структура. Рассмотрен модульный подход в организации комплекса. Отмечены особенности разработки виртуальной лаборатории, а также компьютерной системы тестирования. Представлен опыт проведения управляемой самостоятельной работы студентов. Отмечены достоинства разработанного комплекса.

Ключевые слова: учебно-методический комплекс, электронное средство обучения, среда проектирования, самостоятельная работа, образовательный процесс, тестирование, дистанционное образование

Введение

Основная цель Положения [1] об учебно-методическом комплексе на уровне высшего образования, утвержденного Постановлением Министерства образования Республики Беларусь от 26 июля 2011 г., заключается во введении единых требований к научно-методическому обеспечению дисциплин, а также систематизации процесса использования в образовательном процессе университета электронных учебно-методических комплексов (ЭУМК).

Актуальность исследования заключается в использовании единых технических средств в образовательном процессе. В современных условиях информатизации разработка и внедрение электронных учебно-методических комплексов позволит повысить качество образования, эффективность учебной деятельности студентов, активизировать их самостоятельную работу [2]. ЭУМК предназначен для обеспечения реализации учебных целей и задач на всех этапах образовательного процесса, а также формирования профессиональных компетенций по учебным дисциплинам в соответствии с образовательными стандартами по специальности. ЭУМК создается на научно-методическом и программно-техническом уровне в соответствии с современными информационными и коммуникационными технологиями [3], [4].

Объект исследования

Электронный учебно-методический комплекс по дисциплине "Антенно-фидерные устройства", адресованный студентам специальности "Промышленная электроника" физико-технического факультета Учреждения образования "Гродненский государственный университет имени Янки Купалы" представляет собой программный комплекс всех компонентов учебного процесса изучаемой дисциплины. Комплекс обеспечивает необходимую наглядность, модульность, взаимосвязь учебных модулей и других элементов, компактность, универ-

сальность, предоставляет возможность пользователю самостоятельного выбора учебных модулей, разнообразие проверочных заданий и тестов, что соответствует основным принципам разработки ЭУМК.

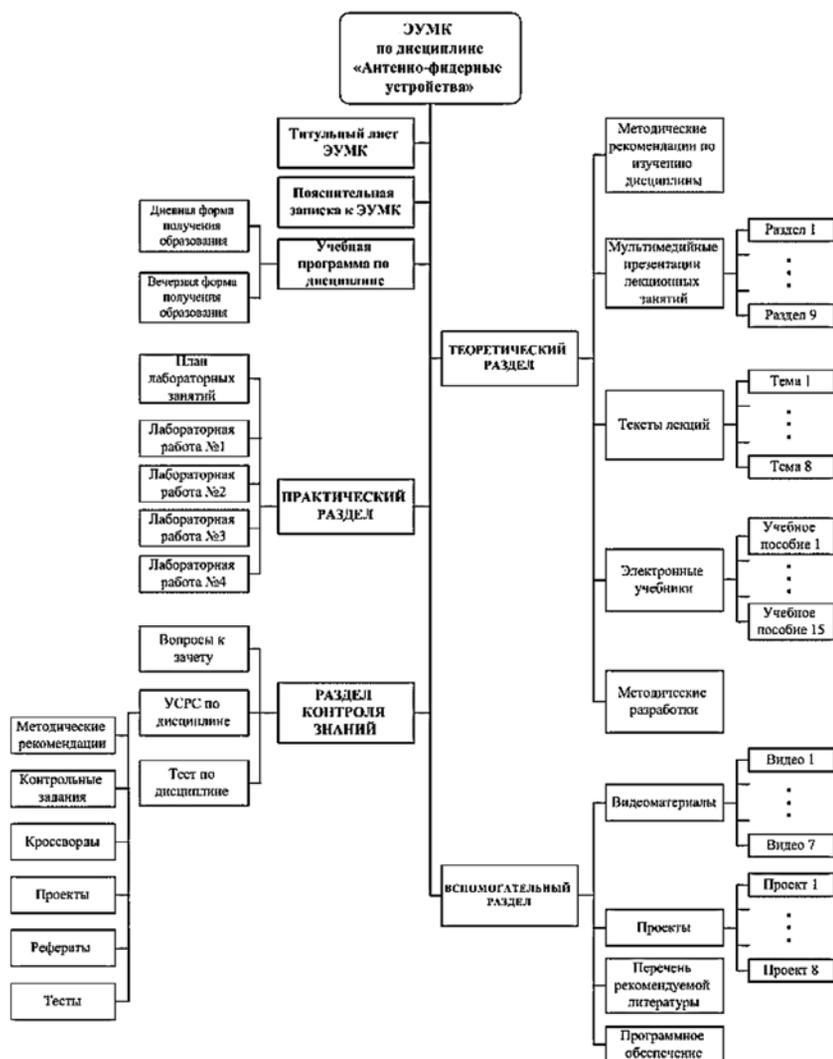
Элементами ЭУМК являются: титульный лист; пояснительная записка; учебные программы по дисциплине; теоретический раздел; практический раздел; раздел контроля знаний и вспомогательный раздел. ЭУМК разбит на соответствующие модули, общим объемом 1,85 Гбайт.

На рисунке представлена структура ЭУМК "Антенно-фидерные устройства".

Теоретический раздел комплекса содержит методические рекомендации по изучению дисциплины, мультимедийные презентации лекционных занятий, тексты лекций, электронные учебники, а также методические разработки по изучению дисциплины. Практический раздел представляет собой виртуальный лабораторный комплекс, который содержит план лабораторных занятий и методические рекомендации по выполнению лабораторных работ. В разделе контроля знаний приведены вопросы к зачету по изучаемой дисциплине, задания на управляемую самостоятельную работу студентов (УСРС) в виде контрольных заданий по расчету антенн, тематики проектов, рефератов, тестовых заданий, кроссвордов, методические рекомендации по организации и проведению УСРС, контрольные вопросы по основным темам, проверочный электронный тест. Вспомогательный раздел содержит видеоматериалы по антенной технике, проекты по темам дисциплины, перечень рекомендуемой литературы, программное обеспечение MMANA-GAL по моделированию антенн, а также мастер кроссвордов для самостоятельной разработки студентами заданий в форме кроссвордов.

Модульная технология обучения

Особенность модульного подхода в организации комплекса, ориентированного на дистанционный



Структура ЭУМК "Антенно-фидерные устройства"

способ обучения, заключается в повышении качества и эффективности подготовки студентов, формировании универсальных и профессиональных навыков [5, 6]. В комплексе четко определены цели обучения, задачи и уровни изучения каждого модуля, определены навыки и умения, а также контроль качества усвоения материала.

Актуальность разработки состоит в том, что при подготовке студентов радиотехнических специальностей осуществляется приобретение ими навыков научно-исследовательской работы, а также способности решать возникающие инженерные задачи. Комплекс содержит: печатные пособия; технические средства обучения (среда моделирования); обучающие программы; тестовые задания; задания на УСРС. Материал для теоретического изучения дисциплины представлен в виде конспекта лекций, электронного учебника. Практический раздел организован по видам учебной работы, предусмотренной учебным планом по специальности, учебной программой, он представляет собой виртуальный лабораторный комплекс связанных программных и

мультимедийных средств, моделирующих опытную установку и физический процесс. Самостоятельная работа студентов рассматривается как организуемая преподавателем внеаудиторная деятельность, выстроенная с учетом особенностей и знаний студентов.

Перечисленные возможности комплекса становятся доступными и распространяемыми благодаря разработке и внедрению современных образовательных технологий. Образовательные технологии представляют и воспитательный аспект, связанный с формированием и развитием личностных качеств. Особенность модульной технологии обучения состоит в том, что сумма знаний, предназначенных для передачи студентам по каждой конкретной дисциплине, разделяется на самостоятельные тематические блоки, которые оформляются в виде отдельных учебных модулей [7]. Система учебных задач модуля соединяет в одно целое материал разных тем и научных направлений, но в каждом из модулей не теряется содержание дисциплины, ее сущность, и также прослеживается логика конкретной науки, особенности путей ее развития.

Виртуальная лаборатория

Средства информационных технологий, к которым можно отнести виртуальные лаборатории, позволяют увеличить степень самостоятельности студента при усвоении учебного материала, повысить наглядность изучения темы, провести моделирование, оптимизировать контроль знаний и умений студентов.

Для разработки виртуальной лаборатории, представленной в работах [8, 9], была создана специализированная среда с использованием языка программирования Borland Delphi 6. Компоненты виртуальной лаборатории включают в себя: теоретический материал; задания по лабораторным работам; графические редакторы; редакторы схем; программы моделирования; таблицы; средства мультимедиа; контрольные вопросы и тесты; требования к оформлению отчета (шаблоны отчета).

Виртуальная лаборатория представляет собой электронную среду. Создать виртуальные лаборатории можно и с помощью уже готовых образовательных платформ, таких как LMS MOODLE, LabVIEW, E-Learning, Claroline, OLAT, CCNet, а также можно создать свою образовательную платформу с помощью различных языков программирования: Visual Basic, C++, Borland Delphi. Выполнять лабораторные задания можно с помощью специализированного программного обеспечения, например, MatLab, MatCad, Proteus.

Применение разработанного виртуального лабораторного комплекса в рамках ЭУМК позволяет создать универсальные и легко перенастраиваемые учебные работы по разным разделам дисциплины. Выполняя лабораторные работы, студенты получают дополнительную информацию о параметрах и характеристиках исследуемых объектов. Кроме этого, студенты имеют возможность изменять параметры исследуемых антенн и наблюдать результаты моделирования на мониторе компьютера. Среду можно применять для любых дисциплин и курсов, она позволяет исследовать и строить графики зависимостей, сравнивать данные, а также с ее помощью можно сделать расчеты и оформить отчет, что упрощает работу студентов и проверку работ преподавателем. Данный способ обучения позволяет студенту вспомнить, повторить и закрепить свои знания по дисциплине с помощью контрольных вопросов и тестов.

Компьютерная система тестирования

Не менее важным элементом комплекса является компьютерная система тестирования. Компьютерная система тестирования знаний как неотъемлемый компонент дистанционного обучения по сравнению с традиционным тестированием имеет ряд преимуществ: возможность индивидуального контроля знаний; регулярность проведения контроля; объективность оценки знаний тестируемых; единство требований, предъявляемых к обучаемым; легкость обработки результатов. Важную роль при проектировании инструментальной тестовой среды играет выбор программ для проверки знаний, разработки и создания тестов. В настоящее время существует большое разнообразие программ, созданных для тестирования результатов образовательного процесса в учебных заведениях. Наиболее известными программами для разработки тестов являются ADSoftTester, EasyTest, Knowing.

Среда тестирования по дисциплине "Антенно-фидерные устройства" [10] разработана с использованием языка программирования Delphi. Среда позволяет присоединить краткий конспект, статистические данные, учебный материал по изучаемой теме, предназначенный для самообучения. Использование в тестовых заданиях графической, звуковой, видеоинформации — одно из преимуществ данной программы. Тестовые задания представлены четырех типов: задание закрытого типа с выбором одного или нескольких правильных ответов; задание в открытой форме (на дополнение); задание на установление правильной последовательности; задание на установление соответствия.

В результате была получена компактная интерактивная среда компьютерного тестирования, способная дистанционно оценить знания студентов. Преимущества среды: возможность индивидуального контроля знаний; объективность оценки тестируемых; единство требований, предъявляемых к обучаемым; легкость обработки результатов. Сле-

дует отметить, что применение компьютерных тестов в образовании является безусловно экономически выгодным и эффективным способом повышения результативности учебного процесса. Гибкость и мобильность разработанной тестовой среды позволяет применить ее к любым дисциплинам и использовать как компонент разработки электронного учебно-методического комплекса. Таким образом, компьютерное тестирование выступает как инструмент управления учебным процессом, дает возможность анализировать и осуществлять контроль над процессом обучения.

Организация управляемой самостоятельной работы студентов

Проведению самостоятельной работы студентов в рамках дисциплины "Антенно-фидерные устройства" отводится особая роль. В современных условиях организации образовательного процесса основная задача заключается в формировании творческой личности студента, способного к саморазвитию, самообразованию и инновационной деятельности. Значимая роль при реализации данной задачи отводится при управляемой самостоятельной работе студентов (УСРС). УСРС позволяет студентам планировать изучение дисциплины с учетом индивидуальных способностей, а также более рационально использовать свое личное время.

В Положении о самостоятельной работе студентов РБ [11] отмечено, что управляемая самостоятельная работа студентов — это организованная целенаправленная деятельность преподавателя и студента, ориентированная на развитие познавательной и интеллектуальной активности студентов, мотивацию приобретения новых знаний, опыта творческой и исследовательской деятельности, формирование и совершенствование практических умений и навыков. Учебный процесс должен быть направлен на поиски путей и условий для непрерывного, динамического, многогранного развития самостоятельной и активной личности. Формирование навыков и умений самостоятельной работы является главной задачей не только учебной, но и профессиональной деятельности. Конкретные формы проведения и виды контроля УСРС определяются с учетом курса обучения, особенности изучаемой дисциплины, а также творческой деятельности преподавателя, что позволяет преподавателю провести индивидуальный подбор заданий, сформировать свою систему планирования самостоятельной работы.

УСРС по учебной дисциплине "Антенно-фидерные устройства" предусматривает различные формы проведения занятий, как аудиторные, так и индивидуальные. В работе [12] представлен опыт проведения УСРС. В рамках данной дисциплины со студентами проводят различные мероприятия: выполнение исследовательских и творческих заданий, связанных с проектированием антенн, определением характеристик; решение задач по расчету параметров антенн;

подготовка сообщений, тематических докладов, рефератов, презентаций, проектов; выполнение практических заданий, связанных, например, с моделированием антенной техники; конспектирование учебной литературы; подготовка отчетов по результатам исследований; составление тестов, кроссвордов по основным разделам дисциплины. Студентам предлагается участие в оформлении информационных стендов и демонстрационных материалов по антенной технике. В целях наглядного изучения антенн со студентами проводятся экскурсии по городу, свои видеонаблюдения студенты демонстрируют в отчетах.

В ЭУМК приведены все необходимые методические рекомендации по выполнению практических заданий, перечни контрольных заданий по расчету основных типов антенн, учебная, справочная, методическая литература, типовые задания, контрольные работы, примеры решения задач, тестовые задания, тематика проектов, мультимедийные и видеоматериалы и др.

В учебной лаборатории для проведения занятий представлены наглядные пособия, конструкции антенн, контрольно-измерительная аппаратура. Отдельные виды антенных устройств, работающих в различном диапазоне частот, выполнены студентами в рамках курсового и дипломного проектирования и являются действующими для проведения ряда исследований, например, измерения интенсивности микроволнового излучения мобильных телефонов. Стоит отметить, что результативность УСРС во многом зависит от системности и регулярности проведения контроля деятельности студентов. На освоение учебного материала в рамках УСРС отводится 20 % аудиторных часов, предусмотренных типовым учебным планом на изучение данной дисциплины. При проведении контроля УСРС по дисциплине "Антенно-фидерные устройства" предусмотрены такие формы контроля, как самостоятельная работа, проведение тестирования; обсуждение рефератов, защита учебных заданий и творческих работ, опрос на аудиторных занятиях. В рамках проведения контроля выполнения заданий УСРС используется рейтинговая система оценки знаний по учебной дисциплине. Набранная студентом сумма баллов служит допуском к экзамену (или зачету), позволяет провести аттестацию и влияет на итоговую оценку.

Заключение

ЭУМК "Антенно-фидерные устройства" позволяет активизировать различные формы восприятия и усвоения студентами учебного материала, способствует повышению интеллектуальной активности студентов, что обеспечивает формирование глубоких знаний, прочных умений и навыков, а также более быстрый процесс адаптации к профессиональной деятельности в современном мире.

К достоинствам разработанного ЭУМК следует отнести: возможность эффективной организации самостоятельной работы студента, индивидуальный

подход к обучению, наглядность, многофункциональность, поддержка мультимедийных возможностей, простая навигация, не требует инсталляции, возможность систематического обновления и добавления учебного материала, возможность работы в удаленном доступе.

Представленный опыт разработки и внедрения ЭУМК позволяет сформулировать требования к содержанию электронных учебно-методических комплексов по радиотехническим дисциплинам и может быть полезен при разработке новых учебно-методических комплексов в образовательном процессе вуза.

Разработанный ЭУМК "Антенно-фидерные устройства" прошел процедуру утверждения ЭУМК в 2015 году и внесен в Государственный регистр информационных ресурсов — рег. свид. № 4141607886 от 18.05.2016 г. [9].

Список литературы

1. **Положение** об учебно-методическом комплексе на уровне высшего образования, утв. Постановлением Министерства образования Республики Беларусь от 26 июля 2011 г., № 167. Минск, 2011. 15 с.
2. **Концепция** информатизации системы образования Республики Беларусь на период до 2020 г.: [Электронный ресурс]: утв. постановлением Министра образования Респ. Беларусь, 24 июня 2013 г.
3. **Балакирева Э. В., Власова Е. З.** Электронный учебно-методический комплекс как средство обеспечения качества подготовки специалистов // ЧиО. 2012. № 4 (33).
4. **Лозицкий В.Л.** Электронный учебно-методический комплекс по дисциплинам социально-гуманитарного цикла. Научно-методические основы создания и системного применения. Минск: РИВШ, 2012. 224 с.
5. **Гаврилова И. Л.** Учебный модуль по курсу "Антенно-фидерные устройства" // Современные информационные технологии в системе научного и учебного эксперимента: опыт, проблемы, перспективы: материалы Респ. науч. метод. конф. Гродно: ГрГУ, 2011. С. 59—61.
6. **Гаврилова И. Л.** Модель структуры виртуального курса // Современные информационные технологии в системе научного и учебного эксперимента: опыт, проблемы, перспективы: материалы Республ. науч. метод. конф. Гродно: ГрГУ, 2011. С. 11—13.
7. **Васюкевич В. В.** Электронный учебно-методический комплекс на основе модульно-рейтинговой технологии обучения // Известия РГПУ им. А. И. Герцена. 2008. № 74-2.
8. **Гаврилова И. Л., Карачарская М. Л.** Специализированная среда разработки для лабораторного практикума // Современные информационные технологии в системе научного и учебного эксперимента: опыт, проблемы, перспективы: материалы II Республ. науч. метод. конф. Гродно: ГрГУ, 2013. С. 72—74.
9. **Гаврилова И. Л., Дедович А. Р.** Оценка инновационных образовательных систем организации виртуальных лабораторий // Электронные засоби та дистанційні технології для навчання протягом життя: тези доповідей IX Міжнародної науково-методічної конференції. Суми: Сумський державний університет, 2013. С. 13—15.
10. **Гаврилова И. Л., Коронкевич О. В.** Компьютерное тестирование как инструмент управления учебным процессом // Информационно-технологическое обеспечение образовательного процесса современного университета [Электронный ресурс]: сб. докл. Междунар. интернет-конф. Минск: БГУ, 2014. С. 117—119.
11. **Положение** о самостоятельной работе студентов (курсантов, слушателей), утв. Приказом Министра образования РБ от 27 мая 2013 г., № 405. Минск, 2013. 5 с.
12. **Гаврилова И. Л., Карачарская М. Л.** Управляемая самостоятельная работа студентов как способ развития познавательной активности // Современные информационные технологии в системе научного и учебного эксперимента: опыт, проблемы, перспективы: материалы III междунар. науч. метод. конф. Гродно: ГрГУ, 2015. С. 26—29.
13. **Гаврилова И. Л.** Антенно-фидерные устройства: Электронный учебно-методический комплекс для студентов специальности 1-360402 "Промышленная электроника", специализации 1-36040202 "Техника и средства электронной связи" [Электронный ресурс]. Электрон., текстовые дан. и прогр. (1,85 Гб). УО "Гродненский государственный университет имени Янки Купалы": Гаврилова И. Л., 2015. 1 электрон. опт. диск (CD-R).

Experience in Development and Implementation of the Learning Process of the University Electronic Educational Complex

Administering to the subject of research — process of training of students in conditions of informatization of educational process. General provisions are considered.

The purpose of the study is to show experience in the development and implementation into study electronic educational complex on discipline "Antenna-feeder devices". Relevance of the research is concluded in using common technical tools in the educational process.

Experience in the development and implementation into study electronic educational complex on discipline "Antenna-feeder devices" is shown in the main part of the work. Main elements of the complex are described in the beginning of the work, structure is also shown. Content of the main parts are disclosed. Modular approach in complex organization is viewed. Features of the technical training are marked.

Special attention is paid for creature specialized sector for working out virtual laboratory which applies to practical part of the complex. Components of the virtual laboratory are described in the work. Universality of the designed industry is considered for conduction practical classes in any discipline and its application.

One of the main element of the complex is designing computer system of testing knowledge. One of the advantage of this program is diversity of the tasks and also using in testing tasks graphic, sound, videomaterials.

In section of the control of knowledge in the complex main role drained to conduction controlled work of students. Experience of the designed work is shown. Different shapes of conduction classes are shown.

Advantages of the designed complex are described in the conclusion.

Received results of the study afford formulate requirements to the content of the electronic complex and can be used in elaboration new methodical complexes in university educational process.

Keywords: methodical complex, e-learning, development environment, independent work, educational process, testing, remote education

References

1. **Polozhenie** ob uchebno-metodicheskom komplekse na urovne vysshego obrazovanija (Regulations on the educational complex at the level of higher education), utv. Postanovleniem Ministerstva obrazovanija Respubliki Belarus' ot 26 ijulja 2011, no. 167. Minsk, 2011. 15 p. (in Russian).
2. **Koncepcija** informatizacii sistemy obrazovanija Respubliki Belarus' na period do 2020 g.: [Jelektronnyj resurs]: utv. postanovleniem Ministra obrazovanija Resp. Belarus', 24 ijunja 2013. (The concept of informatization of the education system of the Republic of Belarus for the period until 2020: [Electronic resource]: Decree of the Minister of Education of the Republic. Belarus, June 24, 2013.) (in Russian).
3. **Balakireva Je.V., Vlasova E. Z.** Jelektronnyj uchebno-metodicheskiy kompleks kak sredstvo obespechenija kachestva podgotovki specialistov (Electronic educational-methodical complex as a means of ensuring the quality of training specialists), *ChiO*. 2012, no. 4 (33) (in Russian).
4. **Lozickij V. L.** Jelektronnyj uchebno-metodicheskiy kompleks po disciplinam social'no-gumanitarnogo cikla. Nauchno-metodicheskie osnovy sozdaniya i sistemnogo primeneniya (Electronic educational and methodical complex on the disciplines of the social and humanitarian cycle. Scientifically-methodical bases of creation and system), Minsk: RIVSh, 2012. 224 p. (in Russian).
5. **Gavrilova I. L.** Uchebnyj modul' po kursu "Antenno-fidernye ustrojstva" (Training module on the course "Antenna-feeder devices"), *Informacionnye tehnologii v sisteme nauchnogo i uchebnogo jeksperimenta: opyt, problemy, perspektivy*: materialy Resp. nauch. metod. konf. Grodno: GrGU, 2011, pp. 59–61 (in Russian).
6. **Gavrilova I. L.** Model' struktury virtual'nogo kursa (The model of the virtual course structure), *Sovremennye informacionnye tehnologii v sisteme nauchnogo i uchebnogo jeksperimenta: opyt, problemy, perspektivy*: materialy Respubl. nauch. metod. konf. Grodno: GrGU, 2011, pp. 11–13. (in Russian).
7. Vasjukevich V. V. Jelektronnyj uchebno-metodicheskiy kompleks na osnove modul'no-rejtingovoj tehnologii obuchenija (Electronic educational-methodical complex on the basis of the modular rating technology of education), *Izvestija RGPU im. A. I. Gercena*. 2008, no. 74–2 (in Russian).
8. **Gavrilova I. L., Karacharskaja M. L.** Specializirovannaja sreda razrabotki dlja laboratornogo praktikuma (Specialized development environment for laboratory practical), *Sovremennye informacionnye tehnologii v sisteme nauchnogo i uchebnogo jeksperimenta: opyt, problemy, perspektivy*: materialy II Respubl. nauch. metod. konf. Grodno: GrGU, 2013, pp. 72–74 (in Russian).
9. **Gavrilova I. L., Dedovich A. R.** Ocenka innovacionnyh obrazovatel'nyh sistem organizacii virtual'nyh laboratorij (Evaluation of innovative educational systems of virtual laboratories), *Elektronni zasobi ta distancijni tehnologii dlja navchannja profjagom zhittja*: tezi dopovidej IX Mizhnarodnoi naukovo-metodichnoi konferencii. Sumi: Sums'kij derzhavnij universitet, 2013, pp. 13–15 (in Russian).
10. **Gavrilova I. L., Koronkevich O. V.** Komp'juternoe testirovanie kak instrument upravlenija uchebnym processom (Computer testing as a learning management tool), *Informacionno-tehnologicheskoe obespechenie obrazovatel'nogo processa sovremennogo universiteta* sb. dokl. mezhdunar. internet-konf. Minsk: BGU, 2014, pp. 117–119 (in Russian).
11. **Polozhenie** o samostojatel'noj rabote studentov (kursantov, slushatelej) (Regulations on independent work of students (students, trainees)), utv. Prikazom Ministra obrazovanija RB ot 27 maja 2013, no. 405. Minsk, 2013. 5 p. (in Russian).
12. **Gavrilova I. L., Karacharskaja M. L.** Upravljaemaja samostojatel'naja rabota studentov kak sposob razvitiya poznavatel'noj aktivnosti (Guided independent work of students as a way of cognitive activity development), *Sovremennye informacionnye tehnologii v sisteme nauchnogo i uchebnogo jeksperimenta: opyt, problemy, perspektivy*: materialy III mezhdunar. nauch. metod. konf. Grodno: GrGU, 2015, pp. 26–29 (in Russian).
13. **Gavrilova I. L.** Antenno-fidernye ustrojstva: Jelektronnyj uchebno-metodicheskiy kompleks dlja studentov special'nosti 1-360402 "Promyshlennaja jelektronika", specializacii 1-36040202 "Tehnika i sredstva jelektronnoj svjazi" (Antenna-feeder devices: Electronic educational methodical complex.), *Jelektron., tekstovye dan. i progr.* (1,85 Gb). UO "Grodenskij gosudarstvennyj universitet imeni Janki Kupaly: Gavrilova I. L., 2015. 1 jelektron. opt. disk (CD-R)." (in Russian).

Диспетчеризация массивов с заявками параболического типа в Grid-системах

Исследовано функционирование Grid-систем централизованной архитектуры с мультисайтным режимом обслуживания заявки при диспетчеризации массивов с заявками параболического типа, оптимальная укладка которых в квадратную оболочку содержит пустоты. Анализируется адаптированность уровневых полиномиальных алгоритмов распределения вычислительно-временных ресурсов при обслуживании указанных множеств требований пользователей.

Ключевые слова: Grid-система централизованной архитектуры, мультисайтный режим обслуживания заявки, среда ресурсных прямоугольников, квадратичный тип заявки, заявка параболического типа, неэвклидова эвристическая мера, полиномиальная трудоемкость алгоритма, уровневые алгоритмы.

Введение

Эффективность функционирования Grid-систем во многом определяется политикой диспетчеризации (*scheduling policy*) [1], реализуемой системой управления ресурсами и диспетчеризации заданий (*resource management and job scheduling system*) [2]. В основе политики диспетчеризации лежит алгоритм диспетчеризации (*scheduling algorithm*), задающий способ назначения заявок на ресурсы [1]. Алгоритм диспетчеризации определяет стадию создания расписания (*schedule generation*) [2]. Диспетчеризация ресурсами (*resource scheduling*) включает обнаружение ресурсов (*resource discovery*), выбор ресурсов (*resource selection*) и выполнение заявки (*job execution*) [3]. Эти же три стадии в работах [4, 5] относят к Grid-диспетчеризации (*Grid scheduling*). В работе [2] Grid-диспетчеризация представляется состоящим из всех четырех вышеперечисленных стадий: обнаружение и выбор ресурсов, создание расписания и выполнение заявки. В работе [6] Grid-диспетчеризация содержит пять стадий: подготовка и сбор информации по поступившим заявкам (*preparation and information gathering on tasks submitted to the Grid*); выбор ресурсов, создание расписания (*computation of the planning of tasks*); размещение заявки (*task allocation*) и контроль хода решения заявки (*task execution monitoring*).

В работе [7] авторы отмечают, что в литературе по Grid-вычислениям (*Grid computing*) диспетчеризация заданий (*job scheduling*) представляется как часть управления ресурсами (*resource management*) и обращают внимание на важность их различать: диспетчер (*scheduler*) фокусируется на выполнении расписания и управлении заданиями (*execution planning and job control*), а менеджер ресурсов (*resource*

manager) — на предоставлении доступа к ресурсам по требованию (*on-demand resource provisioning*). Далее рассмотрим стадию создания расписания.

Заявка пользователя характеризуется двумя параметрами: числом требуемых процессоров (*number of required processors*, *degree of parallelism*, *job size*) и временем решения заявки (*processing time*, *execution time*, *units of time*). Полагаем число требуемых процессоров не меняющимся в процессе выполнения (*rigid job*) [8]. При заранее известном времени решения заявки для диспетчеризации используется термин *clairvoyant scheduling* [9—11]. Далее рассматриваются заявки пользователей с известным временем решения (*clairvoyant rigid jobs*) [10], считающимся целым числом [12—15].

В работе [16] было рассмотрено диспетчеризование с прерываниями (*preemptive scheduling*) и без прерываний (*non-preemptive scheduling*), когда выделенные по заявке процессоры удерживаются в течение всего времени ее выполнения от начала до окончания (*run-to-completion*). Полагаем, что заявка выполняется без прерывания (*non-preemptive rigid jobs*) [13].

В работе [17] заявки назначались как на процессоры с произвольными номерами, так и на процессоры с последовательными номерами (*contiguous processors addresses*). Далее рассматриваем диспетчеризование с последовательными номерами (*contiguous scheduling*) [18] и заявки, выполняемые на процессорах с подряд идущими номерами (*contiguous parallel tasks*) [14].

Следовательно, заявку пользователя при обслуживании диспетчером Grid-системы можно представлять ресурсным прямоугольником с горизонтальным и вертикальным измерениями, соответственно, равными числу единиц ресурса времени и процессо-

ров, требуемых для выполнения заявки [13]. Символом $[a(j), b(j)]$ будем обозначать j -ю заявку, требующую $a(j)$ единиц времени и $b(j)$ единиц процессоров.

Постановка задачи

Grid-системы централизованной архитектуры [19, 20], состоящие из сайтов, содержащих параллельные системы, с возможностью выполнения многопроцессорной заявки на нескольких сайтах одновременно [20, 21], моделируются ресурсным квадратом [22, 23].

Оптимальное распределение вычислительно-временных ресурсов имеет экспоненциальную трудоемкость, поэтому практическую ценность представляют эвристические алгоритмы полиномиальной трудоемкости. В качестве аппарата теории диспетчеризации полиномиальной трудоемкости в [22, 24, 25] предложена среда ресурсных прямоугольников. В среде ресурсных прямоугольников введены операции над ресурсными прямоугольниками и предложены эвристические полиномиальные алгоритмы распределения ресурсов, основанные на введенных операциях [22, 24–28]. Полиномиальные алгоритмы адаптированы под соответствующий квадратичный тип массива заявок. Квадратичный тип для массивов из не менее двух заявок определен в работе [22]. Квадратичный тип одной заявки введен в работе [29]. Качество диспетчеризации массивов с заявками кругового типа исследовалось в [29], с заявками гиперболического типа — в [28]. Массив ресурсных прямоугольников с беспустотной квадратной ресурсной оболочкой, называемый точной формой, определен в работе [30]. Качество диспетчеризации массивов точной формы с заявками параболического типа изучали в работе [31], где тестовыми примерами служили массивы прямоугольников, индуцированные диссекцией квадрата на последовательные домино и тромино, диссекцией соотношения сторон, облицовкой квадрата полосами из меньших квадратов, ханойской облицовкой квадрата.

В настоящей работе ставится вопрос об адаптированности полиномиальных алгоритмов для массивов с заявками параболического типа, оптимальная укладка которых в квадратную оболочку содержит пустоты.

Диспетчеризация массива с заявками параболического типа

Качество диспетчеризации эвристических алгоритмов оценивается неевклидовой эвристической мерой, учитывающей площадь и форму занятой

ресурсной области $\frac{1}{2} \left(\frac{LH + (L - H)^2}{\sum_{j=0}^{k-1} a(j)b(j)} \right)$, где L — про-

тяженность; H — уровень по вертикали ресурсной оболочки (рис. 1) [22].

Эвристическая мера достигает минимума, равного 0,5, при беспустотной укладке в квадрат.

Тестовым примером является массив прямоугольников, не имеющих одинаковых сторон (*no sides of rectangles are the same*) [32], называемых также прямоугольниками, различных размеров (*rectangles with distinct sizes*) [33], $[j, 2k - j + 1], j = 1, 2, \dots, k$. Для $k = 12, 13, 14$ укладки таких массивов прямоугольников в квадрат с минимальной стороной получены в работе [33, с. 370] (рис. 2).

Условие отнесения прямоугольника к параболическому типу [29] $(b(j) - a(j))^2 > a(j)b(j)$ для прямоугольников $[j, 2k - j + 1], j = 1, 2, \dots, k$, приводит к неравенству $j < (2k + 1) \frac{\sqrt{5} - 1}{2\sqrt{5}}$. Так, для $k = 14$

имеем $j < 8,02$. Таким образом, прямоугольники $j = 1, 2, \dots, 8$ относятся к параболическому типу.

Массивы ресурсных прямоугольников различных размеров, упорядоченные по убыванию высот, обозначим следующим образом: для $k = 12$ — массив I; $k = 13$ — массив II; $k = 14$ — массив III.

Следующим тестовым примером служит массив ресурсных прямоугольников равного периметра (*constant-perimeter rectangles* [34], *equal-perimeter rectangles* [35]) $[j, k - j], j = 1, 2, \dots, k - 1$.

Приведенное выше условие отнесения прямоугольника к параболическому типу для прямоугольников $[j, k - j], j = 1, 2, \dots, k - 1$, приводит

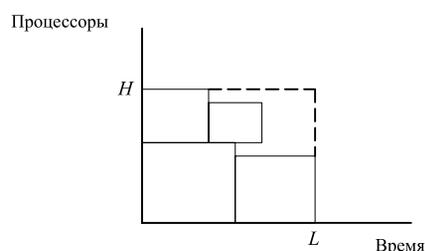


Рис. 1. Ресурсная оболочка заявок пользователей

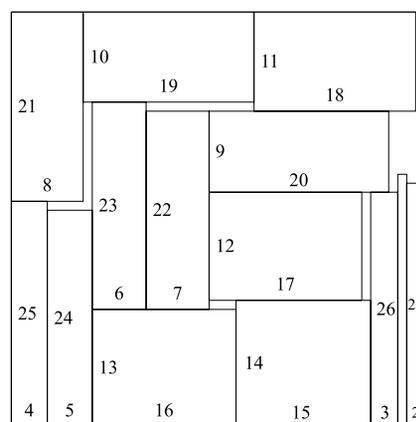


Рис. 2. Укладка 14 прямоугольников различных размеров в квадрат с минимальной стороной [33]

к неравенствам $j < \frac{\sqrt{5}-1}{2\sqrt{5}}k$ и $j > \frac{\sqrt{5}-1}{2\sqrt{5}}k$. Так, для $k = 24$ имеем $j < 6,63$ и $j > 17,37$. Таким образом, прямоугольники $j = 1, 2, \dots, 6$ и $j = 18, 19, \dots, 23$ относятся к параболическому типу.

Массивы ресурсных прямоугольников равного периметра, упорядоченные по убыванию высот, обозначим следующим образом: для $k = 21$ — массив IV; $k = 22$ — массив V; $k = 23$ — массив VI; $k = 24$ — массив VII.

Вычислим эвристические меры ресурсных оболочек, получаемых при диспетчировании этих массивов V -уровневыми алгоритмами по высоте и

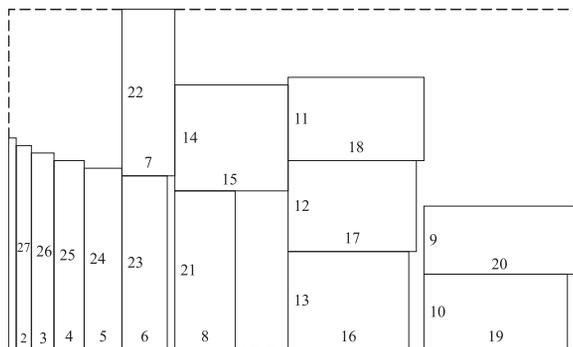


Рис. 3. Укладка массива III V -уровневым алгоритмом по высоте с недостатком

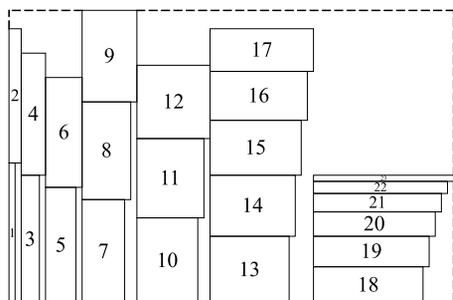


Рис. 4. Укладка массива VII V -уровневым алгоритмом по высоте с недостатком

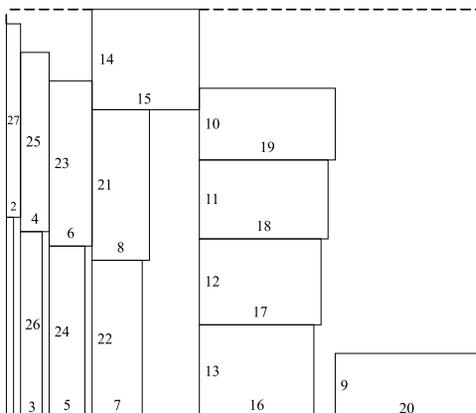


Рис. 5. Укладка массива III V -уровневым алгоритмом по высоте с превышением

H -уровневыми алгоритмами по протяженности, определенными в работе [28]: с недостатком, с превышением и с минимальным отклонением.

Для массивов III, VII построения V -уровневым алгоритмом по высоте с недостатком приведены на рис. 3, 4, по высоте с превышением — на рис. 5, 6, по высоте с минимальным отклонением — на рис. 7, 8. В центре прямоугольника на рис. 4, 6, 8 указано значение его горизонтального измерения.

Эвристические меры ресурсных оболочек V -уровневого алгоритма по высоте для массивов I—VII приведены в табл. 1.

Видим, что эвристические меры ресурсных оболочек V -уровневого алгоритма по высоте с недос-

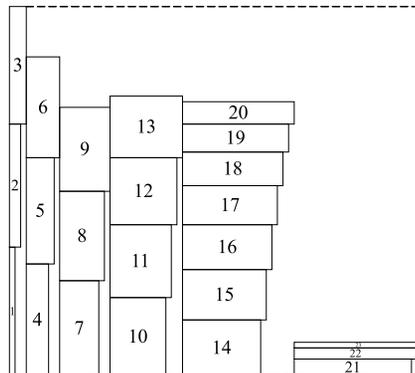


Рис. 6. Укладка массива VII V -уровневым алгоритмом по высоте с превышением

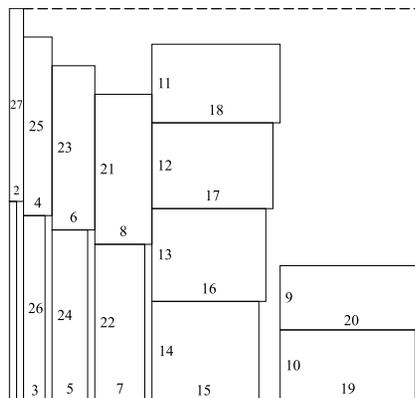


Рис. 7. Укладка массива III V -уровневым алгоритмом по высоте с минимальным отклонением

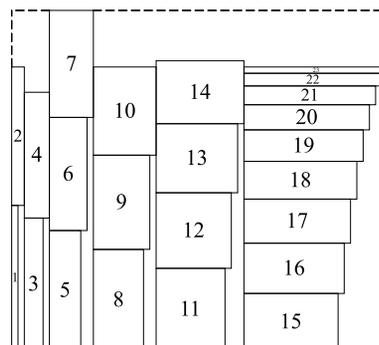


Рис. 8. Укладка массива VII V -уровневым алгоритмом по высоте с минимальным отклонением

Таблица 1

Эвристические меры ресурсных оболочек *H*-уровневого алгоритма

| Номер массива | По высоте с недостатком | По высоте с превышением | По высоте с минимальным отклонением |
|---------------|-------------------------|-------------------------|-------------------------------------|
| I | 0,89 | 0,79 | 0,83 |
| II | 0,82 | 0,97 | 0,86 |
| III | 1,05 | 0,95 | 0,79 |
| IV | 0,76 | 0,78 | 0,9 |
| V | 0,7 | 0,9 | 0,83 |
| VI | 1,01 | 0,85 | 0,76 |
| VII | 0,9 | 1,08 | 0,71 |

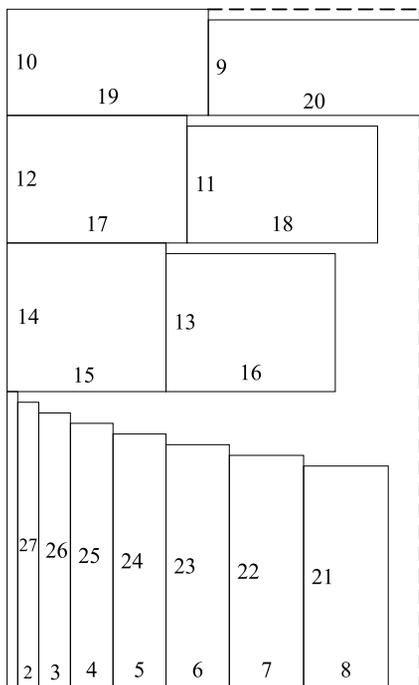


Рис. 9. Укладка массива III *H*-уровневым алгоритмом по протяженности с недостатком

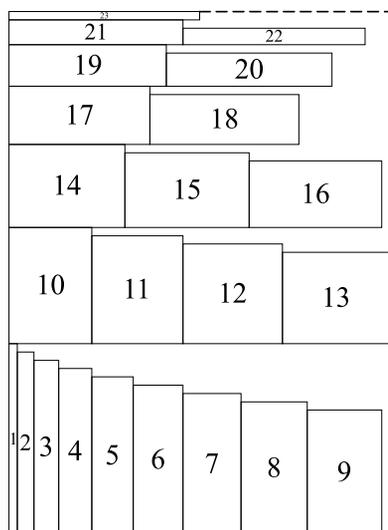


Рис. 10. Укладка массива VII *H*-уровневым алгоритмом по протяженности с недостатком

татком не превосходят значения $0,5 + 0,55$, по высоте с превышением — $0,5 + 0,58$ и по высоте с минимальным отклонением — $0,5 + 0,36$.

Для массивов III, VII построения *H*-уровневым алгоритмом по протяженности с недостатком приведены на рис. 9, 10, по протяженности с превышением — на рис. 11, 12, по протяженности с минимальным отклонением — на рис. 13, 14.

Эвристические меры ресурсных оболочек *H*-уровневого алгоритма по протяженности для массивов I—VII приведены в табл. 2.

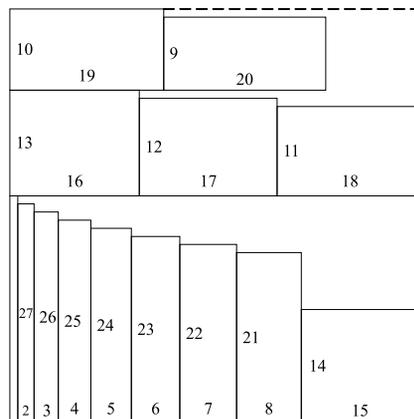


Рис. 11. Укладка массива III *H*-уровневым алгоритмом по протяженности с превышением

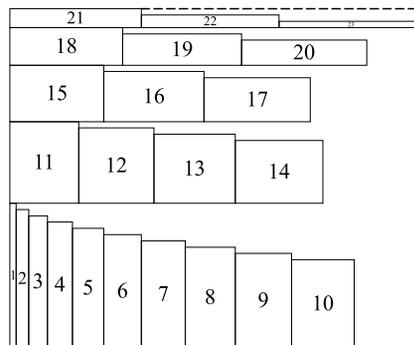


Рис. 12. Укладка массива VII *H*-уровневым алгоритмом по протяженности с превышением

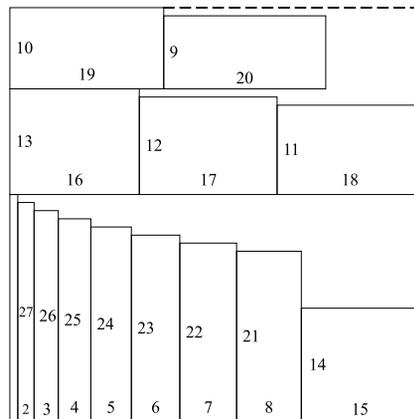


Рис. 13. Укладка массива III *H*-уровневым алгоритмом по протяженности с минимальным отклонением

Видим, что эвристические меры ресурсных оболочек H -уровневого алгоритма по протяженности с недостатком не превосходят значения $0,5 + 0,35$, по протяженности с превышением — $0,5 + 0,35$ и по протяженности с минимальным отклонением — $0,5 + 0,27$.

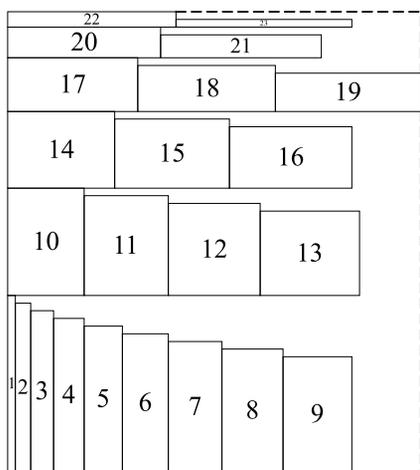


Рис. 14. Укладка массива VII H -уровневым алгоритмом по протяженности с минимальным отклонением

Таблица 2

Эвристические меры ресурсных оболочек H -уровневого алгоритма

| Номер массива | По протяженности с недостатком | По протяженности с превышением | По протяженности с минимальным отклонением |
|---------------|--------------------------------|--------------------------------|--|
| I | 0,77 | 0,74 | 0,77 |
| II | 0,85 | 0,75 | 0,73 |
| III | 0,77 | 0,64 | 0,64 |
| IV | 0,73 | 0,79 | 0,68 |
| V | 0,7 | 0,74 | 0,65 |
| VI | 0,74 | 0,85 | 0,72 |
| VII | 0,69 | 0,81 | 0,71 |

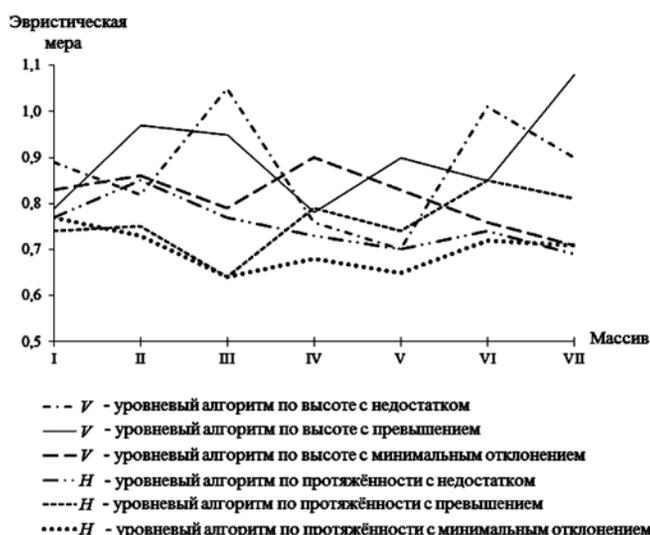


Рис. 15. Эвристические меры ресурсных оболочек уровневых алгоритмов

Графики эвристической меры ресурсных оболочек V -уровневыми алгоритмами по высоте (с недостатком, с превышением, с минимальным отклонением) и H -уровневыми алгоритмами по протяженности (с недостатком, с превышением, с минимальным отклонением) при диспетчеризации массивами I—VII показаны на рис. 15.

Видим, что H -уровневый алгоритм по протяженности с минимальным отклонением имеет наименьшее значение максимума эвристической меры $0,5 + 0,27$ на рассматриваемых тестовых массивах ресурсных прямоугольников. Уровневые полиномиальные алгоритмы диспетчеризации массивов, содержащих заявки параболического типа, могут быть использованы в Grid-системах с централизованной структурой.

Заключение

В настоящей работе оценивается качество различных полиномиальных уровневых алгоритмов при диспетчеризации массивов с заявками параболического типа, оптимальная укладка которых в квадратную оболочку содержит пустоты. На тестовых массивах показана адаптированность исследуемых алгоритмов. Рассматриваемые полиномиальные алгоритмы диспетчеризации рекомендуется применять при обслуживании массивов с заявками параболического типа в Grid-системах.

Список литературы

1. Pugliese A., Talia D., Yahyapour R. Modeling and Supporting Grid Scheduling // J Grid Computing. 2008. Vol. 6, Is. 2. P. 195—213.
2. Li M., Baker M. The grid: core technologies. Chichester: John Wiley & Sons Ltd, 2005. 452 p.
3. Qureshi M., Dehnavi M., Min-Allah N., Qureshi M., Hus-sain H., Rentifis I., Tziritas N., Loukopoulos T., Khan S., Xu C., Zomaya A. Survey on Grid resource allocation mechanisms // J. Grid Computing. 2014. Vol. 12, Is. 2. P. 399—441.
4. Schopf J. Ten actions when Grid scheduling // In Grid resource management-state of the art and future trends / Nabrzyski J. et al. (eds.), Springer US: Kluwer Academic Publishers, 2004. 574 p., pp. 15—23.
5. Pop F. Grid Scheduling: Methods, Algorithms, and Optimization Techniques // In Computational and data Grids: principles, applications and design. Preve N. (ed.), Hershey: IGI Global, 2012, 401 p., pp. 86—111.
6. Khafa F., Abraham A. Computational models and heuristic methods for Grid scheduling problems // Future Generation Computer Systems. 2010. Vol. 26, Is. 4. P. 608—621.
7. Siddiqui M., Fahringer T. Grid resource management: on-demand provisioning, advance reservation, and capacity planning of Grid resources. LNCS. Vol. 5951. Heidelberg: Springer-Verlag Berlin, 2010. 227 p.
8. Feitelson D., Rudolph L., Schwiegelshohn U., Sevcik K., Wong P. Theory and practice in parallel job scheduling // LNCS. 1997. Vol. 1291, pp. 1—34.
9. Tchernykh A., Schwiegelshohn U., Yahyapour R., Kuzjurin N. Online hierarchical job scheduling on Grids // From Grids to service and pervasive computing / Eds. T. Priol, M. Vanneschi. Springer: Science + Business Media, 2008. P. 77—91.
10. Pascual F., Rzacca K., Trystram D. Cooperation in multi-organization scheduling // Concurrency Computat.: Pract. Exper. 2009. Vol. 21, Is. 7. P. 905—921.

11. Dutot P., Pascual F., Rządca K., Trystram D. Approximation algorithms for the multiorganization scheduling problem // IEEE Trans. on Paral. and Distrib. Syst. 2011. Vol. 22, N. 11. P. 1888—1895.
12. Bougeret M., Dutot P., Jansen K., Otte C., Trystram D. A Fast 5/2-approximation algorithm for hierarchical scheduling // LNCS. 2010. Vol. 6271. P. 157—167.
13. Bougeret M., Dutot P., Trystram D., Jansen K., Robenek C. Improved approximation algorithms for scheduling parallel jobs on identical clusters // Theoretical Computer Science. 2015. Vol. 600. P. 70—85.
14. Błażdek I., Drozdowski M., Guinand F., Schepler X. On contiguous and non-contiguous parallel task scheduling // J. Sched. 2015. Vol. 18, Is. 5. P. 487—495.
15. Caramia M., Giordani S., Iovanella A. Grid scheduling by on-line rectangle packing // Networks. 2004. Vol. 44, N. 2. P. 106—119.
16. Błażewicz J., Drabowski M., Węglarz J. Scheduling multiprocessor tasks to minimize schedule length // IEEE Trans. on Comput. 1986. Vol. c-35, N. 5. P. 389—393.
17. Turek J., Wolf J., Yu P. Approximate algorithms for scheduling parallelizable tasks // In Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'92). 1992. San Diego, CA. P. 323—332.
18. Günther E., König F., Megow N. Scheduling and packing malleable and parallel tasks with precedence constraints of bounded width // J. Comb. Optim. 2014. Vol. 27, Is. 1. P. 164—181.
19. Rahman M., Ranjan R., Buyya R., Benatallah B. A taxonomy and survey on autonomic management of applications in grid computing environments // Concurrency Computat.: Pract. Exper. 2011. Vol. 23, Is. 16. P. 1990—2019.
20. Hamscher V., Schwiigelshohn U., Streit A., Yahyapour R. Evaluation of job-scheduling strategies for grid computing // LNCS. 2000. Vol. 1971. P. 191—202.
21. Sonmez O., Mohamed H., Epema D. On the benefit of processor coallocation in multicluster grid systems // IEEE Trans. on Paral. and Distrib. Syst. 2010. Vol. 21, N. 6, P. 778—789.
22. Саак А. Э. Полиномиальные алгоритмы распределения ресурсов в Grid-системах на основе квадратичной типизации массивов заявок // Информационные технологии. 2013. № 7. Приложение. 32 с.
23. Саак А. Э. Управление ресурсами и заявками пользователей в Grid-системах с централизованной архитектурой // Труды XII Всероссийского совещания по проблемам управления ВСПУ-2014. Москва, 16—19 июня 2014 г. М.: Институт проблем управления им. В. А. Трапезникова РАН, 2014. С. 7489—7498.
24. Саак А. Э. Алгоритмы диспетчеризации в Grid-системах на основе квадратичной типизации массивов заявок // Информационные технологии. 2011. № 11. С. 9—13.
25. Саак А. Э. Сравнительный анализ полиномиальных алгоритмов диспетчеризации в Grid-системах // Информационные технологии. 2012. № 9. С. 28—32.
26. Саак А. Э. Уровневые алгоритмы диспетчеризации массивами заявок кругового типа в Grid-системах // Известия ЮФУ. Технические науки. 2015. № 6 (167). С. 223—231.
27. Саак А. Э. Кольцевые алгоритмы диспетчеризации массивами заявок в Grid-системах // Информационные технологии. 2016. Т. 22. № 3. С. 163—169.
28. Саак А. Э. Диспетчеризация массивов заявок кругового и гиперболического типа в Grid-системах // Информационные технологии. 2016. Т. 22. № 5. С. 323—332.
29. Саак А. Э. Диспетчеризация заявок кругового типа в Grid-системах // Информационные технологии. 2016. Т. 22. № 1. С. 37—41.
30. Саак А. Э., Курейчик В. В. О качестве диспетчеризации линейных полиэдров точных форм // Известия ЮФУ. Технические науки. 2015. № 4 (165). С. 56—67.
31. Саак А. Э. О качестве диспетчеризации массивов точных форм с заявками параболического типа в Grid-системах // Информационные технологии. 2016. Т. 22. № 8. С. 610—619.
32. Friedman E. (1999). Problem of the Month (August 1999). Pack n integer-sided rectangles in the smallest possible square so that no sides of rectangles are the same. URL: <http://www2.stetson.edu/~efriedma/mathmagic/0899.html>
33. Beldiceanu N., Carlsson M., Rampon J. Global Constraint Catalog, 2nd Ed. (revision a). Working version of SICS Technical Report T2012:03. 2016. February 20. 4142 p. URL: <http://sofdem.github.io/gccat/>
34. Korf R. Huang E. Optimal Rectangle Packing on Non-Square Benchmarks // In Proceedings of the twenty-fourth AAAI Conference on Artificial Intelligence (AAAI-10). Atlanta, Georgia, USA, July 11—15, 2010. P. 83—88.
35. Huang E., Korf R. Optimal rectangle packing: an absolute placement approach // Journal of Artificial Intelligence Research. 2013. Vol. 46. P. 47—87.

A. E. Saak, D.Sc., Associate Professor, Department of State and Municipal Administration,
Head of Department, e-mail: saak@tgn.sfedu.ru;
Southern Federal University, Rostov-on-Don — Taganrog, Russian Federation

Sheduling of sets of Parabolic Tasks in Grid Systems

The effectiveness of Grid systems functioning is mainly based upon the scheduling policy realizing by the resource management and job scheduling system. Grid systems with centralized architecture, which consist of number of sites containing parallel systems with multisite scheduling that is characterized by the possibility of multiprocessor task performance on several sites simultaneously, are basically modeled by resource quadrant. Incoming user's task is characterized by two parameters: by number of the required processors and the action's execution time. Incoming user's task, being actually monitoring by the dispatcher, is modeling by the resource rectangle with the horizontal and vertical dimension, which are correspondingly equaled to the number of resource units of time and quantity of processors required for task performance.

Optimal redistribution of the calculation and timing recourses basically has an exponential labor intensity, as well as an heuristics algorithms of polynomial labor intensity are the things of practical importance. The author suggests resource rectangles environment as methodology of polynomial completeness in scheduling theory. The heuristic polynomial algorithms for processor and time resource distribution, based upon the introduced operations on resource rectangles, are suggested in resource rectangles environment.

Scheduling quality is estimated by the author as heuristic measure which reaches its minimum at in-square packing without emptiness. The problem posed in the paper concerns polynomial algorithms adaptation for precise form sets of parabolic type tasks.

Leveled polynomial algorithms of scheduling are recommended to be put in use in dispatching of the parabolic type access in Grid systems of the precise form and structure.

Keywords: Grid systems with centralized architecture, multisite mode of task processing, recourse rectangle, an area of recourse rectangles, quadrant type task, parabolic type task, non-Euclidean heuristic measure, polynomial completeness of an algorithm, polynomial algorithms for processor and time resource distribution, leveled algorithms in height, leveled algorithms in length

References

1. **Pugliese A., Talia D., Yahyapour R.** Modeling and Supporting Grid Scheduling, *J. Grid Computing*, 2008, vol. 6, is. 2, pp. 195–213.
2. **Li M., Baker M.** *The grid: core technologies*, Chichester, John Wiley & Sons Ltd, 2005, 452 p.
3. **Qureshi M., Dehnavi M., Min-Allah N., Qureshi M., Hussain H., Rentifis I., Tziritas N., Loukopoulos T., Khan S., Xu C., Zomaya A.** Survey on Grid resource allocation mechanisms, *J. Grid Computing*, 2014, vol. 12, is. 2, pp. 399–441.
4. **Schopf J.** Ten actions when Grid scheduling, In J. Nabrzyski et al. (eds.), *Grid resource management-state of the art and future trends*, Springer US, Kluwer Academic Publishers, 2004, 574 p., pp. 15–23.
5. **Pop F.** Grid Scheduling: Methods, Algorithms, and Optimization Techniques, In Preve N. (ed.), *Computational and data Grids: principles, applications and design*, Hershey, IGI Global, 2012, 401 p., pp. 86–111.
6. **Xhafa F., Abraham A.** Computational models and heuristic methods for Grid scheduling problems, *Future Generation Computer Systems*, 2010, vol. 26, is. 4, pp. 608–621.
7. **Siddiqui M., Fahringer T.** Grid resource management: on-demand provisioning, advance reservation, and capacity planning of Grid resources, *LNCS*, vol. 5951, Heidelberg, Springer-Verlag, Berlin, 2010, 227 p.
8. **Feitelson D., Rudolph L., Schwiegelshohn U., Sevcik K., Wong P.** Theory and practice in parallel job scheduling, *LNCS*, 1997, vol. 1291, pp. 1–34.
9. **Tchernykh A., Schwiegelshohn U., Yahyapour R., Kuzjurin N.** Online hierarchical job scheduling on Grids, In Priol T., Vanneschi M. (eds.), *From Grids to service and pervasive computing*, Springer, Science + Business Media, 2008, 241 p., pp. 77–91.
10. **Pascual F., Rzadca K., Trystram D.** Cooperation in multi-organization scheduling, *Concurrency Computat.: Pract. Exper.*, 2009, vol. 21, is. 7, pp. 905–921.
11. **Dutot P., Pascual F., Rzadca K., Trystram D.** Approximation algorithms for the multiorganization scheduling problem, *IEEE Trans. on Paral. and Distrib. Syst.*, 2011, vol. 22, no. 11, pp. 1888–1895.
12. **Bougeret M., Dutot P., Jansen K., Otte C., Trystram D.** A Fast 5/2-approximation algorithm for hierarchical scheduling, *LNCS*, 2010, vol. 6271, pp. 157–167.
13. **Bougeret M., Dutot P., Trystram D., Jansen K., Robenek C.** Improved approximation algorithms for scheduling parallel jobs on identical clusters, *Theoretical Computer Science*, 2015, vol. 600, pp. 70–85.
14. **Bładek I., Drozdowski M., Guinand F., Schepler X.** On contiguous and non-contiguous parallel task scheduling, *J. Sched.*, 2015, vol. 18, is. 5, pp. 487–495.
15. **Caramia M., Giordani S., Iovanela A.** Grid scheduling by online rectangle packing, *Networks*, 2004, vol. 44, no. 2, pp. 106–119.
16. **Błazewicz J., Drabowski M., Węglarz J.** Scheduling multiprocessor tasks to minimize schedule length, *IEEE Trans. on Comput.*, 1986, vol. c-35, no. 5, pp. 389–393.
17. **Turek J., Wolf J., Yu P.** Approximate algorithms for scheduling parallelizable tasks, In *Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'92)*. San Diego, CA, 1992, pp. 323–332.
18. **Günther E., König F., Megow N.** Scheduling and packing malleable and parallel tasks with precedence constraints of bounded width, *J. Comb. Optim.*, 2014, vol. 27, is. 1, pp. 164–181.
19. **Rahman M., Ranjan R., Buyya R., Benatallah B.** A taxonomy and survey on autonomic management of applications in grid computing environments, *Concurrency Computat.: Pract. Exper.*, 2011, vol. 23, is. 16, pp. 1990–2019.
20. **Hamscher V., Schwiegelshohn U., Streit A., Yahyapour R.** Evaluation of job-scheduling strategies for grid computing, *LNCS*, 2000, vol. 1971, pp. 191–202.
21. **Sonmez O., Mohamed H., Epema D.** On the benefit of processor coallocation in multicluster grid systems, *IEEE Trans. on Paral. and Distrib. Syst.*, 2010, vol. 21, no. 6, pp. 778–789.
22. **Saak A. Eh.** Polinomialnye algoritmy raspredeleniya resursov v Grid-sistemakh na osnove kvadrachnoy tipizatsii massivov zayavok [Polynomial algorithms for resource allocation in Grid-based systems for quadratic typing, arrays applications], *Informacionnyye tekhnologii*, 2013, no. 7, *Prilozhenie*, 32 p. (in Russian).
23. **Saak A. Eh.** Upravleniye resursami i zayavkami polzovatelye v Grid-sistemakh s centralizovannoy arkhitektyroy [Resource and multiprocessor task management in Grid system of centralized architecture], *Trudy XII Vserossiyskogo soveshanya po problem upravleniya VSPU-2014. Moskva, 16 ijunya — 19 ijunya 2014 g.* [Proceedings of XII all-Russian conference "Control problems" RCCP'2014. Moscow, 16 June — 19 June 2014]. Moscow: Institut problem upravleniya im. V. A. Trapeznikova RAN, 2014, pp. 7489–7498 (in Russian).
24. **Saak A. Eh.** Algoritmy dispatcherizatsii v Grid-sistemakh na osnove kvadrachnoy tipizatsii massivov zayavok [Algorithms scheduling in Grid-based systems for quadratic typing, arrays applications], *Informatsionnyye tekhnologii [Information Technologies]*, 2011, no. 11, pp. 9–13.
25. **Saak A. Eh.** Sravnitelniy analiz polinomialnykh algoritmov dispatcherizatsii v GRID-sistemakh [Comparative analysis of polynomial algorithms for scheduling in GRID-systems], *Informatsionnyye tekhnologii*, 2012, no. 9, pp. 28–32 (in Russian).
26. **Saak A. Eh.** Urovnevy algoritmy dispatcherizatsii massivamy zayavok krygovogo tipa v Grid-sistemakh [Level algorithms of scheduling by circle type task sets in grid systems], *Izvestiya YuFU. Tekhnicheskie nauki*, 2015, no. 6 (167), pp. 223–231 (in Russian).
27. **Saak A. Eh.** Kolcivie algoritmy dispatcherizatsii massivamy zayavok v Grid-sistemakh [Ring algorithms for scheduling in grid systems by sets of tasks], *Informatsionnyye tekhnologii*, 2016, vol. 22, no. 3, pp. 163–169 (in Russian).
28. **Saak A. Eh.** Dispatcherizatsiya massivov zayavok krygovogo i giperbolicheskogo tipa v Grid-sistemakh [Scheduling of sets of circular-type and hyperbolic-type tasks in grid-systems], *Informatsionnyye tekhnologii*, 2016, vol. 22, no. 5, pp. 323–332 (in Russian).
29. **Saak A. Eh.** Dispatcherizatsiya zayavok krygovogo tipa v Grid-sistemakh [Circular-typed multiprocessor tasks scheduling in grid systems], *Informatsionnyye tekhnologii*, 2016, vol. 22, no. 1, pp. 37–41 (in Russian).
30. **Saak A. Eh., Kureichik V. V.** O kachestve dispatcherizatsii lineynix poliedralei tochnix form [To quality of precisely formed linear polyedra scheduling], *Izvestiya YuFU. Tekhnicheskie nauki*, 2015, No. 4 (165), pp. 56–67 (in Russian).
31. **Saak A. Eh.** O kachestve dispatcherizatsii massivov tochnix form s zayavkami parabolicheskogo tipa v Grid-sistemakh [To quality of scheduling of precise form sets of parabolic type tasks in grid systems], *Informatsionnyye tekhnologii*, 2016, vol. 22, no. 8, pp. 610–619 (in Russian).
32. **Friedman E.** *Problem of the Month (August 1999)*. Pack n integer-sided rectangles in the smallest possible square so that no sides of rectangles are the same, 1999. URL: <http://www2.stetson.edu/~efriedma/mathmagic/0899.html>
33. **Beldiceanu N., Carlsson M., Rampon J.** *Global Constraint Catalog, 2nd Ed. (revision a)*, Working version of SICSts Technical Report T2012:03, February 20, 2016, 4142 p. URL: <http://sofdem.github.io/gccat/>
34. **Korf R., Huang E.** Optimal Rectangle Packing on Non-Square Benchmarks, In *Proceedings of the twenty-fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, Atlanta, Georgia, USA, July 11–15, 2010, pp. 83–88.
35. **Huang E., Korf R.** Optimal rectangle packing: an absolute placement approach, *Journal of Artificial Intelligence Research*, 2013, vol. 46, pp. 47–87.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОРГАНИЗАЦИОННЫХ И СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ INFORMATION TECHNOLOGIES IN THE ORGANIZATIONAL AND SOCIO-ECONOMIC SYSTEMS

УДК 004.942

В. И. Сумин, д-р техн. наук, проф., e-mail: viktorsumin51@yandex.ru,
ФКОУ ВПО Воронежский институт ФСИН России,

Т. Е. Смоленцева, доц., канд. техн. наук, e-mail: smoltan@bk.ru,
ФГБОУ ВПО "Липецкий государственный технический университет"

Формирование структуры иерархических многоуровневых организационных систем

Описывается процесс построения структуры организаций и их систем управления, которые являются многоуровневыми по своей природе. Системы управления таких организаций рассматриваются как сложные иерархические многоуровневые организационные системы. Приведен подход к формированию вновь создаваемой структуры организации в виде иерархической многоуровневой организационной системы, осуществляемой на основе анкетирования, экспертного и кластерного анализов. С помощью кластерного анализа определяются структурные иерархические элементы такой организации. В дальнейшем структура организации представляется в виде ориентированного графа без контуров. Структура графа строится в соответствии с иерархией целей организации.

Ключевые слова: иерархическая многоуровневая организационная система, система управления, анкетирование, экспертный анализ, кластерный анализ, ориентированный граф, итеративный метод

Введение

Одной из главных задач разработки эффективной системы управления (СУ), которую будем представлять в виде иерархической многоуровневой организационной системой (ИМОС), является формирование ее структуры в соответствии с поставленными целями и задачами, а также требованиям нормативных документов. Построение структур ИМОС должно осуществляться на основе:

- описания структур;
- формализации функционирования элементов структуры;
- идентификации цели каждого из элементов структуры ИМОС;
- формализации процедур корректировки структур.

При разработке ИМОС решаются следующие задачи:

- разработка метода построения СУ ИМОС;
- разработка алгоритмов формирования ИМОС.

Разработка метода построения СУ ИМОС

В наше время динамически изменяются требования к функционированию ИМОС, поэтому не-

обходимо постоянно корректировать цели и задачи организации и в связи с этим возникает необходимость корректировать также структуру ИМОС.

Разработка новой структуры ИМОС осуществляется на основе анкетирования и экспертного анализа взаимодействия элементов этой структуры. Анкетирование должно осуществляться с определением следующей информации:

- частота управленческих решений, которые принимаются на иерархических уровнях ИМОС;
- модели принятия управленческих решений руководителями для подчиненных;
- степень участия руководителя в разработанной модели принятия управленческих решений для подчиненных.

Такое анкетирование позволит определить вновь создаваемую структуру ИМОС в соответствии с предполагаемыми целями и задачами с использованием кластерного анализа.

Кластерный анализ позволит сгруппировать данные, которые описывают схожие объекты сложных систем в классы. Главной целью кластерного анализа является выявление групп (кластеров) похожих объектов в выборке данных. Определение каждого

кластера происходит на основе идентификации для каждого кластера точки пространства, относительно которой определяются метрические расстояния точек этого кластера. Размерность пространства равна числу характеристик, на базе которых происходит описание объекта [1, 2].

Процесс группировки объектов в ходе решения задачи по выборки данных осуществляется на основе следующих методов:

- метода итеративных группировок;
- метода поиска модальных значений плотности;
- методов факторного анализа;
- метода сгущений;
- теории графов.

Необходимо отметить, что группировка схожих объектов сложной системы на основе применения различных методов может привести к противоречивым результатам. Поэтому необходимо детальное изучение каждого метода с последующим выявлением всех достоинств и недостатков каждого из них.

Использование метода итеративных группировок осуществляется на основе первичных данных с использованием матрицы сходств между объектами. Методы итеративных группировок имеют возможность обрабатывать большое количество информации при нескольких просмотрах данных и, следовательно, компенсируют последствия неэффективного исходного разбиения данных. Эти методы образуют кластеры одного ранга, которые не являются вложенными и поэтому не могут быть элементами иерархии и не допускают наложения кластеров друг на друга [3].

Эти методы могут ограничить число итераций по выявлению принадлежности к одному из классов. Поэтому определяется минимальное количество объектов, которые переходят из класса в класс, и в этом случае итерационный процесс заканчивается с требуемой точностью.

Структура процесса группирования объектов на основе итеративных методов приведена на рисунке.

Рассмотрим использование метода итеративной группировки объектов базовой информации (БИ) для сложных систем на базе анализируемых характеристик объектов [3–7].



Структура процесса группирования объектов на основе метода итеративных группировок

Представим БИ, циркулирующую в сложных системах, в виде множеств $\{P_{i,j}, A_i, B_j\}$. Исходными данными разбиения объектов на классы являются:

$i = \overline{1, I}$ — индекс объекта;

$j = \overline{1, J}$ — индекс характеристики объекта;

$P_{i,j}$ — количественное значение j -й характеристики i -го объекта;

A_i — наименование объекта;

B_j — наименование соответствующей характеристики;

T_I — требуемая точность разбиения в процентах;

K_3 — требуемое количество классов разбиения.

Требуется объекты $P_{i,j}$ разбить с точностью T_I на K_3 классов. Значение T_I определяет число итераций. В том случае, если число итераций увеличивать, то это приведет к уменьшению точности разбиения. Число классов разбиения K_3 определяется лицом, принимающим решение (ЛПР) [4, 8, 9].

Для того чтобы анализируемое разбиение имело требуемое число классов, которое должно быть не меньше K_3 , требуется определить масштабный коэффициент L , которому вначале присваивается значение, равное 1.0.

Разработка алгоритмов формирования ИМОС

Ниже приведен алгоритм разбиения на классы.

0. Для образования первоначального разбиения на требуемое число классов необходимо образовать множество $\{r_i, d_i, a_i, k_i\}$ размерностью $I = 1, I$.
Здесь

r_i — смешанный момент корреляции или угловая мера

$$r_i = \sum_{j=1}^J (P_{i,j} - \bar{P}_{i,j}) / \sqrt{\sum_{j=1}^J (P_{i,j} - \bar{P}_{i,j})^2},$$

где

$$\bar{P}_{i,j} = \frac{1}{J} \sum_{j=1}^J P_{i,j}$$

$$d_i = \sqrt{\sum_{j=1}^J (0 - P_{i,j})^2} — \text{евклидово расстояние};$$

a_i — индекс объекта;

k_i — номер класса, к которому будет принадлежать i -й объект, первоначально все $k_i = 0$.

1. Первоначальное разбиение на требуемое число классов [1, 2].

1.1. Для начала итеративного процесса принимается:

- $C_k = 0, k_1 = 1, i = 1, C_i = 1$;
- вычисляется среднее расстояние s между всеми элементами

$$s = \sum_{i=1}^{I-1} (d_i - d_{i+1})/I.$$

1.2. Определяется пороговое значение α для определения принадлежности i -го объекта к классу C_k :

$$\alpha = s \times L;$$

$$i = C_i$$

1.3. Определяется расстояние между i -м и следующим объектом

$$\Delta d = d_i - d_{(i+1)}.$$

1.4. Анализируется принадлежность $(i + 1)$ -го объекта к классу C_k .

Если $\Delta d \leq \alpha$, то $k_{(i+1)} = C_k, i = i + 1$ и:

- если $i \leq I$, то переход к п. 1.3;
- если $i > I$, то переход к п. 1.5.

Если $\Delta d > \alpha$, то $C_k = C_k + 1, i = i + 1, C_i = i$ и переход к п. 1.2.

1.5. Осуществляется интеграция на основе смешанного момента корреляции.

1.6. Вначале элементы $\{r_i, k_i\}$ переупорядочиваются по возрастанию элементов k_i и r_i и определяются $C_k = 1, C_{kt} = 1, k_1 = 1, i = 1, C_i = 1$.

1.7. Определяется пороговое значение α с фиксацией принадлежности $(i + 1)$ -го объекта классу C_i :

$$\alpha = (r_i - r_{(i+1)}) \times L.$$

Если $\alpha = 0$, то $i = i + 1$ и α вычисляется заново.

Если $|\alpha| > 0$, то $\alpha = |\alpha|$ и $i = C_i$.

1.8. Определяется, закончились ли объекты класса C_{kt}

Если $C_{kt} = k_i$, то переход к п. 1.9.

Если $C_{kt} \neq k_i$, то $C_{kt} = C_{kt} + 1, C_k = C_k + 1, i = i + 1, C_i = I$ и:

- если $i > I$, то переход к п. 1.11;
- если $i \leq I$, то переход к п. 1.7.

1.9. Определяется расстояние между смежными объектами и следующим $\Delta r = r_i - r_{(i+1)}$.

1.10. Определяется принадлежность $(i + 1)$ -го объекта к классу C_k :

Если $\Delta r \leq \alpha$, то $k_{(i+1)} = C_k, i = i + 1$ и:

- если $i \leq I$, то переход к п. 1.9;
- если $i > I$, то переход к п. 1.11.

Если $\Delta d > \alpha$, то $C_k = C_k + 1, i = i + 1, C_i = I$ и переход к п. 1.7.

1.11. Результаты $P_{i,j}$ разбиты на K классов.

Если $K = K_3$, то требуемое разбиение получено и переход к п. 2 [7].

Если $K > K_3$, то увеличивается параметр L и переход к п. 1.1. Если $K < K_3$, то уменьшается параметр L и переход к п. 1.1.

2. Вычисляются $\bar{P}_{k,j}$ — центры тяжести полученных классов:

$$\bar{P}_{k,j} = \frac{\sum_{(1, k_i)} P_{a_i j}}{\sum_{(1, k_i)} I}, k = \overline{1, K_3}, \text{ — индекс}$$

полученных классов.

3. Определяется, находится ли каждый объект в ближайшем классе.

3.1. Первоначально принимаем $i = 1, n = 0$.

3.2. Определяется квадрат отклонения объекта a_i от центра тяжести всех классов:

$$F_{ka_i} = \sum_{j=1}^J (P_{a_i j} - \bar{P}_{k,j})^2,$$

где $k = \overline{1, K_3}$ — индекс полученных классов; $j =$

$\overline{1, J}$ — индекс характеристики, участвовавшей в формировании результата $P_{i,j}$ объекта a_i .

3.3. Если $\min(F_{ka_i})$ достигается при $k = k_i$, то объект a_i находится в ближайшем классе, изменения его класса не происходит.

Если $\min(F_{ka_i})$ достигается при $k \neq k_i$, то объект a_i не находится в ближайшем классе, поэтому $k_i = k$ (класс объекта заменился на ближайший) и $n = n + 1$ (объект a_i перешел в другой класс)[2].

3.4. Увеличивается $i = i + 1$ и проверяется:

- если $i > I$, то закончился просмотр всех объектов и переход к п. 4;
- если $i \leq I$, то переход к п. 3.2.

4. Если $\frac{n}{I} \cdot 100 > T_p$, то требуемая точность итеративного процесса не достигнута и переход к п. 2.

Если $\frac{n}{I} \cdot 100 \leq T_p$, то требуемая точность итеративного процесса достигнута. Получено окончательное разбиение $P_{i,j}$ по классам.

Результатом разбиения объектов на классы являются:

K — число полученных классов;

$\bar{P}_{k,j}$ — центры тяжести полученных классов;

k_i — номер класса, к которому принадлежит объект;

a_i — индекс объекта.

После анализа сложных систем структуру ИМОС можно представлять в виде графа $G(V, E)$, где V — множество структурных элементов ИМОС, E — множество управляющих связей.

Если граф G не представляется в иерархическом виде, то его цели необходимо привести к иерархическому виду.

Рассмотрим, как формировать связи между множеством структурных элементов ИМОС (V) в графе G .

Будем считать, что $E(v)$ — множество всех дуг вида (v', v) , где $v' \in V$.

Определим коэффициенты инцидентности $U(e)$ так, чтобы $e \in E$ для любого $v \in V$:

$$\sum_{e \in E(v)} U(e) = 1. \quad (1)$$

Тогда в вершину v входит только дуга e и $U(e) = 1$.

Широко распространенный подход приведения графа G к иерархической структуре можно осуществлять так, чтобы из множества $E(v)$ существовала бы одна дуга, для которой справедливо выражение

$$U(e_0) = \max_{e \in E(v)} U(e). \quad (2)$$

Такой путь приведения графа G к иерархической структуре не может полностью отражать функционирование ИМОС.

Другой путь приведения графа G к иерархической структуре можно осуществить за счет укрупненного графа G вершины, который формируется объединением определенной совокупности структурных элементов из множества V в один элемент. В этом случае граф G будет ориентированным графом без контуров. Для реализации такого подхода разобьем множество V на подмножества V_1, V_2, \dots, V_n и на этом подмножестве будем осуществлять следующие действия.

Выделим множество вершин, для которых нет входящих дуг $e \in E$ и множество всех вершин $v \in V_k = V \setminus V_1 \dots V_k$ таких, что им идентичны только дуги $(v', v) \in E$, где $v' \in V_1 \cup V_2 \cup \dots \cup V_k$ для $k = 1, 2, \dots, n$. Будем считать, что n — максимальное значение индекса k при v_k .

Разбиение на попарно непересекающиеся подмножества V_1, V_2, \dots, V_n множества вершин графа G является многоуровневой иерархической структурой. Для множеств V_1, V_2, \dots, V_n не существует дуг $e(v_1, v_2)$ таких, что $v_1 \in V_k, v_2 \in V_j$ и $k > j$. Определение подхода приведения графа G к иерархической структуре осуществляется в соответствии со спецификой ИМОС.

Очень часто используется подход приведения графа G к иерархической структуре с учетом связи его структуры с иерархией целей ИМОС. Вначале формируется иерархическая структура ИМОС и начинают ее идентифицировать с глобальной целью ИМОС. Обозначим глобальную цель ИМОС через C_1 .

Подцель λ -го уровня дерева целей будем определять через C_α где $\alpha = (1, i_1, i_2, \dots, i_{\lambda-1})$ — число, которое эквивалентно значению λ -го уровня дере-

ва целей. В дальнейшем будем считать, что ρ^α есть состояние системы S_α , где $|\alpha| = L, |\alpha| \in \{1, \dots, L\}$, а u_α — управляющее воздействие этой системы на элементы ИМОС нижних уровней.

Интегрированная управляющая информация E из системы S_α , формируемая на основе цели C_α , передается на более низкий уровень ИМОС в соответствии с целями $C_{\alpha, i}, i = 1, \dots, \gamma^\alpha$, этих уровней, на которых решаются задачи из семейства $\{Z^{\alpha_j}\}$, $j = 1, \dots, \mu_\alpha$, и, в свою очередь, в соответствии с этими целями $E = (e^{\alpha, 1}, \dots, e^{\alpha, \gamma^\alpha})$ разбивается на более мелкие информационные объекты.

Заключение

В статье приведен алгоритм построения структуры организаций и их систем управления, которые представляются в виде иерархической многоуровневой организационной системы. Вновь создаваемая структура такой организации формируется с использованием анкетирования, экспертного и кластерного анализов. Использование кластерного анализа позволяет определить структурные иерархические элементы этой организации. После определения иерархических структурных элементов организации эта структура представляется в виде ориентированного графа без контуров. Структура графа формируется в соответствии с иерархией целей организации. Такой подход к формированию структуры организации позволяет связать ее структуру с ее целями и задачами.

Список литературы

1. Мендель И. Д. Кластерный анализ. М.: Финансы и статистика, 1988. 176 с.
2. Сумин В. И., Смоленцева Т. Е., Дыбова М. А. Методика группирования базовой информации для информационных процессов сложных систем // Фундаментальные исследования. (Часть 5). 2015. № 2. С. 931–934.
3. Вагин В. Н., Головина Е. Ю. Достоверный и правдоподобный вывод в интеллектуальных системах. М.: Физматлит, 2004. 764 с.
4. Саати Т. В. Аналитическое планирование: организация систем. М.: Радио и связь, 1998. 224 с.
5. Laudon K., Laudon J. Management Information Systems: New Approaches to Organization and Technology, Prentice Hall, 1998. 675 p.
6. Журавлев Ю. И., Рязанов В. В., Сенько О. В. Распознавание. Математические методы. Программная система. Практические применения. М.: Фазис, 2006. 159 с.
7. Замятина О. М. Моделирование систем. Томск: Изд. Томского политехнического университета, 2009. 204 с.
8. Сумин В. И., Цветков В. В. Об алгоритмах и моделях, данных в решениях задач принятия решения // Научные ведомости Белгород. гос. ун-та. Сер. История. Политология. Экономика. Информатика. 2010. № 13 (84). Вып. 15/1. С. 120–128.
9. Сумин В. И., Смоленцева Т. Е. Моделирование обучения с использованием временных рядов наблюдений. М.: Научная книга, 2014. 104 с.

V. I. Sumin, Professor, e-mail: viktorsumin51@yandex.ru,
Voronezh institute of the Russian Federal Penitentiary Service,
T. E. Smolentseva, Associate Professor, e-mail: smoltan@bk.ru,
Lipetsk state technical University

The Formation of the Structure of Hierarchical Multi-Level Organizational Systems

The article describes the process of building the structure of organizations and their management systems, which are multilevel in nature. The control system of such organizations are seen as complex hierarchical multi-level organizational system. In the article the approach to formation of the newly created organization structure in a hierarchical multi-level organizational system implemented on the basis of questionnaires, expert and cluster analysis. With the help of the cluster analysis are determined by structural hierarchical elements of such an organization. In the future, the structure of the organization is represented as a directed graph without contours. The structure of the graph is constructed in accordance with the hierarchy of objectives of the organization.

Keywords: hierarchical multi-level organizational system, management system, questionnaire survey, expert analysis, cluster analysis, directed graph, an iterative method

References

1. Mendel' I. D. *Klasternyj analiz*, Moscow, Finansy i statistika, 1988. 176 p. (in Russian).
2. Sumin V. I., Smolenceva T. E., Dybova M. A. Metodika grupirovaniya bazovoj informacii dlja informacionnyh processov slozhnyh sistem, *Fundamental'nye issledovanija* (Chast' 5), 2015, no. 2, pp. 931–934. (in Russian).
3. Vagin V. N., Golovina E. Ju. *Dostovernij i pravdopodobnyj vyvod v intellektual'nyh sistemah*, Moscow, Fizmatlit, 2004. 764 p. (in Russian).
4. Saati T. V. *Analiticheskoe planirovanie: organizacija sistem*, Moscow, Radio i svjaz', 1998. 224 p. (in Russian).
5. Laudon K., Laudon J. *Management Information Systems: New Approaches to Organization and Technology*, Prentice Hall, 1998. 675 p.
6. Zhuravlev Ju.I., Rjazanov V. V., Sen'ko O. V. *Raspoznvanie. Matematicheskie metody. Programmnaja sistema. Prakticheskie primenenija*, Moscow, Fazis, 2006, 159 p. (in Russian).
7. Zamjatina O. M. *Modelirovanie sistem*, Tula, Izdatel'stvo TPU, 2009. 204 p. (in Russian).
8. Sumin V. I., Cvetkov V. V. Ob algoritmah i modeljah, dannyh v reshenijah zadach prinjatija reshenija, *Nauchnye vedomosti Belgorod. gos. un-ta. Ser. Istorija. Politologija. Ekonomika. Informatika*, 2010, no. 13 (84), is. 15/1, pp. 120–128 (in Russian).
9. Sumin V. I., Smolenceva T. E. *Modelirovanie obuchenija s ispol'zovaniem vremennyh rjadov nabljudenij*, Moscow, Nauchnaja kniga, 2014, 104 p. (in Russian).

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор Е. В. Конова.

Корректор Н. В. Яшина.

Сдано в набор 10.04.2017. Подписано в печать 26.05.2017. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT617. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.