ВЕСТНИК
НОВОСИБИРСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

Научный журнал
Основан в ноябре 1999 года

Серия: Информационные технологии 2018. Том 16, № 2

СОДЕРЖАНИЕ

VESTNIK
NOVOSIBIRSK STATE UNIVERSITY

Scientific Journal
Since 1999, November
In Russian

Series: Information Technologies 2018. Volume 16, № 2

CONTENTS

The journal is published quarterly in Russian since 1999
by Novosibirsk State University Press

The address for correspondence

Centre of Information Technologies, Novosibirsk State University

Pirogov Street 2, Novosibirsk, 630090, Russia

Tel. +7 (383) 363 40 28

E-mail address: inftech@vestnik.nsu.ru

On-line version: http://elibrary.ru

УДК 004.048:519.765
DOI 10.25205/1818-7900-2018-16-2-5-18

Т. В. Батура [1, 2], **С. Е. Стрекалова** [1]

[1] *Новосибирский государственный университет*
ул. Пирогова, 1, Новосибирск, 630090, Россия

[2] *Институт систем информатики им. А. П. Ершова СО РАН*
пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия

tatiana.v.batura@gmail.com, svetlana.strekalova@gmail.com

ПОДХОД К ПОСТРОЕНИЮ РАСШИРЕННЫХ ТЕМАТИЧЕСКИХ МОДЕЛЕЙ ТЕКСТОВ НА РУССКОМ ЯЗЫКЕ

Представлен новый подход для получения расширенных тематических моделей текстов научных статей на русском языке. Под расширенной моделью понимается тематическая модель, содержащая кроме однословных терминов термины, состоящие из нескольких слов (также называемые многословные термины или ключевые фразы). Такие модели лучше интерпретируются пользователями и точнее описывают предметную область документа, чем модели, состоящие только из униграмм (отдельных слов).

На основе предложенного подхода была разработана система, в результате работы которой для каждого документа предоставляется набор содержащихся в нем тем с указанными вероятностями, ключевыми словами и фразами для каждой темы.

Предложенный в статье подход может быть полезен при построении рекомендательных систем и систем автореферирования.

Ключевые слова: тематические модели, обработка текста, извлечение ключевых слов, извлечение многословных терминов, определение темы текста.

Введение

В современном мире непрерывно производятся огромные объемы электронной информации. Значительную ее часть составляют тексты на естественном языке. В связи с этим становится все более актуальной задача автоматической обработки таких текстов с целью извлечения из них структурированных данных, пригодных для дальнейшего использования в машинном анализе.

Одним из современных инструментов обработки естественного языка являются тематические модели. Тематическое моделирование заключается в построении модели некоторой коллекции текстовых документов. В такой модели каждая тема представляется дискретным распределением вероятностей слов, а документы – дискретным распределением вероятностей тем [1].

Следует обратить внимание на то, что нельзя смешивать понятия тематического моделирования и тематической классификации. Основное отличие состоит в том, что при определении тем текстов отсутствует какая-либо информация о темах: неизвестно ни их количество, ни их содержание (что подразумевается под каждой темой). Для классификации же необходимы априорные знания о структуре классов. В этом смысле процесс тематического моделирования больше похож на кластеризацию, чем на классификацию. Однако ни классификация, ни кластеризация не справляются с синонимией и полисемией, в отличие от тематического

Батура Т. В., *Стрекалова С. Е.* Подход к построению расширенных тематических моделей текстов на русском языке // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 5–18.

моделирования. А, как известно, важнейшим препятствием при создании систем автоматической обработки текстов является лексическая неоднозначность. Так, в тематической модели слова, являющиеся синонимами, с большой вероятностью попадут в одну и ту же тему, так как зачастую они используются в одинаковом контексте. В то же время омонимы (слова одинаковые по написанию, но имеющие разное значение) с большой вероятностью будут отнесены к разным темам, так как обычно контексты их использования не совпадают.

В данной статье описан новый подход для получения расширенных тематических моделей текстов научных статей на русском языке. Под расширенной моделью здесь понимается тематическая модель, содержащая помимо однословных терминов термины, состоящие из нескольких слов (также называемые многословными терминами или ключевыми фразами). Такие модели лучше интерпретируются пользователем и точнее описывают предметную область документа, чем модели, состоящие только из униграмм (отдельных слов).

Основные понятия и постановка задачи построения тематической модели

Тематическое моделирование – построение тематической модели некоторой коллекции текстовых документов. Тематическая модель представляет собой описание коллекции с помощью тематик, использующихся в документах этой коллекции, и определяет слова, относящиеся к каждой из тематик [1].

Вероятностная тематическая модель представляет каждую тему как дискретное распределение на множестве слов, а документ – как дискретное распределение на множестве тем [2].

Одной из разновидностей тематических моделей являются тематические модели, выявляющие ключевые фразы (термины) предметной области. Под ключевой фразой в данной работе подразумевается устойчивая последовательность слов (*n*-грамма), имеющая определенную семантику в контексте заданной предметной области, относящаяся к одной из выявленных в тексте тем и обладающая значительной частотой встречаемости по сравнению с другими *n*-граммами.

Задача построения тематической модели

Пусть задана некоторая коллекция документов D, тогда W – множество всех встречающихся в данной коллекции терминов (слов или *n*-грамм). Каждый документ $d \in D$ представляется в виде последовательности терминов $\left(w_1,...,w_{n_d}\right)$ длиной n_d, $w \in W$, при этом каждый термин может встретиться в документе несколько раз.

Предполагается, что существует некоторое множество тем T, причем каждое вхождение термина w связано с некоторой темой t. Коллекция документов рассматривается как множество троек (d, w, t), выбранных случайно и независимо из дискретного распределения $p(d, w, t)$, заданного на конечном множестве $D \times W \times T$. При этом документы $d \in D$ и термины $w \in W$ являются наблюдаемыми переменными, а тема $t \in T$ – скрытой переменной.

Гипотеза о том, что элементы выборки независимы, эквивалентна предположению «мешка слов»: порядок слов в тексте документа не имеет значения, и тематику можно выявить даже при произвольной перестановке терминов в тексте. В этом случае каждый документ можно представить как подмножество $d \subseteq W$, в котором в соответствие с каждым элементом w_d поставлено количество вхождений n_{d_w} термина w в документ d.

Согласно определению условной вероятности, формуле полной вероятности и гипотезе условной независимости

$$p\left(w|d\right) = \sum_{t \in T} p\left(t|d\right) \cdot p\left(w|t\right).$$

Тогда задача построения тематической коллекции документов заключается в нахождении для известной коллекции D множества всех использующихся в ней тем T, а также для каждого

$d \in D$ по распределению слов по документам $p(w|d)$ восстановить распределения тем в документе $p(t|d)$ и слов по темам $p(w|t)$.

Обзор существующих решений

В настоящее время тематические модели находят применение в самых различных областях. К примеру, в [3] авторы используют тематическое моделирование с помощью алгоритма Latent Dirichlet Allocation (LDA) на отзывах пользователей для создания персонализированных медицинских рекомендаций. В работе [4] авторы используют тематическую модель, включающую в себя авторов, тексты и цитирования, для библиографического анализа. Также тематическое моделирование применяется в обучении: в работе [5] авторы предлагают использовать тематическое моделирование для упрощения оценки учителем письменных работ учеников. Помимо этого, тематическое моделирование применяется для анализа данных социальных сетей [6–8], для многоязычного информационного поиска [9], выявления трендов в новостных потоках или научных публикациях [10], для автоматического присвоения тегов веб-страницам [11], в рекомендательных системах, учитывающих контекст [12], в анализе террористической активности в сети Интернет [13] и мн. др.

Современные требования к тематическим моделям довольно разнообразны. Основное из них заключается в том, что тематические модели должны хорошо поддаваться интерпретации, конечному пользователю должны быть понятны причины выделения определенных тем в тексте и структура самих тем. Эта особенность является главным преимуществом тематических моделей перед набирающими популярность нейронными сетями. Кроме того, часто требуется, чтобы тематические модели учитывали разнородные данные, выявляли динамику тем во времени, автоматически разделяли темы на подтемы, использовали не только отдельные ключевые слова, но и многословные термины и т. д.

Основными подходами к тематическому моделированию являются алгоритмы PLSA (Probabilistic Latent Semantic Analysis, вероятностный латентный семантический анализ), LDA (Latent Dirichlet Allocation, латентное размещение Дирихле) и библиотека ARTM (Additive Regularization for Topic Modeling, аддитивная регуляризация тематических моделей).

PLSA – вероятностная тематическая модель представления текста на естественном языке. Модель называется латентной, так как предполагает введение скрытого (латентного) параметра, являющегося темой. Впервые описана Томасом Хофманном в 1999 г. [14].

LDA – модель, позволяющая объяснять результаты наблюдений с помощью неявных групп, благодаря чему возможно выявление причин сходства некоторых частей данных. Например, если наблюдениями являются слова, собранные в документы, утверждается, что каждый документ представляет собой смесь небольшого количества тем и появление каждого слова связано с одной из тем документа [15].

ARTM является обобщением большого числа алгоритмов тематического моделирования, позволяет комбинировать регуляризаторы, тем самым комбинируя тематические модели. При таком подходе PLSA представляет собой тематическую модель без регуляризаторов, а LDA – тематическую модель, в которой каждая тема сглажена одним и тем же регуляризатором Дирихле. Модель ARTM в предложена 2014 г. [16]. В настоящее время ARTM приобретает все большую популярность благодаря своей универсальности и гибкости настройки параметров моделей.

Многословные термины

Проблема извлечения многословных терминов

Как уже говорилось, основным требованием к тематическим моделям является их интерпретируемость. При этом в большинстве алгоритмов тематического моделирования в качестве терминов используются только слова, а не *n*-граммы. В то же время для человека использование ключевых фраз для обозначения тем может упростить интерпретацию выявленной

темы и разрешить возможную неоднозначность. При этом стоит заметить, что в русском языке задача извлечения ключевых фраз является гораздо более сложной, чем, например, в английском. Это связано с тем, что русский язык флективный, т. е. каждое слово в речи может быть представлено множеством различных словоформ. Обычные алгоритмы извлечения ключевых фраз, основанные на относительной частоте встречаемости *n*-грамм в документах, показывают низкий уровень точности извлечения. Каждую словоформу такие алгоритмы воспринимают как различные термины, и из-за этого частота встречаемости снижается в несколько раз.

Существует несколько основных подходов к решению данной проблемы. Во-первых, для распознавания словоформ можно использовать словари, содержащие все возможные формы слова [17]. Очевидно, что в этом случае точность определения будет высокой для имеющихся в словаре слов. Однако очевидно, что применимость словарных алгоритмов ограничена предметной областью словаря.

Другой подход к этой задаче – использование лексико-синтаксических шаблонов [18; 19]. В [18] описана стратегия распознавания в заданном тексте фрагментов, соответствующих заданному лексико-синтаксическому шаблону, предложен язык записи шаблонов, позволяющий задавать лексические и грамматические свойства входящих в него элементов. В статье [19] приводится описание системы с возможностью ручной настройки видов шаблонов для извлечения словосочетаний с помощью набора морфологических признаков. К сожалению, основными недостатками методов, основанных на шаблонах, является их большая трудоемкость.

Проблему многословных терминов можно обойти, если использовать стемминг (нахождение основы слова) или лемматизацию (приведение слова к его начальной форме). Однако тогда возникает проблема с восстановлением изначальных словосочетаний: так, биграмма будет после стемминга выглядеть как «тематическ моделировании», а после лемматизации – как «тематический моделирование». Очевидно, такие биграммы не могут быть использованы в качестве ключевых фраз в научной статье или на веб-странице, и для дальнейшего использования нужно преобразовать их в изначальное словосочетание.

Предложенное решение проблемы многословных терминов

Для решения проблемы согласования словосочетаний применялись лексико-синтаксические шаблоны. Исследование многословных ключевых терминов, выбранных для статей авторами, позволило составить базовый набор шаблонов. Мы не можем утверждать, что этот набор является полным, так как для составления полного набора шаблонов понадобилось бы привлечь экспертов-лингвистов с целью проведения дополнительного исследования. По этой причине вопрос о полноте набора шаблонов терминов пока остается открытым. Однако предусмотрено возможное расширение набора шаблонов, и в случае увеличения их количества потребуются лишь минимальные изменения в модуле согласования словосочетаний. Выделенные шаблоны удобно записать при помощи логики предикатов первого порядка.

Рассмотрим словарь V – множество слов коллекции документов. Пусть $x, x_1, x_2, ..., x_i, ..., x_n$ – множество прилагательных из V; $y, y_1, y_2, ..., y_i, ..., y_m$ – множество существительных из V. Для морфологических признаков введем следующие обозначения: $z_1 = \{mal, fem, neu\}$ содержит информацию о категории рода (мужской, женский, средний); $z_2 = \{sin, plu\}$ – о категории числа (единственное, множественное); $z_3 = \{nom, gen, dat, acc, ins, pre\}$ – о категории падежа (именительный, родительный, дательный, винительный, творительный, предложный). Далее введем четырехместные предикаты $A(x, z_1, z_2, z_3)$ для прилагательных и $N(y, z_1, z_2, z_3)$ для существительных. Теперь шаблоны многословных терминов можно записать в виде формул исчисления предикатов, т. е. в случае согласованных словосочетаний будут истинны следующие шаблоны.

1. $MWE_1(x, y): A(x, z_1, z_2, nom) \wedge N(y, z_1, z_2, nom)$.

Например, «линейное уравнение».

2. $MWE_2(y_1, y_2): N(y_1, z_1^1, z_2^1, nom) \wedge N(y_2, z_1^2, z_2^2, gen)$.

Например, «разработка системы».

3. $MWE_3(y_1, x, y_2): N(y_1, z_1^1, z_2^1, nom) \wedge A(x, z_1^2, z_2^2, gen) \wedge N(y_2, z_1^2, z_2^2, gen)$.

Например, «гипотеза условной независимости».

4. $MWE_4(x_1, x_2, y): A(x_1, z_1, z_2, nom) \wedge A(x_2, z_1, z_2, nom) \wedge N(y, z_1, z_2, nom)$.

Например, «вероятностная тематическая модель».

5. $MWE_5(y_1, y_2, y_3): N(y_1, z_1^1, z_2^1, nom) \wedge N(y_2, z_1^2, z_2^2, gen) \wedge N(y_3, z_1^3, z_2^3, gen)$.

Например, «определение тематики документа».

6. $MWE_6(x, y_1, y_2): A(x, z_1^1, z_2^1, nom) \wedge N(y_1, z_1^1, z_2^1, nom) \wedge N(y_2, z_1^2, z_2^2, gen)$.

Например, «общая теория относительности».

7. $MWE_7(y_1, y_2): N(y_1, z_1^1, z_2^1, nom) \wedge N(y_2, z_1^2, z_2^2, ins)$.

Например, «умножение столбиком».

8. $MWE_8(y_1, y_2, y_3): N(y_1, z_1^1, z_2^1, nom) \wedge N(y_2, z_1^2, z_2^2, ins) \wedge N(y_3, z_1^3, z_2^3, gen)$.

Например, «решение методом прогонки».

Обобщение шаблонов 1 и 4 можно переписать в виде

$$\bigwedge_{i=1}^{n} A(x_i, z_1^i, z_2^i, nom) \wedge N(y, z_1, z_2, nom).$$

Обобщение шаблонов 2 и 5 запишем в виде

$$N(y_1, z_1^1, z_2^1, nom) \wedge \bigwedge_{j=2}^{m} N(y_j, z_1^j, z_2^j, gen).$$

Был разработан модуль согласования словосочетаний на основе вышеперечисленных шаблонов, использующий для извлечения морфологической информации программу Mystem [1]. На вход модулю подаются лемматизированные словосочетания, которые сопоставляются с каждым шаблоном из набора. После определения требуемого шаблона словосочетание приводится в согласованный вид путем преобразования зависимых слов в форму, обусловленную формой главного слова и видом связи в словосочетании.

Данный модуль показывает приемлемые результаты, а набор модулей покрывает значительную часть используемых в качестве ключевых фраз многословных терминов. Для улучшения результатов работы можно использовать как расширение набора шаблонов, так и дополнительные способы согласования.

Основным недостатком текущей версии модуля является невозможность построения словосочетаний, в которых существительные находятся во множественном числе. Для решения данной проблемы в дальнейшем планируется использовать модуль поиска начальной формы из базового подхода, модифицировав его для поиска всех вариантов заданного лемматизированного словосочетания, а затем применить морфологический анализатор для определения нужного числа существительного.

Также к недостаткам модуля можно отнести несовершенство изменения формы слов с точки зрения лингвистики. В русском языке множество исключений, например, слова, оканчивающиеся на -мя (время, пламя и др.) не относятся к первому, второму или третьему склонению, а склоняются смешанным способом, причем при склонении к корню добавляется -ен (времени, пламени). Этот вид исключений был учтен в разработанной программе, однако, чтобы учесть все варианты исключений, встречающихся в русском языке, потребуется участие эксперта-лингвиста.

[1] MyStem – Технологии Яндекса. URL: https://tech.yandex.ru/mystem/

Разработка системы

Предобработка текста

Для лемматизации текста и построения морфологического словаря коллекции документов используется программа Mystem. Программа лемматизирует слова, используя анализ контекста для снятия лексической неоднозначности, а также предоставляет морфологическую информацию (часть речи, род, число, падеж, склонение и др.) для каждого слова. Программа распространяется бесплатно для некоммерческого использования.

Тематическое моделирование

Выбор методов тематического моделирования объясняется наличием определенных особенностей. Для сравнения некоторые из них приведены в табл. 1.

Таблица 1

Сравнение методов тематического моделирования

Название метода	Увеличение количества параметров модели с ростом числа документов	Применимость к большим наборам данных	Использование многословных терминов	Единственность и устойчивость решения
PLSA	да, есть линейная зависимость	нет	нет	нет
LDA	нет	да	нет	нет
ARTM	нет	да	нет	да
ARTM + Turbotopic (предлагаемый)	**нет**	**да**	**да**	**да**

Также для выбора базового алгоритма построения униграммных тематических моделей был проведен ряд экспериментов. Была подготовлена коллекция текстов научных статей на русском языке на основе выложенных в открытом доступе архивов журналов «Программные продукты и системы» [2], «Сибирский психологический журнал» [3] и «Cloud of Science» [4]. Статьи очищены от формул, таблиц, рисунков и библиографических ссылок, аннотация и ключевые слова были удалены. Размер коллекции составляет более двухсот шестидесяти текстов.

Для оценки результатов были выбраны следующие метрики, реализованные в библиотеке BigARTM и описанные в работе [20]: перплексия, разреженность матриц Φ и Θ, доля фоновых слов, мощность ядер тем, чистота ядер тем, контрастность ядер тем.

Первоначальные эксперименты выявили, что LDA показывает значительно худшие результаты перплексии по сравнению с PLSA и ARTM. В связи с этим дальнейшее сравнение проводилось только для двух последних алгоритмов при числе проходов по коллекции 100. Результаты представлены в табл. 2.

[2] http://www.swsys.ru/
[3] http://journals.tsu.ru/psychology/
[4] https://cloudofscience.ru/

Таблица 2

Сравнение алгоритмов PLSA и ARTM

Метрика	PLSA	ARTM
Перплексия	754.784	751.888
Разреженность матрицы Φ	0.769	0.769
Разреженность матрицы Θ	0.000	0.635
Доля фоновых слов	0.059	0.050
Средняя чистота ядер тем	0.370	0.364
Средняя контрастность ядер тем	0.787	0.788
Средняя мощность ядер тем	2085.000	2085.600

По результатам эксперимента, приведенным в табл. 2, можно увидеть, что ARTM показывает аналогичные либо лучшие результаты по сравнению с PLSA для всех метрик, за исключением средней чистоты ядер, где ухудшение незначительно. В совокупности с особенностями алгоритмов, приведенными в табл. 1, было принято решение использовать в качестве алгоритма построения униграммных тематических моделей алгоритм ARTM в реализации библиотеки BigARTM [16].

Извлечение ключевых фраз

Для извлечения многословных терминов из текстов используется адаптированный алгоритм извлечения ключевых слов Turbotopics. Суть оригинального алгоритма Turbotopics, описанного в работе [21], обобщенно состоит в следующем.

Первоначально строится униграммная модель текста при помощи алгоритма LDA. Затем производится расширение модели многословными терминами. Для каждого отдельного ключевого слова, полученного при помощи LDA, или уже добавленной фразы w осуществляется проверка в исходном тексте на наличие соседних слов u, которые с высокой вероятностью будут предшествовать w в тексте или следовать за ним. Пара таких найденных слов (u,v) или (v,u) считается многословным термином и добавляется к списку ключевых фраз. Данный алгоритм был разработан для применения в текстах на английском языке на основе алгоритма построения тематических моделей LDA и показал довольно хорошие результаты. Поэтому в данной работе он был адаптирован для работы с русскими текстами с использованием алгоритма ARTM библиотеки BigARTM.

Для определения списка ключевых слов для каждого документа изначально предполагалось использовать список наиболее часто встречающихся терминов (одно- и многословных) для каждой темы, к которой относится данный документ. Однако этот подход привел к тому, что из документа извлекались ключевые слова темы, а не самой статьи: для различных документов списки ключевых слов были очень похожи, а термины, которые должны быть ключевыми исходя из текста статьи, не попадали в список из-за низкой частоты встречаемости. Для решения данной проблемы было предложено использовать TF-IDF – статистическую меру, оценивающую важность каждого слова для документа, в котором оно встречается [22]. Наибольшее значение TF-IDF будут иметь слова, которые часто встречаются в данном документе, но редко встречаются в остальных документах коллекции.

Общий вид системы

В рамках исследования была разработана система, позволяющая строить расширенные тематические модели, включающие многословные термины, для коллекций научных статей на русском языке. Система написана на языке Python 3 с использованием библиотеки BigARTM. Используемые в системе алгоритмы из этой библиотеки были настроены таким

образом, чтобы получить оптимальные результаты относительно различных метрик (перплексия, разреженность и др.) при использовании текстов научных статей на русском языке. Обобщенная схема работы системы представлена ниже. Далее приведено подробное описание процесса построения расширенной тематической модели и извлечения ключевых фраз разработанной системой.

Схема работы системы

Опишем схему работы системы как последовательность шагов.

Шаг 0. На вход системе подается коллекция документов в формате .txt. Каждый документ должен быть представлен одним файлом, все документы помещены в одну директорию, путь к которой передается программе в качестве параметра.

Шаг 1. В модуле предобработки текста каждый документ очищается от специальных символов (отличных от кириллических и латинских букв), из документа удаляются стоп-слова, все слова приводятся к нижнему регистру. Далее строится корпус коллекции в формате последовательного Vowpal Wabbit.

Шаг 2. Производится вызов программы Mystem, на вход которой подается файл с построенным на предыдущем этапе работы корпусом. Результатом работы является файл лемматизированного корпуса (формат, аналогичный полученному ранее корпусу, только каждое слово заменено его начальной формой), а также файл морфологического словаря, где каждой строке соответствует слово и описывающая его морфологическая информация.

Шаг 3. На лемматизированном корпусе производится поиск ключевых слов и *n*-грамм с помощью алгоритма Turbotopics.

Шаг 4. Найденные алгоритмом Turbotopics *n*-граммы преобразуются из лемматизированного вида в согласованный с использованием шаблонов, описанных выше, и морфологического словаря, полученного на шаге 2.

Шаг 5. Для лемматизированного корпуса строится тематическая модель коллекции документов с использованием алгоритма ARTM. Параметры алгоритма можно подобрать автоматически или использовать заранее вычисленные (так как подбор параметров – задача весьма трудоемкая и занимает значительное время).

Шаг 6. Полученная на шаге 5 тематическая модель расширяется с помощью многословных терминов, извлеченных из коллекции на шаге 3 и согласованных на шаге 4.

Шаг 7. Для каждого документа строится словарь TF-IDF: с каждым словом в лемматизированном документе сопоставляется значение меры TF-IDF. Слова в словаре сортируются по убыванию значения меры.

Шаг 8. На основе матрицы распределения тем по документам с каждым документом сопоставляется набор присутствующих в нем тем и их вероятностей (учитываются только темы, вероятность появления которых в данном документе превышает порог $\delta = \frac{1}{N_t}$, где N_t – количество тем в модели).

После этого сравниваются два множества: первые N_1 слов из отсортированного словаря TF-IDF и первые N_2 слов и словосочетаний для каждой темы, отсортированных по вероятности встретить этот термин в документе. Итоговыми ключевыми словами для темы документа будет пересечение этих множеств. N_1 и N_2 могут настраиваться; по умолчанию эти значения равны 100 и 300 соответственно. Такие значения параметров были подобраны эмпирическим путем, чтобы каждому документу в среднем соответствовало порядка 5–10 ключевых слов и фраз.

Результатом работы программы является текстовый файл, содержащий следующую информацию:

- название исходного документа;
- список тем, для каждой из которых указана вероятность содержания ее в тексте как десятичная дробь от 0 до 1;
- список ключевых слов и фраз для каждой темы.

Также для пользователя доступен файл с описанием тем, где с каждой темой сопоставлено множество слов и словосочетаний с наибольшей вероятностью для этой темы.

Полученные результаты

Поскольку невозможно автоматически оценить интерпретируемость тем и приемлемость извлеченных ключевых фраз, результаты были оценены вручную. Далее приведены несколько примеров работы алгоритма для различных публикаций разной направленности. Некоторые из наиболее частотных слов и фраз для первых пяти тем расширенной тематической модели коллекции, представлены в табл. 3.

Таблица 3

Расширенная тематическая модель коллекции научных статей

Тема	Расширенная тематическая модель
Тема 1	'алгоритм', 'решение', 'задача', 'значение', 'вершина', 'значение параметра', 'время распознавания', 'класс объекта', 'обработка информации', 'алгоритм поиска', 'вершина графа', 'изображение объекта', 'граница решения', 'задача поиска', 'граф решения'
Тема 2	'метод', 'данные', 'алгоритм', 'классификация', 'текст', 'слово', 'классификатор', 'обучение', 'значение параметра', 'класс объекта', 'множество признака', 'представление текста', 'процесс обучения', 'метод классификации', 'построение модели', 'задача классификации', 'качество классификации', 'обучение классификатора', 'классификация текста'
Тема 3	'человек', 'ребенок', 'психологический', 'группа', 'отношение', 'стратегия восприятия', 'процесс формирования', 'образ мира', 'группа испытуемая', 'уровень развития', 'респондент группы', 'развитие ребенка'
Тема 4	'система', 'управление', 'процесс', 'модель', 'требование', 'разработка', 'система управления', 'орган управления', 'процесс разработки', 'модель прогнозирования', 'критерий эффективности проекта', 'этап прогнозирования', 'критерий эффективности', 'эффективность проекта'
Тема 5	'исследование', 'отношение', 'испытуемый', 'элемент', 'диагностический', 'результат исследования', 'значение параметра', 'удовлетворенность отношения', 'процесс формирования', 'поиск решения', 'вид деятельности', 'группа испытуемая', 'удовлетворенность брака', 'формирование религиозности’

По представленным в табл. 3 результатам можно отметить, что темы из разных предметных областей (технические науки и психология) очень хорошо различимы в тематической модели. При этом граница между более узкими темами не настолько четкая: если тема 4 довольно хорошо интерпретируется как отдельная предметная область, связанная с управлением проектами и процессом разработки, темы 1 и 2 связаны с классификацией и распознаванием, а темы 3 и 5 – с психологической диагностикой. При этом важно заметить, что в теме 5 многословные термины («удовлетворенность отношения», «формирование религиозности» и т. д.) улучшают интерпретируемость темы как относящуюся к психологии, тогда как термины «исследование», «испытуемый» являются более общими.

В табл. 4 представлены извлеченные программой ключевые слова и фразы для нескольких научных публикаций.

Таблица 4

Ключевые слова и фразы

№	Название статьи	Извлеченные ключевые слова и фразы
1	Алгоритм детектирования объектов на фотоснимках с низким качеством изображения	объект, класс, изображение, набор, автокодировщик, обучение, объект, класс, набор, изображение, слой, пиксел
2	Проектирование интерфейса программного обеспечения с использованием элементов искусственного интеллекта	программный, пользователь, система управления, уровень развития, нечеткий, интерфейс, характеристика, эксперт, система управления
3	Родительское отношение как фактор формирования религиозности личности	ребенок, отношение, родитель, формирование, религиозность, религиозный, религия, семья, родительский, решение задачи
4	Прогнозирование платежеспособности клиентов банка на основе методов машинного обучения и марковских цепей	прогнозирование, состояние, клиент, классификатор, ак, заемщик, решение задачи, дерево решения
5	Разработка системы хранения ансамблей нейросетевых моделей	данные, модель, набор, ансамбль, ряд, преобразование, хранение, нейросетевой, оценка качества, процесс формирования, классификация текста

Можно утверждать, что извлеченные ключевые слова и фразы соответствуют содержанию статей и хорошо определяют предметную область исследований. При этом можно заметить, что в некоторых случаях они дают большее представление о содержании публикации, чем ее название: например, ключевая фраза «дерево решения» дает понять, что в качестве алгоритма машинного обучения в четвертой статье использовались деревья решений, а ключевая фраза «классификация текста» в статье 5 указывает, что ансамбли нейросетевых моделей здесь использовались для классификации текста (а не только изображений, например).

Заключение

Тематические модели позволяют автоматически систематизировать большие коллекции текстовых документов на естественном языке, повышают эффективность информационного поиска. В ходе данного исследования была разработана система построения тематических моделей и извлечения ключевых слов и фраз для текстов научных статей на русском языке. Для проведения экспериментов была подготовлена коллекция «очищенных» текстов научных статей на русском языке из размещенных в открытом доступе журналов [5].

[5] Коллекция текстов доступна по ссылке: https://github.com/Serenitas/topic-modeller/.

Разработанная система способна строить расширенные тематические модели, включающие, помимо униграмм, словосочетания в согласованном виде. Для каждого документа предоставляется набор содержащихся в нем тем с указанными вероятностями и ключевыми словами и фразами для каждой темы.

Благодаря расширению тематической модели многословными терминами темы хорошо интерпретируются. Извлекаемые ключевые слова и фразы соответствуют содержанию документа.

Предложенный в статье подход может быть полезен при построении рекомендательных систем и систем автореферирования.

Список литературы

1. *Коршунов А., Гомзин А.* Тематическое моделирование текстов на естественном языке // Тр. Ин-та системного программирования РАН. 2012. С. 215–242.

2. *Воронцов К. В.* Вероятностное тематическое моделирование. 2013. URL: http://www.machinelearning.ru/wiki/images/2/22/Voron-2013-ptm.pdf

3. *Yin Zhang, Min Chen, Dijiang Huang, Di Wu, Yong Li.* iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization // Future Generation Computer Systems. 2017. Vol. 66. P. 30–35.

4. *Kar Wai Lim, Wray Buntine.* Bibliographic Analysis with the Citation Network Topic Model // JMLR: Workshop and Conference Proceedings. 2014. Vol. 39. P. 142–158.

5. *Ye Chen, Bei Yu, Xuewei Zhang, Yihan Yu.* Topic modeling for evaluating students' reflective writing: a case study of pre-service teachers' journals // LAK '16 Proceedings of the Sixth International Conference on Learning Analytics & Knowledge. 2016. P. 1–5.

6. *Zhao X. W., Wang J., He Y., Nie J.-Y., Li X.* Originator or propagator: Incorporating social role theory into topic models for Twitter content analysis // Proceedings of the 22nd ACM International Conference on Conference on Information and Knowledge Management. CIKM '13. New York, NY, USA: ACM, 2013. P. 1649–1654.

7. *Varshney D., Kumar S., Gupta V.* Modeling information diffusion in social networks using latent topic information // Intelligent Computing Theory / Ed. by D.-S. Huang, V. Bevilacqua, P. Premaratne. Springer International Publishing, 2014. Vol. 8588 of Lecture Notes in Computer Science. P. 137–148.

8. *Pinto J. C. L., Chahed T.* Modeling multi-topic information diffusion in social networks using latent Dirichlet allocation and Hawkes processes // Tenth International Conference on Signal-Image Technology & Internet-Based Systems. 2014. P. 339–346.

9. *Vulic I., De Smet W., Tang J., Moens M.-F.* Probabilistic topic modeling in multilingual settings: an overview of its methodology and applications // Information Processing & Management. 2015. Vol. 51, no. 1. P. 111–147.

10. *Cui W., Liu S., Tan L., Shi C., Song Y., Gao Z., Qu H., Tong X.* TextFlow: Towards better understanding of evolving topics in text // IEEE transactions on visualization and computer graphics. 2011. Vol. 17, no. 12. P. 2412–2421.

11. *Allahyari M., Kochut K. J.* Semantic Tagging Using Topic Models Exploiting Wikipedia Category Network // IEEE Tenth International Conference on Semantic Computing (ICSC). 2016.

12. *Allahyari M., Kochut K.* Semantic Context-Aware Recommendation via Topic Models Leveraging Linked Open Data // International Conference on Web Information Systems Engineering. WISE 2016. Lecture Notes in Computer Science. Vol. 10041. P. 263–277.

13. *Золотарев О. В., Шарнин М. М., Клименко С. В.* Семантический подход к анализу террористической активности в сети Интернет на основе методов тематического моделирования // Вестн. Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2016. № 3. С. 64–71.

14. *Hofmann T.* Probabilistic Latent Semantic Indexing // Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99). 1999. P. 289–296.

15. *Blei D. M., Ng A. Y., Jordan M. I.* Latent Dirichlet allocation // Journal of Machine Learning Research. 2003. No. 3. P. 993–1022.

16. *Воронцов К. В., Фрей А. И., Ромов П. А., Янина А. О., Суворова М. А., Апишев М. А.* BigARTM: библиотека с открытым кодом для тематического моделирования больших текстовых коллекций. 2014. URL: http://docplayer.ru/27022431-Bigartm-biblioteka-s-otkrytym-kodom-dlya-tematicheskogo-modelirovaniya-bolshih-tekstovyh-kollekciy.html

17. *Кипяткова И. С., Карпов А. А.* Аналитический обзор систем распознавания русской речи с большим словарем // Тр. СПИИРАН. 2010. Вып. 12. С. 7–20.

18. *Большакова Е. И., Баева Н. В., Бордаченкова Е. А., Васильева Н. Э., Морозов С. С.* Лексико-синтаксические шаблоны в задачах автоматической обработки текста // Компьютерная лингвистика и интеллектуальные технологии: Тр. Междунар. конф. «Диалог 2007». М.: Изд-во РГГУ, 2007. С. 70–75.

19. *Загорулько М. Ю., Сидорова Е. А.* Система извлечения предметной терминологии из текста на основе лексико-синтаксических шаблонов // Тр. XIII Междунар. конф. «Проблемы управления и моделирования в сложных системах» / Под ред. Е. А. Федосова, Н. А. Кузнецова, В. А. Виттиха. 2011. С. 506–511.

20. *Воронцов К. В., Потапенко А. А.* Регуляризация вероятностных тематических моделей для повышения интерпретируемости и определения числа тем // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Междунар. конф. «Диалог» М.: Изд-во РГГУ, 2014. Вып. 13 (20). С. 676–687.

21. *Blei D. M., Lafferty J. D.* Visualizing Topics with Multi-Word Expressions // Semantic Scholar. 2009. URL: https://arxiv.org/pdf/0907.1013.pdf

22. *Leskovec J., Rajaraman A., Ullman J. D.* Mining of Massive Datasets. 2014. 513 p.

Материал поступил в редколлегию 03.03.2018

T. V. Batura [1, 2], **S. E. Strekalova** [1]

[1] *Novosibirsk State University*
1 Pirogov Str., Novosibirsk, 630090, Russian Federation

[2] *A. P. Ershov Institute of Informatics Systems SB RAS*
6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation

tatiana.v.batura@gmail.com, svetlana.strekalova@gmail.com

AN APPROACH TO BUILDING EXTENDED TOPIC MODELS OF RUSSIAN TEXTS

A new approach to building extended topic models of Russian scientific texts is described in this article. An extended topic model is a topic model containing not only one-word terms, but also multiword terms (key phrases). Such models are better interpreted for the user and more accurately describe the subject area of the document than models consisting only of unigrams (separate words).

On the basis of the proposed approach, a system was developed which, as a result of the work, provides for each document a set of topics with probabilities, key words and phrases for each topic.

The approach proposed in the article can be useful for development of recommendation systems and summarization systems.

Keywords: topic models, text processing, keyword extraction, multiword term extraction, topic detection.

References

1. Korshunov A., Gomzin A. Topic modelling of natural languge texts. *Proceedings of the Institute for System Programming of the RAS*, 2012, p. 215–242. (in Russ.)

2. Vorontsov K. V. Probabilistic topic modeling. 2013. URL: http://www.machinelearning.ru/wiki/images/2/22/Voron-2013-ptm.pdf (in Russ.)

3. Yin Zhang, Min Chen, Dijiang Huang, Di Wu, Yong Li iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. *Future Generation Computer Systems*, 2017, vol. 66, p. 30–35.

4. Kar Wai Lim, Wray Buntine, Bibliographic Analysis with the Citation Network Topic Model. *JMLR: Workshop and Conference Proceedings*, 2014, vol. 39, p. 142–158.

5. Ye Chen, Bei Yu, Xuewei Zhang, Yihan Yu Topic modeling for evaluating students' reflective writing: a case study of pre-service teachers' journals. *LAK '16 Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, 2016, p. 1–5.

6. Zhao X. W., Wang J., He Y., Nie J.-Y., Li X. Originator or propagator?: Incorporating social role theory into topic models for Twitter content analysis. *Proceedings of the 22nd ACM International Conference on Conference on Information and Knowledge Management. CIKM '13*. New York, NY, USA, ACM, 2013, p. 1649–1654.

7. Varshney D., Kumar S., Gupta V. Modeling information diffusion in social networks using latent topic information. *Intelligent Computing Theory*. Ed. by D.-S. Huang, V. Bevilacqua, P. Premaratne. Springer International Publishing, 2014, vol. 8588 of Lecture Notes in Computer Science, p. 137–148.

8. Pinto J. C. L., Chahed T. Modeling multi-topic information diffusion in social networks using latent Dirichlet allocation and Hawkes processes. *Tenth International Conference on Signal-Image Technology & Internet-Based Systems*, 2014, p. 339–346.

9. Vulic I., De Smet W., Tang J., Moens M.-F. Probabilistic topic modeling in multilingual settings: an overview of its methodology and applications. *Information Processing & Management*, 2015, vol. 51, no. 1, p. 111–147.

10. Cui W., Liu S., Tan L., Shi C., Song Y., Gao Z., Qu H., Tong X. TextFlow: Towards better understanding of evolving topics in text. *IEEE transactions on visualization and computer graphics,* 2011, vol. 17, no. 12, p. 2412–2421.

11. Allahyari M., Kochut K.J. Semantic Tagging Using Topic Models Exploiting Wikipedia Category Network. *IEEE Tenth International Conference on Semantic Computing* (*ICSC*), 2016.

12. Allahyari M., Kochut K. Semantic Context-Aware Recommendation via Topic Models Leveraging Linked Open Data. *International Conference on Web Information Systems Engineering. WISE. Lecture Notes in Computer Science*, 2016, vol. 10041, p. 263–277.

13. Zolotarev O. V., Sharnin M. M., Klimenko S. V. Semantic approach for terroristic activity analysis in the Internet based on topic modelling methods. *Russian New University Bulletin. Series: Complex systems: models, analysis and control*, 2016, vol. 3, p. 64–71. (in Russ.)

14. Hofmann T. Probabilistic Latent Semantic Indexing. *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR-99*), 1999, p. 289–296.

15. Blei D. M., Ng A. Y., Jordan M. I. Latent Dirichlet allocation. *Journal of Machine Learning Research*. 2003, no. 3, p. 993–1022.

16. Vorontsov K. V., Frey A. I., Romov P. A., Yanina A. O., Suvorova M. A., Apishev M. A. BigARTM: Open Source Library for Topic Modeling of Large Text Collections. 2014. URL: http://docplayer.ru/27022431-Bigartm-biblioteka-s-otkrytym-kodom-dlya-tematicheskogo-modelirovaniya-bolshih-tekstovyh-kollekciy.html (in Russ.)

17. Kipyatkova I. S., Karpov A. A. Analytical review of recognition systems for Russian language with large dictionary. *Proceedings of SPIIRAS*, 2010, vol. 12, p. 7–20. (in Russ.)

18. Bolshakova E. I., Baeva N. V., Bordachenkova E. A., Vasilyeva N. E., Morozov S. S. Lexico-syntactic templates in natural language processing. *Computational linguistics and intellectual technologies: Proceedings of the international conference "Dialogue 2007"*. Moscow, RSUH, 2007, p. 70–75. (in Russ.)

19. Zagorulko M. Yu., Sidorova E. A. System of extraction of subject terminology from text based on lexico-syntactic templates. *Proceedings of XIII International conference "Problems of control and modelling in complex systems".* Eds. E. A. Fedosova, N. A. Kusnetsova, V. A. Vittikh. 2011, p. 506–511. (in Russ.)

20. Vorontsov K. V., Potapenko A. A. Regularization of Probabilistic Topic Models to Improve Interpretability and Determine the Number of Topics. *Computational linguistics and intellectual technologies: Proceedings of the annual international conference "Dialogue"*. Moscow, RSUH, 2014, vol. 13 (20), p. 676–687. (in Russ.)

21. Blei D. M., Lafferty J. D. Visualizing Topics with Multi-Word Expressions. *Semantic Scholar*, 2009. URL: https://arxiv.org/pdf/0907.1013.pdf

22. Leskovec J., Rajaraman A., Ullman J. D. Mining of Massive Datasets, 2014, 513 p.

For citation:

Batura T. V., Strekalova S. E. An Approach to Building Extended Topic Models of Russian Texts. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 5–18. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-5-18

УДК 004.043
DOI 10.25205/1818-7900-2018-16-2-19-30

И. В. Бельченко, **Р. А. Дьяченко**

Кубанский государственный технологический университет
ул. Московская, 2, Краснодар, 350072, Россия

ilur@mail.ru, emessage@rambler.ru

МЕТОДИКА ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ НЕБОЛЬШИХ ИНФОРМАЦИОННЫХ СИСТЕМ ЗА СЧЕТ ОПТИМАЛЬНОЙ РЕСТРУКТУРИЗАЦИИ ДАННЫХ НА ОСНОВЕ МНОГОМОДАЛЬНОГО РАСПРЕДЕЛЕНИЯ АТРИБУТОВ

Рассматривается системный подход к повышению производительности небольших информационных систем за счет оптимальной реструктуризации табличных структур данных. Авторами сформулирована задача оптимизации количества информационных блоков, необходимых для выполнения группы запросов на считывание информации, предложена целевая функция и структурные ограничения. Проанализирована невозможность использования грубых методов поиска оптимального решения. Предложена методика многомодального распределения атрибутов в зависимости от частоты появления в группе запросов. Проведен эксперимент, подтверждающий эффективность разработанной методики для небольших информационных систем.

Ключевые слова: система поддержки принятия решений, оптимизация, структуры данных, базы данных, системный анализ.

Введение

Большинство многопользовательских информационных веб-систем выделяется такими требованиями, как оперативное взаимодействие с пользователем [1]. Эффективное исполнение данного требования зависит не только от аппаратной составляющей, включающей в себя серверное оборудование и линии связи, но и от реализации программных компонентов, среди которых программное приложение, реализованное с применением веб-технологий и система управления базой данных (СУБД). В статье рассмотрена методика повышения производительности информационной системы, за счет уменьшения среднего времени выполнения группы запросов на чтение информации базы данных. От структуры данных, способах ее физического размещения на жестких дисках зависит количество обращений к дисковым накопителям, которые сопровождаются соответствующими прерываниями и задержками по времени [2].

Важным понятием при рассмотрении вопроса физической организации баз данных является понятие блока. Блок – это минимальный адресуемый элемент внешней памяти, с помощью которого осуществляется обмен информацией между оперативной и внешней памятью. Запись и чтение блоков осуществляется через буферную часть оперативной памяти. Для организации каждого файла базы данных в зависимости от его размера во внешней памяти выделяется от одного до N блоков, где размещаются записи. В одном блоке могут разместиться все записи или в нескольких блоках одна запись, или в одном блоке одна запись. От этого

Бельченко И. В., *Дьяченко Р. А.* Методика повышения производительности небольших информационных систем за счет оптимальной реструктуризации данных на основе многомодального распределения атрибутов // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 19–30.

будет зависеть время считывания и записи элементов файла. Записи в блоках размещаются плотно, без промежутков, последовательно. В блоке часть памяти отводится под служебную информацию: относительный адрес свободных участков памяти, указатели на следующий блок и т. д. Для хранения поступающих данных, которые должны размещаться в одном блоке, заполненном уже полностью, выделяется дополнительный блок памяти в области переполнения записи, организованной в виде одного блока, где записи связываются указателями в одну цепь.

Таким образом, на скорость поиска влияют: объем блока в байтах, объем файла, количество записей в блоке файла, количество записей в блоке индекса, количество блоков в файле, доля резервной части блока, число полей в записи, размер записи в байтах [2].

Постановка задачи вертикальной реструктуризации табличных структур данных

Процесс построения оптимальной модели данных информационной системы включает оптимальное вертикальное распределение таблиц базы данных по блокам на дисковом накопителе. Основным критерием оптимизации модели данных информационной системы является минимальный размер строки таблицы реляционной базы данных, позволяющий в одном блоке хранить больше данных и, как следствие, минимизировать количество операций чтения блоков данных с жесткого диска при выполнении запросов к базе данных. Это достигается за счет уменьшения объема данных, побочно участвующих в запросе [3].

В рамках методики предлагается разделить таблицы базы данных на несколько сущностей, связанных отношением один к одному. В соответствии с принципами блочного хранения данных в СУБД каждая таблица будет храниться в отдельном наборе блоков. При выполнении запроса на чтение информации СУБД считывает блоки данных с жесткого диска в оперативную память каждой таблицы, атрибуты которой участвуют в запросе.

Задача повышения производительности информационной системы сводится к поиску оптимального разделения табличных структур базы данных с учетом конкретной группы запросов на чтение информации, выявленной статистически в рамках жизненного цикла БД [4].

Оптимизация табличных структур данных информационной системы

Для формализации задачи рассмотрим множества и параметры, влияющие на скорость обработки запросов на чтение информации к исследуемой таблице базы данных.

1. Целочисленный параметр TS, равный количеству атрибутов в исследуемой таблице.

2. Вектор типов данных $DBT = \left\{ dbt_{idbt} \middle| idbt = \overline{1, ndbt} \right\}$, которые поддерживаются конкретной выбранной СУБД. Элемент вектора – занимаемый элементом типа размер данных в байтах памяти.

3. Набор атрибутов (столбцов таблицы) TA, который задан бинарной матрицей, элемент которой $ta_{ita,\,jta}$ равен единице, если столбец ita таблицы имеет тип jta, $ita = 1,...,TS$, $jta = 1,...,ndbt$.

4. Множество, представляющее группу запросов $Q = \left\{ q_{iq} \middle| iq = \overline{1, nq} \right\}$ на чтение информации из таблицы базы данных, элемент множества – кортеж из двух элементов $q_{iq} = \left\{ SFQ_{iq}, QA_{iq} \right\}$, где SFQ_{iq} – числовой параметр, равный частоте появления запроса за выбранный период времени, $QA_{iq} = \left\{ qa_{iqa} \middle| iqa = \overline{1, TS} \right\}$ – бинарный вектор, размерность которого равна количеству атрибутов таблицы TS. $qa_{iqa} = 1$, если атрибут таблицы TA участвует в за-

просе, и 0 в противном случае. nq – количество запросов в статистической выборке, выявленной в рамках жизненного цикла БД.

5. Множество индексов, характеризующихся набором полей таблицы, по которым построен индекс $IN = \left\{ in_{iin} \middle| iin = \overline{1, nin} \right\}$. Элемент множества $in_{iin} = \left\{ in_{iin,jin} \middle| jin = \overline{1, TS} \right\}$ – бинарный вектор, размерность которого равна количеству атрибутов таблицы TS, $in_{iin,jin} = 1$, если атрибут jin таблицы TA участвует в индексе in_{iin}, и 0 в противном случае.

6. Хранимые процедуры и функции $PF = \left\{ pf_{ipf} \middle| ipf = \overline{1, npf} \right\}$, характеризующиеся набором полей, используемых в теле хранимой процедуры или функции. Элемент множества $pf_{ipf} = \left\{ pf_{ipf,jpf} \middle| jpf = \overline{1, TS} \right\}$ – бинарный вектор, размерность которого равна количеству атрибутов таблицы TS, $pf_{ipf,jpf} = 1$, если атрибут jpf таблицы TA участвует в теле хранимой процедуры или функции pf_{ipf}, и 0 в противном случае.

7. Множество триггеров базы данных $TG = \left\{ tg_{itg} \middle| itg = \overline{1, ntg} \right\}$, характеризующихся набором полей таблицы, используемых в теле триггера. Элемент множества $tg_{itg} = \left\{ tg_{itg,jtg} \middle| jtg = \overline{1, TS} \right\}$ – бинарный вектор, размерность которого равна количеству атрибутов таблицы TS, $tg_{itg,jtg} = 1$, если атрибут jtg таблицы TA участвует в теле триггера tg_{itg}, и 0 в противном случае.

Анализ влияния количества физических блоков данных, используемых таблицей базы данных, на общее время выполнения группы запросов к ней

Множество запросов Q к рассматриваемой таблице обрабатывается СУБД за время $T(Q, TA, DBT)$. Временные затраты $T(Q, TA, DBT)$ можно представить в виде суммы временных затрат на чтение блоков данных таблиц $T_h(Q, TA, DBT)$, участвующих в запросах Q, и остальных временных затрат $T_o(Q, TA, DBT)$, к которым относятся временные затраты на выполнение плана обработки запроса, на передачу информации и т. д.:

$$T(Q, TA, DBT) = T_h(Q, TA, DBT) + T_o(Q, TA, DBT).$$

В рамках методики предлагается уменьшить слагаемое, влияющее на общее время выполнения запроса $T_h(Q, TA, DBT)$. Временные затраты $T_h(Q, TA, DBT)$ в общем виде зависят от количества операций чтения блоков данных таблиц с жесткого диска. Пусть временная задержка, связанная со считыванием одного блока данных равна T_b, тогда

$$T_h(Q) = \left(\sum_{iq}^{nq} B(q_{iq}, TA, DBT) \right) \cdot T_b,$$

где $B(q_{iq}, TA, DBT)$ – число информационных блоков, которые необходимо считать с жесткого диска в кэш СУБД для дальнейшего выполнения запроса q_{iq} к таблице, заданной бинарной матрицей TA. Кэш СУБД находится в оперативной памяти вычислительного устройства.

Функция $B(q_{iq}, TA, DBT)$ вычисляется как отношение:

$$B\left(q_{iq},TA,DBT\right)=\frac{RC\cdot RS\left(q_{iq},TA,DBT\right)}{DB},$$

где

RC – количество строк в рассматриваемой таблице;

DB – фиксированный размер блока данных выбранной СУБД (в большинстве СУБД он равен 8Кб);

$RS\left(q_{iq},TA,DBT\right)$ – величина, характеризующая дисковое пространство, занимаемое одной строкой таблицы в байтах:

$$RS\left(q_{iq},TA,DBT\right)=RSS\left(q_{iq},TA,DBT\right)+RST\left(q_{iq},TA,DBT\right).$$

Здесь

$RSS\left(q_{iq},TA,DBT\right)$ – количество памяти, занимаемое служебными отметками СУБД для строки, считываемое при выполнении запроса q_{iq};

$RST\left(q_{iq},TA,DBT\right)$ – количество памяти, занимаемое атрибутами таблицы в строке, считываемое при выполнении запроса q_{iq}.

Параметры RC и DB остаются неизменными.

Так как временную задержку T_b, связанную со считыванием одного блока данных, допускается считать постоянной величиной, на сумму временных затрат на чтение блоков данных таблицы TA – $T_h\left(Q,TA,DBT\right)$ влияет количество блоков, необходимое для считывания, которое вычисляется как функция

$$F\left(Q,TA,DBT\right)=\sum_{iq}^{nq}B\left(q_{iq},TA,DBT\right).$$

Подставим в формулу $F\left(Q,TA,DBT\right)$, формулу функции $B\left(q_{iq},TA,DBT\right)$. Функция, определяющая количество блоков, необходимых для считывания с жесткого диска в оперативную память при выполнении множества запросов Q к рассматриваемой таблице *TA*:

$$F\left(Q,TA,DBT\right)=\sum_{iq}^{nq}\left(\frac{RC\cdot RS\left(q_{iq},TA,DBT\right)}{DB}\right).$$

Методика уменьшения количества физических блоков данных, используемых СУБД для выполнения группы запросов к таблице, за счет ее оптимального разделения на дочерние таблицы

В рамках методики предлагается разделить рассматриваемую таблицу на $NB\in\left[1;TS\right]$ дочерних таблиц, связанных с родительской отношением один к одному, 1:1.

Введем следующую переменную:

$$x_{ij}=\begin{cases}1, & \text{если } j\text{-й атрибут нужно выделить в } i\text{-ю таблицу,}\\ 0 & \text{в противном случае.}\end{cases}$$

Переменные представляют собой бинарную матрицу для таблицы реляционной базы данных размерностью $TS \times TS$, где TS – количество атрибутов таблицы. Строки матрицы соответствуют таблицам, на которые разбивается родительская таблица, а столбцы соответствуют их атрибутам.

Количество блоков $BM(Q, RC, DB, DBT, TA, X)$, которое необходимо считать с жесткого диска для выполнения множества запросов Q к таблице TA, вычисляется как функция, равная сумме блоков, которые необходимо считать с жесткого диска для выполнения множества запросов Q к каждой из дочерних таблиц. Максимальное количество дочерних таблиц равно числу атрибутов родительской таблицы TA и равно NB.

$$BM(Q, RC, DB, DBT, TA, X) =$$
$$= \sum_{iq=1}^{nq} \left| \frac{RC \cdot RSM(DBT, TA, X_1) \cdot FQ(q_{iq}, X_1)}{DB} \right| +$$
$$+ \sum_{iq=1}^{nq} \left| \frac{RC \cdot RSM(DBT, TA, X_{irs}) \cdot FQ(q_{iq}, X_{irs})}{DB} \right| +$$
$$+ \sum_{iq=1}^{nq} \left| \frac{RC \cdot RSM(DBT, TA, X_{nb}) \cdot FQ(q_{iq}, X_{nb})}{DB} \right|,$$

где

$$RSM(DBT, TA, X_{irs}) = \left(\sum_{j}^{TS} \left[x_{irs,j} \cdot \left(\sum_{idbt}^{ndbt} DBT_{idbt} \cdot TA_j \right) \right] + RDS(DBT, TA, X_{irs}) \right),$$

$$FQ(q_{iq}, X_{irs}) = \begin{cases} q_{iq}(SFQ), & \text{если } \sum_{u}^{TS} (X_{irs})_u \cdot (q_{iq}(QA))_u > 0, \\ 0 & \text{в противном случае.} \end{cases}$$

$$irs = 1, ..., nb, \quad j = 1, ..., TS, \quad idbt = 1, ..., ndbt.$$

Параметры RC и DB являются постоянными, RSM – функция, характеризующая количество информации, которая занимает одну строку дочерней таблицы irs в байтах, $RDS(DBT, TA, X_{irs})$ – функция, характеризующая дисковое пространство, занимаемое служебными отметками СУБД в строке дочерней таблицы irs в байтах. Следовательно, задача повышения производительности системы сводится к поиску такого разделения таблицы на дочерние, при котором сумма блоков, которые необходимо считать в кэш СУБД для выполнения множества запросов Q, минимально.

Целевая функция

$$\min_{x_{irs,j}} \sum_{iq,irs} \left| \frac{\left(\left(\sum_{j}^{TS} \left[x_{irs,j} \cdot \left(\sum_{idbt}^{ndbt} dbt_{idbt} \cdot ta_j \right) \right] + RDS(DBT, TA, x_{irs}) \right) \right) \cdot RC \cdot FQ(q_{iq}, x_{irs})}{DB} \right|,$$

где

$$FQ\left(q_{iq},x_{irs}\right)=\begin{cases}q_{iq}\left(SFQ\right), & \text{если } \sum\limits_{u}^{TS}\left(x_{irs}\right)_u\cdot\left(q_{iq}\left(QA\right)\right)_u>0,\\ 0 & \text{в противном случае.}\end{cases}$$

$$irs=1,...,nb,\quad j=1,...,TS,\quad idbt=1,...,ndbt.$$

При структурных ограничениях:

1) каждый атрибут родительской таблицы может присутствовать только в одной дочерней таблице

$$\sum_{m_1}x_{m_1,i_1}=1,\quad m_1=1,...,TS,\quad i_1=1,...,TS;$$

2) атрибуты таблицы, используемые при построении индексов, должны принадлежать хотя бы одной дочерней таблице

$$\forall ix,\quad ix=1,...,\left|IN\right|:\prod_{m_2}^{TS}\left[\sum_{i_2}^{TS}\left(x_{m_2,i_2}\cdot in_{ix,i_2}-in_{ix,i_2}\right)\right]=0,$$

$$m_2=1,...,TS,\quad i_2=1,...,TS;$$

3) атрибуты таблицы, используемые в теле хранимых процедур или функций, должны принадлежать хотя бы одной дочерней таблице

$$\forall px,\quad px=1,...,\left|PF\right|:\prod_{m_3}^{TS}\left[\sum_{i_3}^{TS}\left(x_{m_3,i_3}\cdot pf_{px,i_3}-pf_{px,i_3}\right)\right]=0,$$

$$m_3=1,...,TS,\quad i_3=1,...,TS;$$

4) атрибуты таблицы, используемые в работе триггеров исследуемой таблицы, должны принадлежать хотя бы одной дочерней таблице

$$\forall tx,\quad tx=1,...,\left|TG\right|:\prod_{m_4}^{TS}\left[\sum_{i_4}^{TS}\left(x_{m_4,i_4}\cdot tg_{tx,i_4}-tg_{tx,i_4}\right)\right]=0,$$

$$m_4=1,...,TS,\quad i_4=1,...,TS;$$

5) отношение количества физических блоков данных, необходимого для хранения данных рассматриваемой таблицы до применения методики, к количеству блоков, необходимому для хранения данных в дочерних таблицах, полученных после применения методики, не должно превышать заданного параметра $TSIZE$ $\left(TSIZE\in\left(0;1\right]\right)$

$$TSIZE=\frac{\sum\limits_{iq}^{nq}\left(\dfrac{RC\cdot RS\left(q_{iq},TA,DBT\right)}{DB}\right)}{\dfrac{\left(\left(\sum\limits_{j}^{TS}\left[x_{irs,j}\cdot\left(\sum\limits_{idbt}^{ndbt}dbt_{idbt}\cdot ta_j\right)\right]+RDS\left(DBT,TA,x_{irs}\right)\right)\right)\cdot RC\cdot FQ\left(q_{iq},x_{irs}\right)}{DB}}.$$

Методика нахождение оптимального разделения таблицы на дочерние для выполнения группы запросов путем многомодального распределения атрибутов таблицы по частоте их появлений в запросах

Целевая функция не линейна, а также не линейны ограничения. Переменная X – бинарная матрица размерностью $TS \times TS$. Представим переменную в виде машинного слова длиной $TS \cdot TS$. Следовательно, количество возможных комбинаций переменной определяется как $2^{TS \cdot TS}$. Исходя из этого задача обладает экспоненциальной сложностью и является *NP*-трудной [5; 6].

Для решения задачи разработана методика, основанная на многомодальном распределении атрибутов исследуемой таблицы по критерию появления их в группе запросов на чтение информации. Для получения оптимального разбиения исследуемой таблицы с числом атрибутов, равным TS, необходимо выполнить следующие действия.

1. Получить для каждого атрибута значение частоты его появления в группе запросов к базе данных. Вектор частот появлений атрибутов исследуемой таблицы в группе запросов Q, где $FTA = \left\{ fta_{ifta} \middle| ifta = \overline{1,TS} \right\}$,

$$fta_{ifta} = \sum_{iq=1}^{nq} q_{iq}\left(SFQ\right) \cdot \left(q_{iq}\left(QA\right)\right)_{ifta},$$
$$iq = 1,...,nq.$$

2. Отсортировать атрибуты по частоте их появления в группе запросов:

$$FTA' = \left\{ fta'_{ifta} \middle| ifta = \overline{1,TS},\ fta' \in FTA,\ fta'_{ifta} \le fta'_{ifta+1} \right\}.$$

3. Сформировать группы атрибутов с одинаковой частотой. Получим разбиение конечного множества FTA'. $GF = \left\{ gf_1,...,gf_{bn} \right\}$, в котором

$$gf_1 \cup ... \cup gf_{bn} = FTA',\ gf_{bi} \neq \varnothing,\ 1 \le bi \le bn,$$

где $\forall ifta,\ fta'_{ifta} \in gf_{bi}$, если $\forall \left(gf_{bi}\right)_{bj} = fta'_{ifta},\ ifta = \overline{1,TS},\ bj = \overline{1,\left|gf_{bi}\right|}$.

4. Получить разбиение множества групп атрибутов. Получим разбиение конечного множества GF. $GF' = \left\{ gf'_1,...,gf'_{gn} \right\}$, в котором

$$gf'_1 \cup ... \cup gf'_{gn} = GF,\ gf'_{gi} \neq \varnothing,\ 1 \le gi \le gn,$$

где $\forall gf_{bi},\ gf_{bi} \in gf'_{gi},\ \dfrac{\left(gi-1\right) \cdot TS}{K} \le bi \le \dfrac{gi \cdot TS}{K},\ 1 \le bi \le bn.$ K – коэффициент, характеризующий количество разбиений множества групп атрибутов, $K \in \left[1, gn\right]$.

Варьируя параметр $K \in \left[1, gn\right]$, мы можем получать решения, эффективность которых оценивается при помощи выведенной в рамках исследования целевой функции.

Практическая апробация методики

Для анализа эффективности полученной методики были выделены исходные данные и базовые множества.

1. Параметр $TS = 16$, характеризующий количество столбцов в таблице:

№ п/п	Наименование столбца	Тип данных СУБД
1	Id	bigint
2	Attr1	nchar(10)
3	Attr2	nchar(10)
4	Attr3	nchar(10)
5	Attr4	nchar(10)
6	Attr5	nchar(10)
7	Attr6	nchar(10)
8	Attr7	nchar(10)
9	Attr8	nchar(10)
10	Attr9	nchar(10)
11	Attr10	nchar(10)
12	Attr11	nchar(10)
13	Attr12	nchar(10)
14	Attr13	nchar(10)
15	Attr14	nchar(10)
16	KeyForSearch	nchar(10)

2. Множество типов данных *DBT*, которые поддерживаются конкретной выбранной СУБД MS SQL 2012. Задано вектором, характеризующим занимаемое типом данных дисковое пространство в байтах, $DBT = \{4, 8, 20\}$ (количество типов данных СУБД уменьшено для компактности). Множество типов данных СУБД MS SQL 2012 может быть представлено в виде таблицы:

№ п/п	Наименование типа данных	Занимаемое дисковое пространство, байты
1	int	4
2	bigint	8
3	nchar(10)	20

3. Набор атрибутов (столбцов таблицы) *TA*:

	DBT_1	DBT_2	DBT_3
A_1	0	1	0
A_2	0	0	1
A_3	0	0	1
A_4	0	0	1
A_5	0	0	1
A_6	0	0	1
A_7	0	0	1
A_8	0	0	1
A_9	0	0	1
A_{10}	0	0	1
A_{11}	0	0	1
A_{12}	0	0	1

Окончание таблицы

	DBT_1	DBT_2	DBT_3
A_{13}	0	0	1
A_{14}	0	0	1
A_{15}	0	0	1
A_{16}	0	0	1

4. Множество, представляющее группу запросов Q на получение информации из базы данных, состоящее из 3 элементов:

№ п/п	Количество запросов, поступивших на сервер за выбранный период времени	Бинарный вектор атрибутов, участвующих в запросе
1	10	<0,0,1,1,0,0,0,0,0,0,0,0,0,0,1,1>
2	20	<1,1,1,1,1,1,0,0,1,1,0,1,1,1,0,1>
3	5	<1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1>

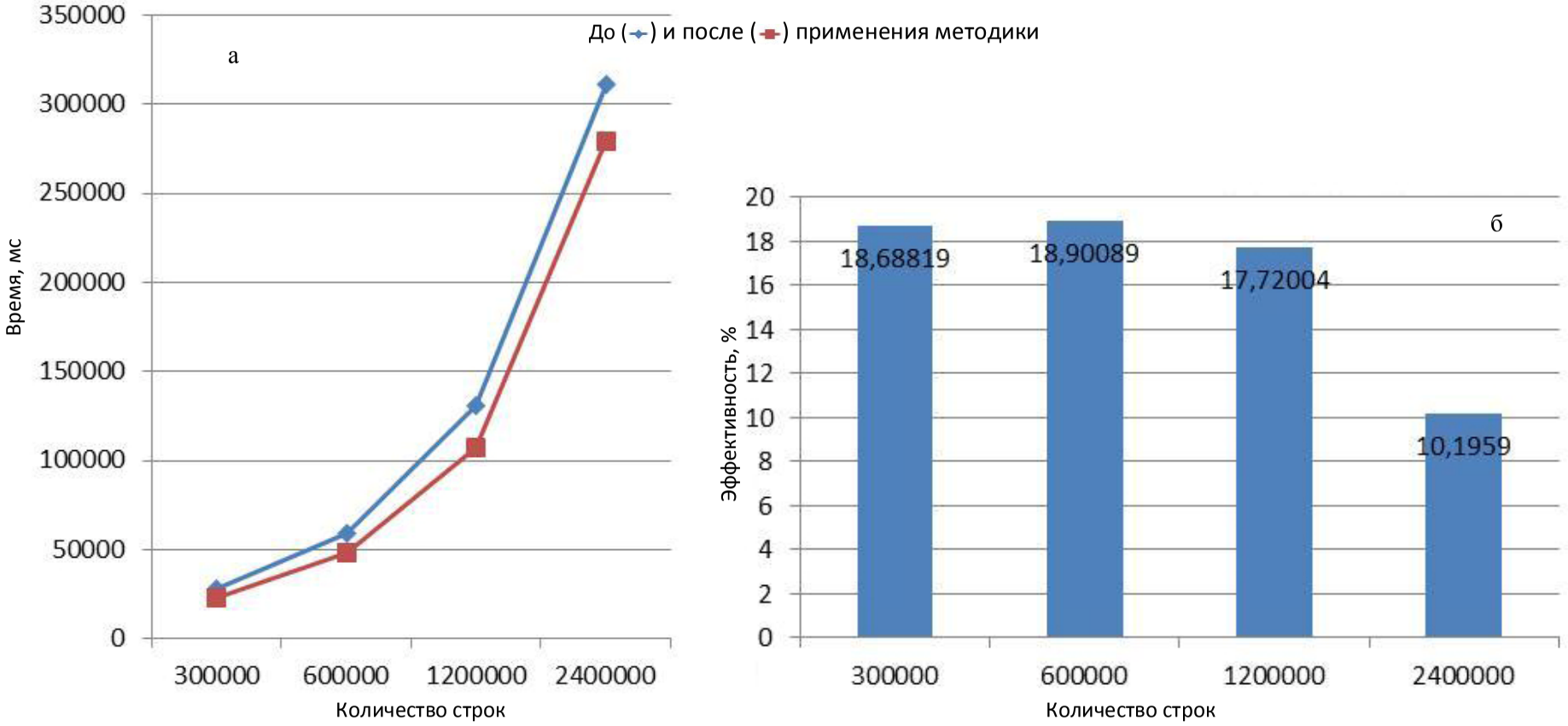
Для проведения эксперимента были исключены ограничения в виде индексов, триггеров и хранимых процедур, а также был исключен коэффициент дополнительного использования памяти. Это позволило продемонстрировать преимущества предлагаемой методики над традиционным подходом. В результате применения методики было предложено разделить исследуемую таблицу на четыре дочерних таблицы:

№	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
T1	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0
T2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
T3	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0	0
T4	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1

Полученное решение было проверено статистически на всей группе запросов. Для повышения достоверности экспериментов были выполнены все 35 запросов на чтение информации. Это позволило получить сведения о запросах, которые после применения методики стали выполняться быстрее, а также выделить подмножество медленных запросов. Результаты экспериментов, а также суммарное время выполнения группы запросов представлены в виде таблицы:

Количество строк в исследуемой таблице	Время выполнения группы запросов к исследуемой таблице, мс	Время выполнения группы запросов к таблицам после разделения на дочерние, мс
300 000	28 312	23 021
600 000	59 521	48 271
1 200 000	130 485	107 363
2 400 000	311 223	279 491

Эффективность применения методики представлена на рисунке.



Результаты применения методики в зависимости от количества строк:
а – время обработки группы запросов; *б* – эффективность, выраженная в процентах

Заключение

В результате проведенного исследования была сформулирована проблема повышения производительности информационной системы за счет реструктуризации табличных структур данных. Получено ее описание в теоретико-множественном представлении. Сформулированы целевая функция и ограничения. Предложен подход к нахождению субоптимального разбиения исследуемой табличной структуры на дочерние путем многомодального распределения атрибутов по частоте их появлений в запросах на чтение информации. Предложенная методика особенно актуальна для таблиц БД, которые используют небольшие информационные системы.

Полученные результаты могут быть использованы при проектировании отечественных СУБД. Дальнейшие исследования в этой области связаны с разработкой методик оптимальной реорганизации табличных структур данных для крупных информационных систем. Открытая апробация методики реализуется в виде веб-системы, в которой любой исследователь может ввести сведения о своей БД и получить рекомендуемое оптимальное разделение таблиц.

Список литературы

1. *Богданова А. В., Дьяченко Р. А., Бельченко И. В.* Повышение качества образовательного процесса за счет внедрения системы «Электронное расписание» в учебной организации // Политематический сетевой электронный научный журнал Кубанского государственного агарного университета. 2016. С. 873–885.

2. *Эмблер С. В., Садаладж П. Дж.* Рефакторинг баз данных: эволюционное проектирование: Пер. с англ. М.: ИД Вильямс, 2007. 672 с.

3. *Тоу Д.* Настройка SQL. Для профессионалов. СПб.: Питер, 2004. 333 с.

4. *Чигаркина Е. И.* Базы данных: Учеб. пособие. Самара: Изд-во СГАУ, 2015. 208 с.

5. *Atroshchenko V. A., BelchenkoV. E., Belchenko I. V., Dyachenko R. A.* Development and research of statistical methods and optimization algorithms of search for solutions in intelligence automated systems // International journal of pharmacy and technology. 2016. Vol. 8, no. 2. P. 14137–14149.

6. *Klir G. J.* Facets of Systems Science. Springer, 1991. 664 p.

Материал поступил в редколлегию 02.03.2018

I. V. Belchenko, **R. A. Diyachenko**

Kuban State Technological University
2 Moskovskaya Str., *Krasnodar*, *350072*, *Russian Federation*

ilur@mail.ru, *emessage@rambler.ru*

TECHNIQUES FOR IMPROVING PERFORMANCE OF THE SMALL INFORMATION SYSTEMS THROUGH OPTIMAL RESTRUCTURING DATA BASED ON MULTIMODAL DISTRIBUTIONS ATTRIBUTES

A systematic approach to increasing the productivity of small information systems is considered at the expense of optimal restructuring of tabular data structures. The authors formulated the task of optimizing the number of data blocks that are needed to query the group to read the information offered to the target function, and structural constraints. The impossibility of using crude methods of searching for the optimal solution is analyzed. The technique of multimodal attribute distribution is proposed depending on their frequency of occurrence in the query group. The experiment confirming the effectiveness of the developed methodology for small information systems.

Keywords: decision support system, optimization, data structures, databases, system analysis.

References

1. Bogdanova A. V., Diyachenko R. A., Belchenko I. V. Povyshenie kachestva obrazovatel'nogo protsessa za shchet vnedreniya sistemy «Elektronnoe raspisanie» v uchebnoj organizatsii. *Politematicheskij setevoj elektronnyj nauchnyj zhurnal Kubanskogo gosudarstvennogo agarnogo universiteta*, 2016, p. 873–885. (in Russ.)

2. Embler S. V., Sadaladzh P. Dzh. Refaktoring baz dannykh: evolyutsionnoe proektirovanie. Transl. from Engl. Moscow, Vilyams Publ., 2007, 672 p. (in Russ.)

3. Tou D. Nastrojka SQL. Dlya professionalov. St. Petersburg, Piter, 2004, 333 p. (in Russ.)

4. Chigarkina E. I. Bazy dannykh. Samara, SGAU Press, 2015, 208 p. (in Russ.)

5. Atroshchenko V. A., Belchenko V. E., Belchenko I. V., Diyachenko R. A. Development and research of statistical methods and optimization algorithms of search for solutions in intelligence automated systems. *International journal of pharmacy and technology*, 2016, vol. 8, no. 2, p. 14137–14149.

6. Klir G. J. Facets of Systems Science. Springer, 1991, 664 p.

For citation:

Belchenko I. V., Diyachenko R. A. Techniques for Improving Performance of the Small Information Systems through Optimal Restructuring Data Based on Multimodal Distributions Attributes. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 19–30. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-19-30

УДК 004.942
DOI 10.25205/1818-7900-2018-16-2-31-40

И. В. Брак [1], **Ю. И. Сазонова** [2, 3]

[1] *НИИ физиологии и фундаментальной медицины*
ул. Тимакова, *4*, *Новосибирск*, *630090*, *Россия*

[2] *Институт вычислительных технологий*
пр. Академика Лаврентьева, *6*, *Новосибирск*, *630090*, *Россия*

[3] *Новосибирский государственный университет*
ул. Пирогова, *1*, *Новосибирск*, *630090*, *Россия*

brack@physiol.ru, *i.sazonova@g.nsu.ru*

РАЗРАБОТКА СЕРВИСА ЗАДАНИЯ СЦЕНАРИЕВ ПРЕДЪЯВЛЕНИЯ СТИМУЛОВ С ИСПОЛЬЗОВАНИЕМ МОДЕЛЬНО-ОРИЕНТИРОВАННОГО ПОДХОДА

Современная физиология не может обойтись без методов количественного анализа данных. Необходимым условием для использования математической статистики, анализа сигналов и машинного обучения является наличие должным образом собранных, размеченных и подготовленных данных. С возможностью совместной обработки данных, собранных в разных условиях и в рамках разных протоколов экспериментов, появилась потребность в наличии структурированной метаинформации. В настоящее время существует множество программных систем, позволяющих создавать, редактировать и запускать сценарии представления стимулов. Их проблемой является сложность использования реализованного сценария как в рамках других систем, так и для аннотирования данных, полученных экспериментально. Целью работы является разработка сервиса, позволяющего задавать сценарии представления стимулов с помощью графического интерфейса с возможностью сохранять метаинформацию эксперимента в независимом от платформы формате и исполнять в закрытых системах. В предлагаемом решении используется модельно-ориентированный подход. В основе платформенно-независимой модели лежит открытый формат эксперимента PsychoPy. Для исполнения полученного сценария используется платформа Neurobs Presentation. С помощью преобразования общей модели сценария эксперимента в модель платформы и описания синтаксической структуры предметно-ориентированного языка Presentation автоматически формируется программный код. Реализация данного подхода может быть расширена для других систем представления стимулов.

Ключевые слова: модельно-ориентированный подход, кодогенерация, предметно-ориентированный язык, система предъявления стимулов.

Введение

Количественный анализ данных является важным методом исследований в современной инструментальной физиологии. Высокой прогностической и диагностической значимостью обладают данные биоэлектрической активности головного мозга. Анализ данных количественной электроэнцефалографии (кЭЭГ) является перспективным направлением для применения математической обработки. В кЭЭГ используются такие параметры, как амплитуда, мощность, спектр, когерентность внутри- и межполушарных взаимодействий и другие характеристики осцилляторной активности головного мозга [1]. Регистрация показателей может происходить в состоянии спокойного бодрствования с открытыми или закрытыми глазами, при выполнении функциональных проб или когнитивных заданий. При создании эксперимента необходимо учитывать такие аспекты, как содержание и структура сценария, взаимодействие и синхронизация с аппаратными компонентами, формат выходных данных.

Брак И. В., *Сазонова Ю. И.* Разработка сервиса задания сценариев предъявления стимулов с использованием модельно-ориентированного подхода // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 31–40.

В настоящее время существует ряд программных систем, позволяющих создавать, редактировать и воспроизводить сценарии предъявления стимулов (Cedrus – SuperLab [1], Millisecond Software – Inquisit [2], Mitsar – Psytask [3], Neurobs – Presentation [4], Nottingham University – PsychoPy [5], OkazoLab – EventIDE [6], Psychology Software Tools – E-Prime [7] и др.). Большинство систем платные и позволяют сохранять сценарий либо в виде программы на собственном языке создания сценариев, либо в закрытом формате. Системы, в которых способом создания сценария является написание программы, обладают широкими возможностями исполнения, но сложны для изучения и применения. Проблемой платных систем также является ограничение возможностей использования реализованного сценария как в рамках других систем, так и для аннотирования данных, полученных экспериментально.

Целью работы является разработка сервиса, позволяющего задавать сценарии предъявления стимулов с помощью графического интерфейса с возможностью сохранять метаинформацию о сценарии эксперимента в независимом от платформы формате и исполнять в системах с собственным языком описания сценариев.

В предлагаемом решении используется модельно-ориентированный подход. В основе платформенно-независимой модели лежит открытый формат эксперимента системы PsychoPy. Для исполнения полученного сценария была выбрана платформа Neurobs Presentation. С помощью преобразования платформенно-независимой модели в модель языка Presentation и описания его синтаксической структуры автоматически формируется программный код. Реализация данного подхода может быть расширена для исполнения разработанного сценария в других системах предъявления стимулов с собственным скриптовым языком.

Анализ систем предъявления стимулов

Рассмотрим особенности создания, редактирования и сохранения данных экспериментов для систем задания сценариев предъявления стимулов (табл. 1).

Создание сценария эксперимента может происходить двумя способами: с помощью собственного «скриптового» языка системы (Presentation [8], Inquisit [9]) или с помощью графического интерфейса, функциональность которого иногда дополняют языком общего назначения или его расширением.

Язык Inquisit похож на язык разметки и является декларативным. Он представляет собой набор именованных элементов (стимулы, тестовые пробы, последовательности стимулов и т. п.) и выражений присваивания для их параметров. Кроме констант, списков и ссылок на другие элементы, значениями параметров могут быть выражения (арифметические, присваивающие и условные). Таким образом, потенциальная сложность языка может заключаться в описаниях параметров.

В языке Presentation выделяются декларативная (Scenario Description Language, SDL) и процедурная (Presentation Control Language, PCL) части. Кроме того, в заголовке скрипта Presentation задаются настройки сценария (Header). Декларативная часть языка позволяет описать набор элементов эксперимента и их параметров. В процедурной части есть возможность задать порядок предъявления элементов SDL, используя общие конструкции: переменные, контейнеры, условия и циклы.

Язык для описания сценариев Psytask [10] включает в себя списки стимулов и проб, список предъявления проб и команд, обработку ответной реакции. Набор стимулов ограничен

[1] https://www.cedrus.com/superlab/
[2] https://www.millisecond.com
[3] http://www.mitsar-medical.com/eeg-software/qeeg-software/
[4] https://www.neurobs.com
[5] http://www.psychopy.org
[6] http://www.okazolab.com
[7] https://pstnet.com/products/e-prime/
[8] https://www.neurobs.com/presentation/docs/index_html
[9] https://millisecond.com/support/docs/
[10] http://www.mitsar-eeg.ru/download/manuals/Psytask_UM_RUS_v.1.50.pdf

Таблица 1

Обзор форматов, используемых основными системами предъявления стимулов

Система	Лицензия	Способ задания сценария	Формат	
			сценария	выходных данных
E-Prime	Комм.	Графический интерфейс E-Studio, дополнительная функциональность реализуется с помощью языка E-Basic (на основе Visual Basic)	***.es** – формат для хранения и редактирования в E-Studio; ***.ebs** – формат для исполнения в E-Run	***.edat**
Inquisit	Комм.	Редактор для языка Inquisit	***.iqx –** текстовый файл скрипта	***.tsv**
EventIDE	Комм.	Графический интерфейс (есть возможность использования XAML для графических элементов), дополнительная функциональность реализуется с помощью расширения языка C#	***.eve** – формат EventIDE	**Delimiter separated values**
Presentation	Комм.	Редактор языка Presentation и графический интерфейс для дополнительных настроек	***.sce** – текстовый файл скрипта	**Delimiter separated values**
PsychoPy	О.	Графический интерфейс PsychoPy Builder / редактор Python PsychoPy Coder	***.psyexp –** файл формата XML, соответствующий XSD эксперимента / ***.py** – файл программы на Python	**Delimiter separated values** / ***.psydat** (сериализованный Python объект)
Psytask	Пр.	Графический интерфейс / загрузка файла сценария	***.PRO** – текстовый файл скрипта	***.dbf** (запись в базу данных)
SuperLab	Комм.	Графический интерфейс	***.sl** – формат SuperLab	***.tsv**

Примечание. О системе SuperLab см.: https://www.cedrus.com/superlab/manual.htm/. Обозначения лицензии: Комм. – коммерческая, Пр. – проприетарная, О. – открытая.

Таблица 2

Особенности систем предъявления стимулов с графическим интерфейсом

Система	Графическое представление последовательности предъявления	«Полнота» графического интерфейса	Универсальность системы
E-Prime	+	–	+
EventIDE	+	–	+
PsychoPy	+	+	+
Psytask	–	+	–
SuperLab	–	+	+

несколькими форматами, порядок показа линейный, без возможности рандомизации, за счет чего язык понятен и хорошо подходит для создания простых сценариев. Более универсальные предметно-ориентированные языки (Inquisit, Presentation) включают в себя понятия разных уровней: от уровня предметной области (функциональная проба) до особенностей реализации (цвет шрифта). Вместе с особенностями порядка предъявления стимулов (рандомизацией) и обработкой реакции исследуемого такие языки будут достаточно сложными для людей, не знакомых с программированием.

При создании сценариев с помощью графического интерфейса существующие системы используют формы для задания параметров. Кроме того, некоторые системы используют визуальное представление последовательности стимулов в виде потока работ (PsychoPy [11]), ориентированного графа (EventIDE [12]) или древовидной структуры (E-Prime [13]). Системы E-Prime и EventIDE расширяют функциональность графического интерфейса с помощью расширений для языков общего назначения Visual Basic и C#. Особенности графических систем предъявления стимулов показаны в табл. 2. Под «полнотой» понимается возможность задания любого реализуемого в системе сценария с помощью графического интерфейса (без использования языка программирования). Универсальными названы системы со встроенной рандомизацией и возможностью проектировать нелинейные сценарии**.**

Сценарий в платных системах сохраняется в закрытом формате и может исполняться только внутри системы. В случае, когда сценарий является скриптом, его можно редактировать как текст. Форматы выходных данных чаще всего представляют собой набор значений (delimiter separated values) и хорошо подходят только для анализа показателей в рамках одного исследования.

Исполнение реализованного в определенной системе сценария невозможно без ручного переноса информации в другую систему. Различные аспекты сценария (оформление стимулов, настройки последовательности и времени предъявления, аппаратные особенности, формат вывода и пр.) в большинстве случаев собраны вместе, затрудняя тем самым изучение и применение системы для создания и изменения сценариев. Кроме того, полезная для дальнейшей обработки данных метаинформация о сценарии эксперимента не может быть напрямую получена из закрытого формата эксперимента и из скрипта сценария.

Для решения задачи платформенно-независимой разработки сценариев с использованием возможностей существующих систем предъявления стимулов (графический интерфейс и функциональность) и обеспечением модульной интеграции между ними наиболее целесообразно применить модельно-ориентированный подход [2; 3].

Применение модельно-ориентированного подхода к задаче разработки сценариев

С «архитектурой, управляемой моделями» (Model Driven Architecture, MDA [14]), связывают стандарт MDA, разрабатываемый консорциумом Object Management Group [15] с 2000 г. Согласно методологии MDA модели являются главными элементами процесса разработки. Для конструирования программного приложения должна быть построена подробная, формально точная модель, из которой потом может быть автоматически получен исполняемый программный код. Под моделью понимается выборочное (ограниченное) представление некоторой системы, форма и содержание которого могут быть выражены с помощью набора понятий (концептов). Для описания модели могут быть использованы различные нотации и форматы. Метамодель определяет абстрактный синтаксис языка моделирования.

По стандарту «Метаобъектного средства» (Meta-Object Facility, MOF [16]) различают четыре уровня моделирования: M0–M3. Языком описания верхнего уровня часто является Unified Modeling Language (UML [17]).

[11] http://www.psychopy.org/builder/builder.html#builder
[12] http://www.okazolab.com
[13] https://support.pstnet.com/hc/en-us/categories/115000291167-E-Prime-3-x
[14] https://www.omg.org/mda/
[15] https://www.omg.org
[16] https://www.omg.org/spec/MOF

В процессе модельно-ориентированной разработки можно выделить следующие шаги [4]:

1) создание модели предметной области, независимой от платформы (Platform Independent Model, PIM);

2) создание модели платформы (Platform Specific Model, PSM), которая определяет специфику конкретной реализации;

3) преобразование PIM → PSM, с помощью которой с каждым формальным понятием модели предметной области сопоставляется его реализация;

4) генерирование необходимых артефактов.

К каждому из шагов 1–3 можно возвращаться, расширяя модель. При этом изменения уже существующих элементов повлекут за собой изменения на следующих стадиях.

В настоящее время существуют инструменты для применения стандарта OMG MDA [18]. В работе использовалась свободно распространяемая система Eclipse Modeling Framework (EMF) [5]. Проект EMF представляет собой платформу для моделирования с возможностью генерирования программного кода для создания инструментов и приложений на основе структурированной модели данных. В качестве модели верхнего уровня в EMF используется язык Ecore [6], который похож на UML, но формально не является расширением.

Рассмотрим задачу разработки сценариев с точки зрения модельно-ориентированного подхода.

На рис. 1 показаны разные уровни моделирования для платформенно-независимых (PIM) и платформенно-зависимых (PSM) моделей сценария эксперимента. В качестве модели верхнего уровня в обоих случаях выступает Ecore. В терминах Ecore может быть описан физиологический эксперимент и эксперимент, реализованный в системе Presentation, которая выступает в качестве платформы. На уровне модели могут находиться сценарии конкретных экспериментов (например, «GoNoGo Task»). В общей модели экземпляром эксперимента может быть его представление в виде потока работ, а в платформенно-зависимой – скрипт языка Presentation.

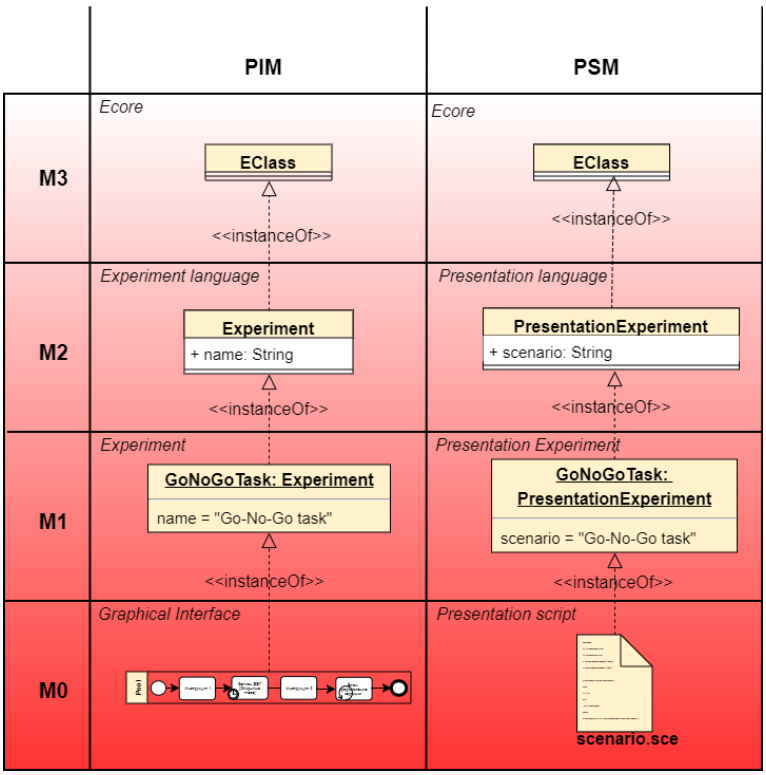


Рис. 1. Уровни моделирования задачи разработки сценария эксперимента

[17] https://www.omg.org/spec/UML/

[18] http://mda-directory.omg.org/vendor/list.htm

В качестве основы для платформенно-независимой модели была использована XML схема эксперимента системы PsychoPy [19]. В соответствии со схемой сохраняются получаемые с помощью PsychoPy Builder [20] эксперименты, при этом она достаточно общая и не содержит информации о деталях реализации. С помощью инструмента EMF была получена Ecore модель, соответствующая схеме на рис. 2.

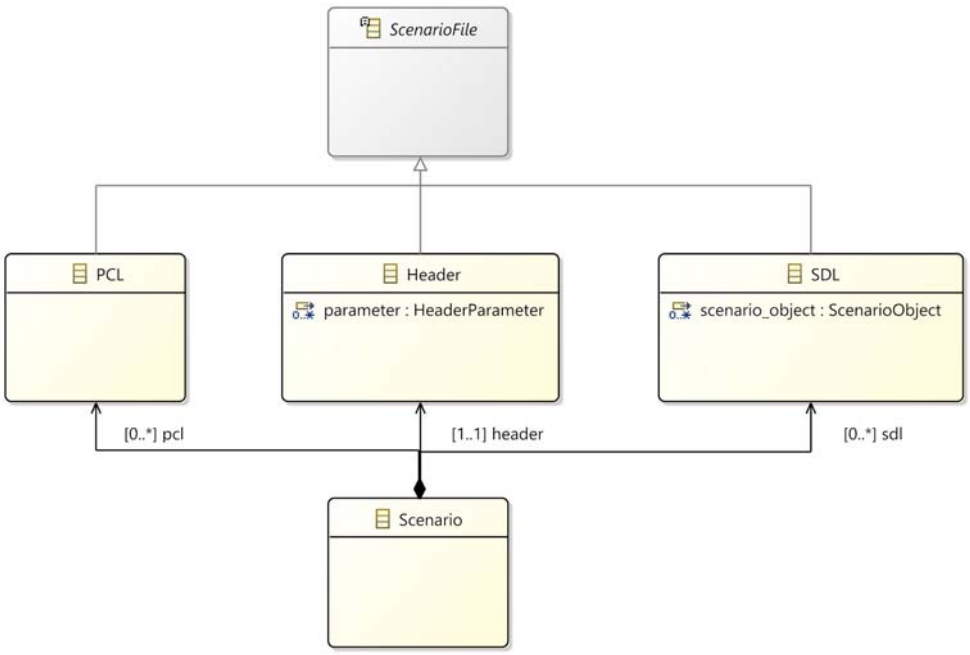


Рис. 2. Описания платформенно-зависимой модели с помощью Ecore (файл сценария)

Платформенно-зависимая модель языка Presentation создавалась с помощью редактора Ecore моделей. Она содержит понятия языка, соответствующие его синтаксическим конструкциям. На рис. 2. показаны концепты моделей верхнего уровня для описания структуры скрипта, выраженных с помощью Ecore.

Работу системы можно представить в виде преобразований моделей (рис. 3):

1) получение экземпляра сценария (1);
2) представление экземпляра сценария в виде экземпляра Ecore модели (1 → 2);
3) преобразование платформенно-независимой модели к модели платформы Presentation;
4) генерирование скрипта для исполнения на Presentation (3 → 4).

Получение экземпляра сценария (1) происходит с помощью графического редактора PsychoPy. Далее, файл сценария в формате XML преобразуется к экземпляру обобщенной модели в формате Ecore с помощью сгенерированного Java кода.

Преобразование 2 → 3 экземпляра обобщенной модели эксперимента в экземпляр модели эксперимента Presentation происходит на уровне M1. Для этого задается трансформация для моделей уровня M2. В предлагаемой реализации используется язык преобразования моделей Epsilon [7]. Он выбран в силу того, что позволяет задавать трансформации Ecore-моделей и может запускаться в качестве независимого программного модуля. Трансформация представляет собой набор правил, описывающих соответствие элементов обобщенной модели элементам платформы Presentation. Запуск преобразования начинается с корня XML-документа и рекурсивно вызывается для вложенных элементов с помощью механизма «ленивого правила» [8].

[19] http://www.psychopy.org/psyexp.html
[20] http://www.psychopy.org/builder/builder.html

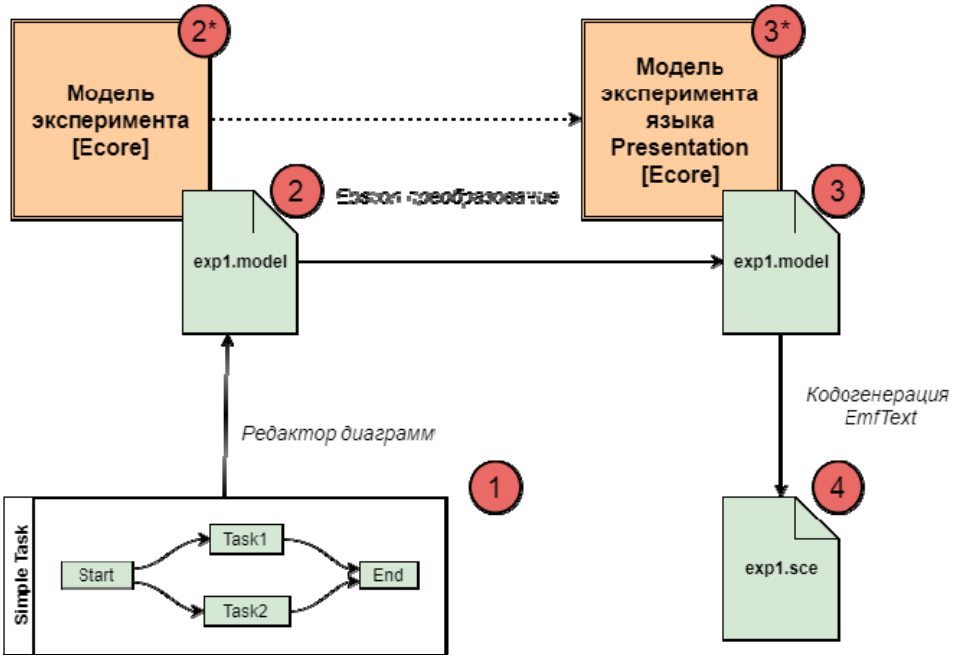


Рис. 3. Преобразования моделей системы

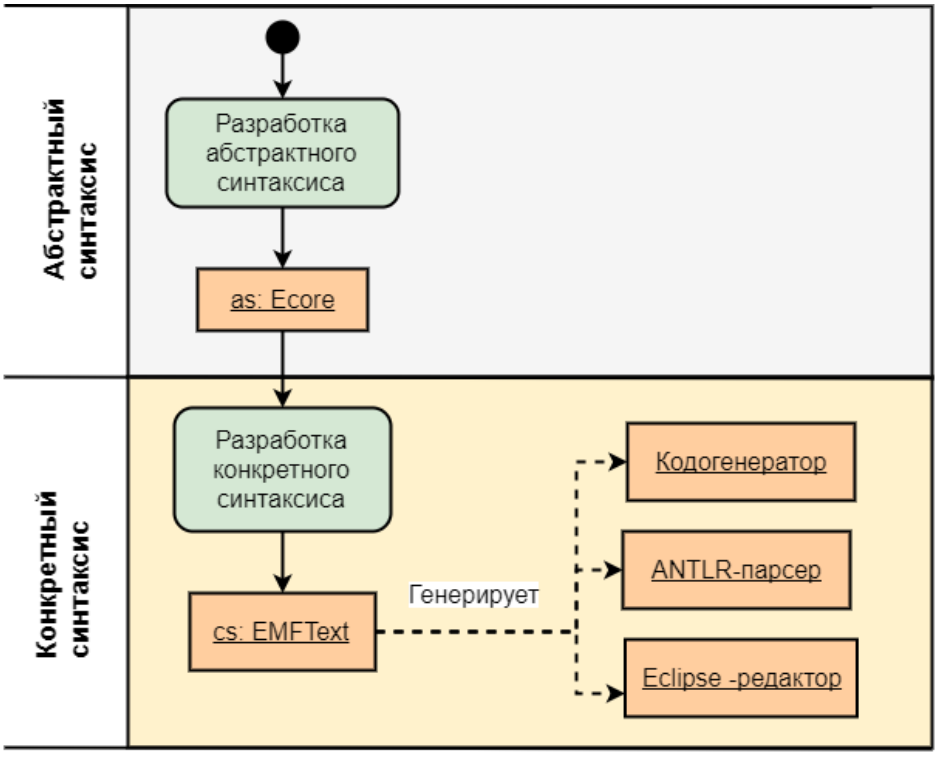


Рис. 4. Процесс разработки в EMFText

Получение исходного кода происходит с помощью инструмента EMFText [9]. На основе модели абстрактного синтаксиса и описания конкретного синтаксиса он позволяет получить ANTLR [21]-парсер, кодогенератор и редактор для предметно-ориентированного языка (рис. 4). Модель абстрактного синтаксиса представляется в формате Ecore. Модель конкретного син-

[21] http://www.antlr.org

таксиса выражается с помощью файла в формате *.cs, содержащего описание токенов, стилей подсветки для редактора и правил грамматики в форме Бэкуса-Науэра, где в качестве служебных символов используются элементы модели абстрактного синтаксиса и их атрибуты.

Сервис представляет собой консольное приложение, написанное на языке Java и позволяющее запустить цепочку преобразований исходного экземпляра сценария эксперимента, полученного с помощью интерфейса PsychoPy, в текст скрипта для исполнения на Presentation.

Заключение и перспективы развития

На основе модельно-ориентированного подхода разработан сервис задания сценариев предъявления стимулов для физиологических экспериментов. Показана работоспособность данного подхода: с помощью преобразования модели эксперимента в модель платформы и описания синтаксической структуры предметно-ориентированного языка Presentation генерируется программный код. Полученный в результате запуска экземпляр модели сценария эксперимента может быть использован в качестве структурированного источника метаданных о сценарии эксперимента. Таким образом, была осуществлена интеграция графического интерфейса PsychoPy и функциональных возможностей Presentation с помощью модельно-ориентированного подхода.

Использованный подход позволил организовать взаимодействие между различными системами за счет информации, выраженной в платформенно-независимой модели. Установлено, что для его применения к системам необходима информация о структуре сценария и стабильность данной структуры для разных версий. Такие условия выполняются для систем с предметно-ориентированным языком создания сценариев. Применение подхода к системам с закрытым форматом не представляется возможным без получения дополнительной информации.

В дальнейшем планируется:

1) расширение подхода для других систем предъявления стимулов, позволяющих задавать сценарии с помощью предметно-ориентированного языка;

2) реализация обратной трансформации: получение экземпляров независимой от платформы модели на основе написанных скриптов Presentation, для чего нужно реализовать обратную трансформацию PSM → PIM;

3) добавление нового уровня моделирования для типовых функциональных проб (закрытые и открытые глаза, классические протоколы).

Список литературы

1. *Кропотов Ю. Д.* Количественная ЭЭГ, когнитивные вызванные потенциалы мозга человека и нейротерапия. Донецк, 2010.

2. *Логвинова К. В.* Современные технологии и средства разработки программного обеспечения // Бизнес-информатика. 2007. № 2.

3. *Paige R. F. et al.* User Experience for Model-Driven Engineering: Challenges and Future Directions // ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems. Institute of Electrical and Electronics Engineers Inc., 2017.

4. *Kleppe A. G., Warmer J. B., Bast W.* MDA explained: the model driven architecture: practice and promise. Addison-Wesley Professional, 2003.

5. *Сорокин А. В., Кознов Д. В.* Обзор Eclipse Modeling Project // Системное программирование. 2010. Т. 5, № 1.

6. *Steinberg D. et al.* EMF: eclipse modeling framework. Pearson Education, 2008.

7. *Kolovos D. S., Paige R. F., Polack F. A. C.* The epsilon transformation language // International Conference on Theory and Practice of Model Transformations. Springer, Berlin, Heidelberg, 2008. P. 46–60.

8. *Kolovos D. et al.* The epsilon book // Structure. 2010. Vol. 178. P. 1–10.

9. *Heidenreich F. et al.* Model-based language engineering with EMFText // Generative and Transformational Techniques in Software Engineering IV. Springer, Berlin, Heidelberg, 2013. P. 322–345.

Материал поступил в редколлегию 20.04.2018

I. V. Brak [1], **Yu. I. Sazonova** [2, 3]

[1] *State Scientific Research Institute of Physiology & Basic Medicine*
4 Timakov Str., Novosibirsk, 630090, Russian Federation

[2] *Institute of Computational Technologies SB RAS*
6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation

[3] *Novosibirsk State University*
1 Pirogov Str., Novosibirsk, 630090, Russian Federation

brack@physiol.ru, i.sazonova@g.nsu.ru

DEVELOPMENT OF THE SERVICE FOR STIMULI SCENARIO REPRESENTATION BASED ON MODEL DRIVEN ARCHITECTURE

Methods of quantitative data analysis are important in modern physiology. Necessary condition for usage of mathematical statistics, signal analysis and machine learning is the availability of properly collected, marked and prepared data. Thus, preservation of meta-information and structuring results will be useful for their further processing. Physiological experiment consists of a set of trials (samples), in which instructions and certain stimuli are presented to the participant. Reaction on the test sample is recorded as physiological measures. Currently there are many software systems that allow you to create, edit and present scenarios of stimuli representation. Existing systems of presentation stimulus scenario can solve a wide range of tasks but they are not suitable for reusing and there is no universal way to extract metadata of the scenario of the experiment. Purpose of the work is development of the service for stimuli scenario representation with graphical interface, features of saving data in platform independent format and execution in one of the systems. Proposed approach uses model driven architecture principles. The platform-independent model is based on the open format of PsychoPy experiment. Neurobs Presentation system is used to execute scenario. Program code is generated automatically with transformation of the platform-independent model into platform-specific model and describing the syntax of the Presentation domain specific language. Implementation of this approach may be extended to other systems.

Keywords: model driven architecture, code generation, domain specific language, system of stimuli representation

References

1. Kropotov J. D. Quantitative EEG, event-related potentials and neurotherapy. Donetsk, 2010. (in Russ.)

2. Logvinova K. V. Modern technologies and tools for software development. *Business informatics*, 2007, no. 2. (in Russ.)

3. Paige R. F. et al. User Experience for Model-Driven Engineering: Challenges and Future Directions. *ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems*. Institute of Electrical and Electronics Engineers Inc., 2017.

4. Kleppe A. G., Warmer J. B., Bast W. MDA explained: the model driven architecture: practice and promise. Addison-Wesley Professional, 2003.

5. Sorokin A., Koznov D. Review of the Eclipse Modeling Project. *System Programming*, 2010, vol. 5, no. 1. (in Russ.)

6. Steinberg D. et al. EMF: eclipse modeling framework. Pearson Education, 2008.

7. Kolovos D. S., Paige R. F., Polack F. A. C. The epsilon transformation language. *International Conference on Theory and Practice of Model Transformations*. Springer, Berlin, Heidelberg, 2008, p. 46–60.

8. Kolovos D. et al. The epsilon book. *Structure*, 2010, vol. 178, p. 1–10.

9. Heidenreich F. et al. Model-based language engineering with EMFText. *Generative and Transformational Techniques in Software Engineering IV*. Springer, Berlin, Heidelberg, 2013, p. 322–345.

For citation:

Brak I. V., Sazonova Yu. I. Development of the Service for Stimuli Scenario Representation Based on Model Driven Architecture. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 31–40. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-31-40

УДК 004.896
DOI 10.25205/1818-7900-2018-16-2-41-48

И. Е. Букшев

Новосибирский государственный университет
ул. Пирогова, 1, Новосибирск, 630090, Россия

Центр финансовых технологий
ул. Шатурская, 2, Новосибирск, 630055, Россия

bukshev@gmail.com

MEDILUX – СЕРВИС ИНТЕЛЛЕКТУАЛЬНОГО ФОРМИРОВАНИЯ РАСПИСАНИЯ ПОСЕЩЕНИЙ МЕДИЦИНСКИХ УЧРЕЖДЕНИЙ

Предложен новый вид обслуживания – «интеллектуальная регистратура»: в мобильном приложении обеспечена возможность выбора сотрудников медицинского учреждения, далее происходит анализ графика работы выбранного врача и свободного времени пользователя при помощи алгоритмов программирования в ограничениях с целью определения наилучшего времени записи на прием. База знаний о свободном времени пользователя формируется на основе задач, которые агрегируются с мобильного устройства и популярных сервисов по управлению задачами. Для случая, если пользователь не знает, к кому обратиться, продукт снабжен «умным» чатом, в котором можно описать проблему. Текст отправится на сервер, где произойдет синтаксический разбор и семантическое сопоставление с конкретной специальностью врача. В базе данных хранится информация обо всех посещениях и врачебных выписках (электронная медицинская карта), что позволяет, например, напоминать пользователю о необходимости принятия медикаментов.

Практическая ценность продукта заключается в автоматизации бизнес-процесса «прием пациентов», что приводит к экономии времени пациентов, обеспечению высокой доступности услуг и оптимизации трудовых затрат в медучреждениях.

Ключевые слова: регистратура, программирование, мобильное, приложение, чат, запись, прием.

Актуальность

В настоящее время проблема качества медицинского обслуживания стала предметом внимания властей и средств массовой информации, поскольку уровень развития здравоохранения говорит о развитии государства в целом. Бизнес-процесс «прием пациентов» является отличным примером, так как пока еще посещение поликлиники требует много усилий и связано с негативными последствиями, такими как:

1) жалобы пациентов из-за недоступности медицинской помощи;
2) перекрестное инфицирование из-за скопления пациентов в одном месте;
3) самолечение и, как следствие, увеличение материальных затрат на лечение пациентов с осложнениями;
4) периодические стрессовые ситуации, связанные с работой, у персонала медицинских учреждений;
5) падение рейтинга и имиджа, что является существенной проблемой для негосударственных медицинских учреждений.

Анализ возможных источников проблемы дал понять, что к вопросу следует подходить с нескольких сторон. Возможные факторы и причины:

Букшев И. Е. Medilux – сервис интеллектуального формирования расписания посещений медицинских учреждений // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 41–48.

1) кадровая проблема – отсутствие должного количества врачей;
2) нежелание врачей работать в государственных медицинских учреждениях;
3) отсутствие должного взаимодействия между подразделениями медицинской организации и четких алгоритмов работы каждого подразделения;
4) ограничение информированности населения о возможности получения медицинской помощи, несвоевременные обращения, необходимость для пациентов в «лишних посещениях» (запись, чтобы «узнать» или «спросить»);
5) избыточные временные затраты на оформление медицинских документов, дублирование информации, растущая отчетность.

Исходя из перечисленных проблем можно сделать вывод, что не существует универсального решения даже для такого, казалось бы, малого звена в функционировании медицинского учреждения, как бизнес-процесс «прием пациентов».

Предлагаемое решение

Частично помочь пациентам может «клиент-серверное» решение, где в качестве «клиента» будет выступать мобильное приложение, предоставляющее пользователям электронную медицинскую карту, электронную регистратуру и электронный чат с врачами. Данное решение не ново, но предлагаемый сервис отличается от прочих тем, что способен *автоматически определять* удобные дату и время приема как для конечного пользователя, так и для лечащего врача. Помимо этого, в сервисе предлагается интеллектуальный чат, в котором пользователь сможет получить общую информацию или советы в режиме реального времени.

Таким образом, сервис нацелен на анализ повседневного ритма и рабочего графика с последующей возможностью *интеллектуального планирования* и *автоматического осуществления записи* в медицинские учреждения. Мобильное приложение будет выступать интеллектуальным звеном-помощником между пациентами и поликлиниками.

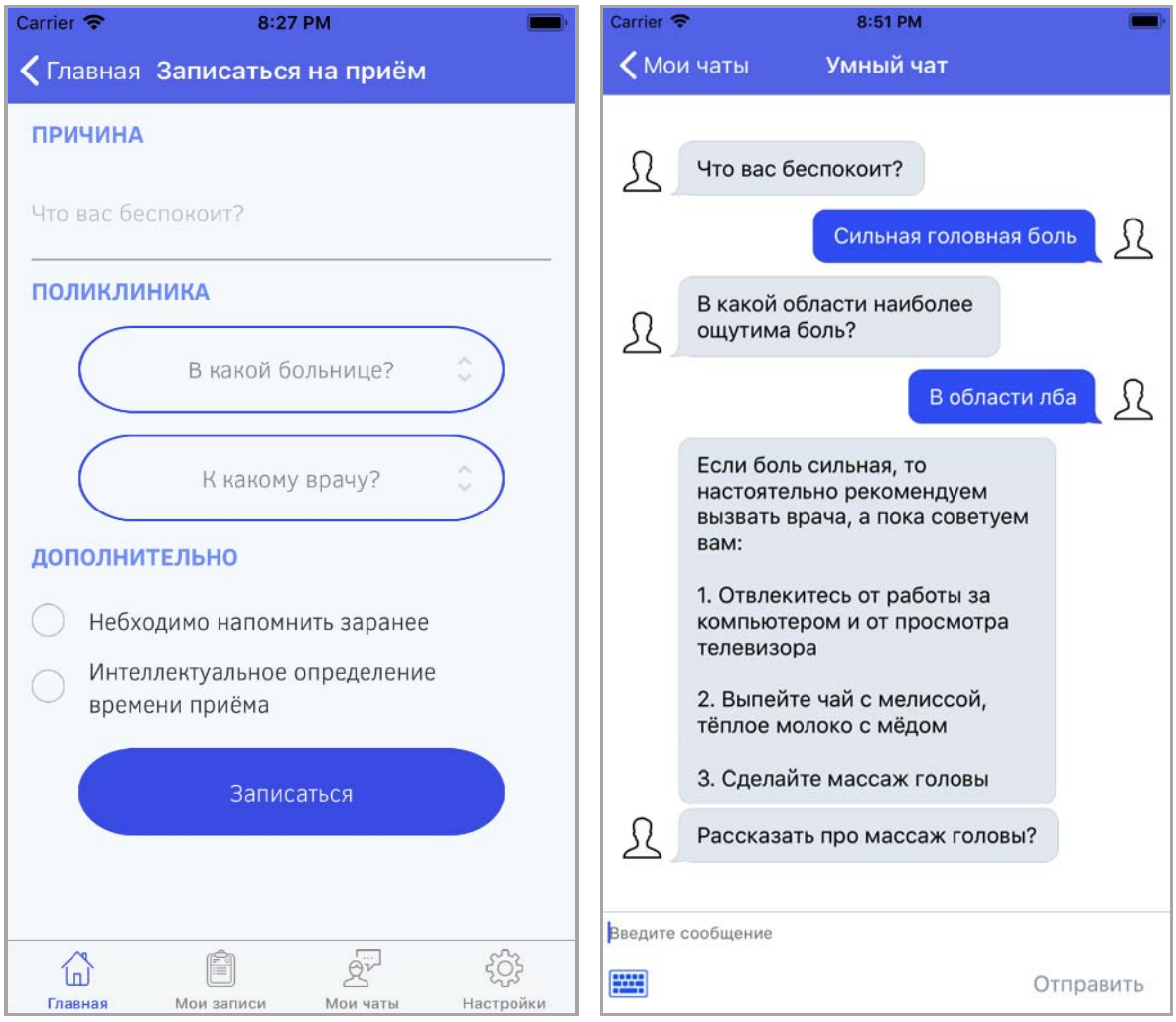
Научная новизна

В работе предложен новый вид обслуживания – «интеллектуальный прием пациентов»: в мобильном приложении имеется возможность выбора сотрудников медицинского учреждения, далее происходит анализ графика работы выбранного врача и свободного времени пользователя при помощи алгоритмов «программирования в ограничениях» с целью определения наилучшего времени записи на прием. База знаний о свободном времени пользователя формируется на основе задач, которые агрегируются с мобильного устройства и популярных сервисов по управлению задачами (Trello, Google Tasks, Wunderlist, Apple Reminders и др.). Для случая, если пользователь не знает, к кому обратиться, продукт снабжен «умным» чатом, в котором можно описать проблему. Текст отправится на сервер, где произойдет синтаксический разбор и семантическое сопоставление с конкретной специальностью врача. В базе данных хранится информация обо всех посещениях и врачебных выписках (электронная медицинская карта), что позволяет, к примеру, напоминать пользователю о необходимости принять медикаменты.

Практическая ценность продукта заключается в *автоматизации* бизнес-процесса «прием пациентов», что приводит к *экономии* времени пациентов, обеспечению *высокой доступности* услуг и *оптимизации* трудовых затрат в медицинских учреждениях.

Технические детали

Клиентская часть выполнена нативными средствами на языке Swift 4.0 с использованием среды разработки Xcode. Основной архитектурой для всех модулей приложения взят VIPER (View-Interactor-Presenter-Entity-Router). Соблюдены принципы SOLID [1] и SOA, что позволяет в дальнейшем без трудностей расширять функциональные возможности приложения [2]. Дизайн выполнен с соблюдением Human Interface Guidelines от компании «Apple», что предоставляет пользователям привычный UX-дизайн (рис. 1).



а б

Рис. 1. Экран «Запись на прием» (*а*) и «Умный чат» (*б*)

Серверная часть реализована на языке Java 8 с использованием среды разработки IntelliJ IDEA. За архитектурную основу взят стиль REST (Representational state transfer) – в этой архитектуре данные передаются без дополнительных слоев, что делает ее менее ресурсоемкой в сравнении с SOAP или XML-RPC, здесь не нужно анализировать запрос, чтобы понять его природу и транслировать данные из одного формата в другой.

В качестве базы данных выбран PostgreSQL 10.

Математическая модель

Очевидно, что возможных решений (вариантов для приема) может быть несколько, и здесь мы имеем дело с комбинаторной задачей, которую для удобства можно записать в виде задачи удовлетворения ограничений (Constraint satisfaction problem, CSP) [3; 4]. В нашем случае CSP включает в себя:

1) variables – переменные;
2) domains – набор возможных значений;
3) constraints – список ограничений.

Правил всего два: для каждой переменной мы задаем набор возможных значений – переменным могут быть присвоены любые значения, а не только 1 или 0 («истина» или «ложь») [5]; а также у нас есть список ограничений, которым удовлетворяют исходные переменные.

Для нас решением задачи удовлетворения ограничений будет нахождение всех возможных значений, которые могут принимать переменные, с учетом существующих ограничений [6].

Термин «время для приема» следует разбить и рассматривать данную модель, как совокупность «дня» и конкретного «времени».

Выделим шесть переменных:

D – множество возможных значений времени, когда лечащий врач может принять пациента;

U – множество возможных дней для пациента;

C – множество общих значений времени, когда запись неосуществима (например, ночное время);

P – множество значений времени, когда пользователь не может посетить врача (множество строится на основе значений времени, указанных в заметках);

A – множество дополнительных значений времени, которое определяется на основании рабочего графика, предпочтительного времени и других подобных факторов, выбранных пользователем в мобильном приложении;

O – множество вычисляемых переменных-ограничений (например, если в заметках указана геолокация мест, то мы можем вычислить время, за которое пользователь доберется от текущего местоположения до цели, тем самым мы получим дополнительные ограничения).

Зададим ограничения, которые будем накладывать на переменные, чтобы получить актуальное множество решений:

$$D = U,\ D \neq C,\ U \neq C,$$
$$U \neq P,\ U \neq A,\ U \neq O.$$

«Программирование в ограничениях» подразумевает собой использование декларативной формы программирования, что в отличие от императивного стиля позволяет нам в разы упростить задачу: нам нужно лишь описать проблему в общем случае, а всеми вычислениями и поиском решений будет заниматься так называемый решатель (Solver), который и содержит эффективные алгоритмы вычислений. К сожалению, исчерпывающих обзоров теории «удовлетворения ограничений» на русском языке нет, однако имеются публикации, которые освещают отдельные аспекты данной предметной области [8; 9].

После того как решатель вычислил множество значений времени, которые удовлетворяют нашим условиям – *мягким* и *глобальным* ограничениям [7], мы можем показать это множество пользователю, после чего осуществить запись на прием в медицинское учреждение.

Схема реализации

Рассмотрим одну из основных функций приложения – запись на прием (рис. 2). С помощью функции «запись на прием» пользователь попадает на экран, на котором расположены:

1) текстовое поле «что вас беспокоит?» – именно здесь пользователь может описать причину обращения, указать симптомы или другую информацию для врача;

2) поле выбора «в какой больнице?» – предоставляется список доступных медицинских учреждений, в которые интегрирован данный сервис;

3) поле выбора «к какому врачу?» – список доступных врачей для выбранной поликлиники;

4) поле «напомнить заранее» – при выбранном значении пользователь будет уведомлен о предстоящем визите заранее, посредством push-нотификации на мобильное устройство или почту;

5) поле «интеллектуальное определение времени приема» – при невыбранном значении пользователь может самостоятельно связаться с медицинским учреждением и согласовать время приема.

Если все-таки пациент выбрал «интеллектуальное определение времени», то на сервер отправляются задачи, которые были интегрированы с мобильного устройства и различных сервисов для планирования, а также отправляется множество выбранных пользователем «пред-

почтительных дат для приема». На «сервере» происходит сопоставление информации, полученной с «клиента», с графиком выбранного врача. Далее, на том же сервере, решатель (Solver) находит решение задачи программирования в ограничениях и отправляет их обратно «клиенту», где пользователь может выбрать наиболее подходящий вариант. После выбора информация вновь отправляется на «сервер», откуда попадает в медицинское учреждение, уведомляя тем самым врача о новой записи.

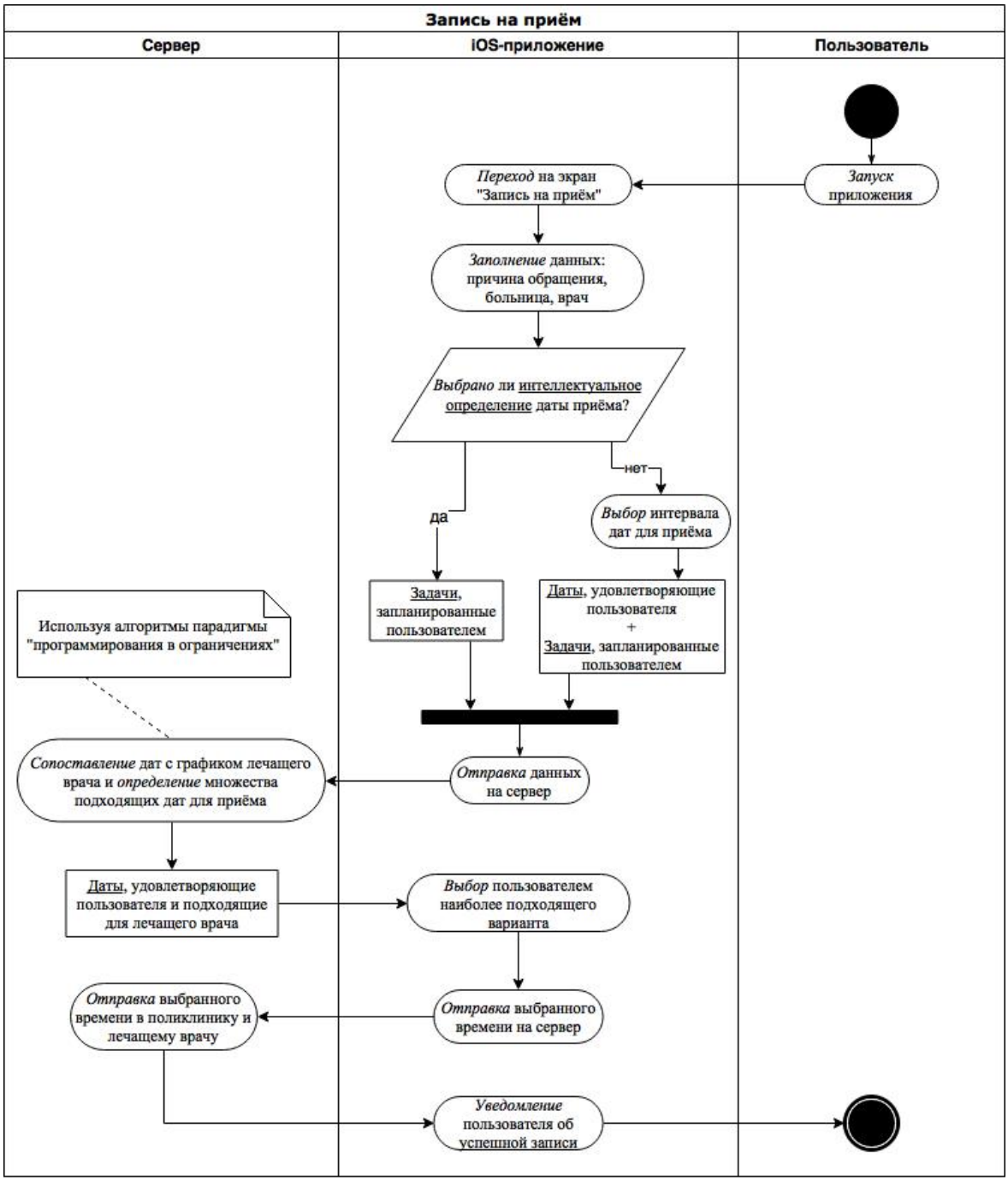


Рис. 2. Алгоритм истории «Запись на прием»

История с интеллектуальным чатом представляет собой творческую задачу. На данном этапе реализован простой синтаксический разбор текста, с последующим разбиением на лексемы, которые сопоставляются с семантическими командами (рис. 3).

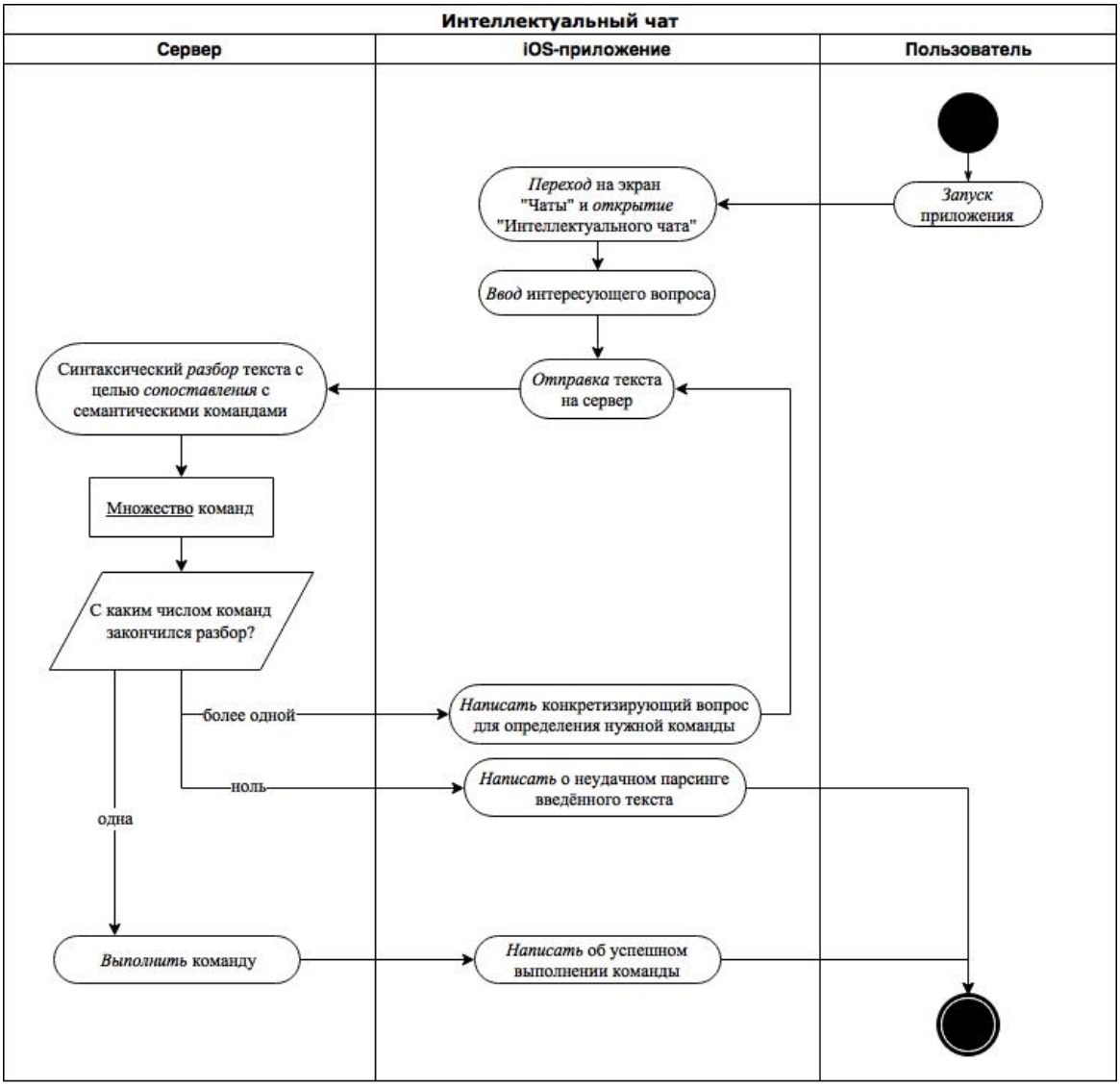


Рис. 3. Алгоритм работы интеллектуального чата

Заключение

Подводя итоги, можно сказать, что готовый продукт действительно является интеллектуальным помощником и промежуточным звеном между пациентами и медицинскими учреждениями. Теперь запись на прием становится простой и доступной для абсолютного большинства людей. Разработанный сервис особенно удобен для людей, у которых рабочий день не нормирован – им присущ плотный график с частыми форс-мажорными ситуациями, из-за чего данный сегмент зачастую пользуется сервисами-планировщиками, которые, в свою очередь, можно интегрировать с разработанным сервисом Medilux, что приведет к более точному определению наилучшего времени записи на прием.

Помимо прочего, интеллектуальный чат в приложении способствует сокращению обращений в поликлиники с целью быстрых вопросов или консультаций – теперь можно получить ответы (которые согласованы со специалистами) на большинство вопросов, не выходя из дома.

Электронная медицинская карта с полной историей записей избавляет от бумажной волокиты и позволяет напоминать пользователю о необходимости принять медицинские препараты через push-нотификации на мобильное устройство.

Сервис Medilux готов составить конкуренцию уже существующим сервисам в данной области. Разработанный продукт отличается уникальными особенностями, которые имеют векторы для развития, что приведет к более точным результатам определения времени записи на прием и расширению «словарного запаса» интеллектуального чата.

Список литературы

1. *Макконел С.* Совершенный код. Мастер-класс: Пер. с англ. М.: ИТД «Русская редакция»; СПб.: Питер, 2005. 896 с.: ил.

2. *Холл Г. М.* Адаптивный код: гибкое кодирование с помощью паттернов проектирования и принципов SOLID: Пер. с англ. 2-е изд. СПб.: Альфа-книга, 2017. 448 с.: ил.

3. *Dechter R.* Constraint processing. San Francisco: Morgan Kaufmann, 2003. 481 p.

4. *Tsang E.* Foundations of Constraint Satisfaction. New York: Academic Press, 1993. 421 p.

5. *Dechter R., Frost D.* Backtracking algorithms for constraint satisfaction problems. University of California, 1999. URL: https://www.ics.uci.edu/~csp/r56-backtracking.pdf (дата обращения 12.02.2018).

6. *Beek P. van.* Reasoning about qualitative temporal information // Arti cial Intelligence. 1992. Vol. 58. P. 297–326.

7. *Rossi F., Beek P. van, Walsh T.* Chapter 4 Constraint Programming in Foundations of Arti cial Intelligence // Handbook of Knowledge Representation. Eds. F. van Harmelen, V. Lifschitz, B. Porter. 2008. Vol. 3. P. 181–211.

8. *Ушаков Д. М., Телерман В. В.* Системы программирования в ограничениях (обзор) // Системная информатика: Сб. науч. тр. Новосибирск: Наука, 2000. Вып. 7: Проблемы теории и методологии создания параллельных и распределенных систем. С. 275–310.

9. *Щербина О. А.* Удовлетворение ограничений и программирование в ограничениях. URL: http://www.intsys.msu.ru/magazine/archive/v15(1-4)/shcherbina-053-170.pdf (дата обращения 23.03.2018).

Материал поступил в редколлегию 24.03.2018

I. E. Bukshev

Novosibirsk State University
1 Pirogov Str., Novosibirsk, 630090, Russian Federation

Center of Financial Technologies
2 Shaturskaya Str., Novosibirsk, 630055, Russian Federation

bukshev@gmail.com

MEDILUX – SERVICE OF INTELLECTUAL FORMING THE SCHEDULE OF VISITING MEDICAL INSTITUTIONS

In this paper, a new type of service «intellectual registry» is offered. In the mobile application, there is a choice of employees of the medical institution. In addition to that, there is an analysis of the schedule of the selected doctor and free time of the patient using programming algorithms in limitations to determine the best time to book an appointment.

The knowledge base of the user's free time is formed based on tasks that are aggregated from the mobile device and popular task management services. In case the user does not know who to contact, the product is equipped with a «smart» chat where the problem can be described. To be exact, the text will be sent to the server where there will be a parsing and a semantic comparison with the specific qualification of the doctor.

The database stores information about all visits and medical statements (electronic medical records), which allows to remind the user about the need to take the medications.

The practical value of the product lies in the automation of the business process «reception of patients», which leads to saving patients' time, ensuring high availability of services and optimizing labor input in medical institutions.

Keywords: registry, intellectual, constraint, programming, Swift, iOS, Java.

References

1. McConnell S. Code Complete. 2nd ed. ISBN 0-7356-1967-0.
2. Hall G. M. Adaptive Code. 2nd ed. ISBN 978-1-5093-0258-1.
3. Dechter R. Constraint processing. San Francisco: Morgan Kaufmann, 2003, 481 p.
4. Tsang E. Foundations of Constraint Satisfaction. New York: Academic Press, 1993, 421 p.
5. Dechter R., Frost D. Backtracking algorithms for constraint satisfaction problems. University of California, Department of Information and Computer Science, 1999.
6. Beek P. van. Reasoning about qualitative temporal information. *Arti cial Intelligence*, 1992, vol. 58, p. 297–326.
7. Rossi F., Beek P. van, Walsh T. Chapter 4 Constraint Programming in Foundations of Arti cial Intelligence. *Handbook of Knowledge Representation*. Eds. F. van Harmelen, V. Lifschitz, B. Porter. 2008, vol. 3, p. 181–211.
8. Ushakov D. M., Telerman V. V. Sistemy programmirovaniya v ogranicheniyah (obzor) [Systems of programming in constraints (overview)]. Novosibirsk, Nauka, 2000, vol. 7, p. 275–310. (in Russ.)
9. Shcherbina O. A. Udovletvorenie ogranicheniy i programmirovanie v ogranicheniyakh [Satisfaction of constraints and constraint programming]. URL: http://www.intsys.msu.ru/magazine/archive/v15(1-4)/shcherbina-053-170.pdf (last access 23.03.2018). (in Russ.)

For citation:

Bukshev I. E. Medilux – Service of Intellectual Forming the Schedule of Visiting Medical Institutions. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 41–48. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-41-48

УДК 004.91, 004.514
DOI 10.25205/1818-7900-2018-16-2-49-61

В. В. Исаченко [1], **З. В. Апанович** [1, 2]

[1] *Новосибирский государственный университет*
ул. Пирогова, 1, Новосибирск, 630090, Россия

[2] *Институт систем информатики им. А. П. Ершова СО РАН*
пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия

vv.isachenko@gmail.com, apanovich@iis.nsk.su

СИСТЕМА АНАЛИЗА И ВИЗУАЛИЗАЦИИ ДЛЯ КРОСС-ЯЗЫКОВОЙ ИДЕНТИФИКАЦИИ АВТОРОВ НАУЧНЫХ ПУБЛИКАЦИЙ

Представлена система разрешения неоднозначности авторства статей на английском языке с использованием русскоязычных источников данных. Система позволяет находить и исправлять ошибки в определении авторства научных публикаций, что может улучшить результаты поиска статей определенного автора и подсчета индекса цитируемости.

В качестве исходного хранилища публикаций использовалась база link.springer.com, для получения достоверной информации об авторах и их статьях использовалась научная электронная библиотека eLIBRARY.ru.

Система предоставляет интерактивную визуализацию результатов и возможность редактирования для повышения качества экспертного анализа. Подходы, используемые в данной системе, применимы для разрешения неоднозначности авторства публикаций из различных библиографических баз данных.

Ключевые слова: разрешение неоднозначности авторства, кросс-языковая идентификация сущностей, обработка естественного языка, интерактивная визуализация, кластеризация.

Введение

Многие научные цифровые библиотеки, такие как DBLP, PubMed, Springer и др., предоставляют функции, которые облегчают исследования целых коллекций документов. Такие системы дают доступ к миллионам библиографических записей, и на данный момент являются важнейшим источником информации для академического сообщества, так как они позволяют производить централизованный поиск публикаций.

Одной из проблем, возникающих при поиске публикаций определенного автора, является то, что такие системы не свободны от ошибок идентификации авторов. Эти ошибки могут быть двух типов: публикации двух разных персон присваиваются одной персоне или публикации одной персоны распределяются по нескольким разным персонам.

От подобного рода ошибок не свободно большинство библиографических систем, в том числе VIAF, SCOPUS и др. Например, на сайте Scopus представлено пять авторов с разными вариантами написания фамилии Непомнящий. При этом публикациям реальной персоны – В. А. Непомнящий, сотрудник ИСИ СО РАН – соответствовали публикации четырех из них, и все они имели различные идентификаторы. Помимо того, что данные ошибки затрудняют поиск статей, относящихся к определенному автору, они могут влиять на такую важную характеристику работы ученых, как индексы цитируемости. Причин возникновения ошибок

Исаченко В. В., *Апанович З. В.* Система анализа и визуализации для кросс-языковой идентификации авторов научных публикаций // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 49–61.

достаточно много: множественные варианты транслитерации с русского языка на английский, ошибки автоматических систем по наполнению библиографических баз данных, невнимательность пользователей.

Наиболее точно решает проблему установления авторства публикаций экспертный анализ. Эксперты могут идентифицировать автора неизвестного документа или определить принадлежность произведения другому автору при помощи характерных языковых особенностей, стиля автора. Однако экспертный анализ – трудоемкий процесс, поэтому разрабатываются системы для автоматизации определения авторства документов. В таких системах применяются подходы из теории распознавания образов, математической статистики и теории вероятностей, алгоритмы нейронных сетей, кластерного анализа и др.

К сожалению, автоматическое разрешение неоднозначности не дает стопроцентной точности, и в любом случае требуется вмешательство эксперта. Задача эксперта усложняется тем, что количество документов в коллекции, для которой необходим анализ, может достигать нескольких сотен или даже тысяч. Для упрощения восприятия результатов анализа применяется интерактивная визуализация информации в виде графов, матриц смежности, диаграмм и т. п. Такое представление коллекции документов и полученных результатов значительно ускоряет процесс экспертного анализа.

Задача, в которой все публикации даны на одном языке (например, на английском) достаточно хорошо изучена: существуют решения, которые работают в условия неполноты и разнородности данных и показывают высокую точность результатов [1; 2]. Однако задача кросс-языковой идентификации сущностей (в частности, данных на английском и русском языках) является достаточно новой и требует детального изучения.

В работах [3; 4] описаны эксперименты по кросс-языковой идентификации сущностей при помощи Открытого архива СО РАН на основе исчерпывающей информации о местах работы авторов. Хотя результаты были достаточно обнадеживающими, основной проблемой был локальный характер этого архива, поскольку он касался только сотрудников СО РАН. Таким образом, возник вопрос, с каким более крупным русскоязычным источником можно провести подобные эксперименты. В качестве такого экспериментального источника данных была выбрана научная электронная библиотека eLIBRARY.ru [1], которая содержит большое количество подтвержденных записей о публикациях российских ученых.

В данной статье описана система анализа и визуализации публикаций на естественном языке для автоматизации процесса устранения неоднозначности авторства научных публикаций (рис. 1). Система производит идентификацию авторов коллекции статей на основании извлекаемых метаданных и текста публикации, а также предоставляет интерактивную визуализацию для упрощения интерпретации полученных результатов и анализа коллекции.

Постановка задачи

В качестве исходного хранилища публикаций использовалась база link.springer.com [2]. По фамилии, имени и отчеству (ФИО) на русском языке из данного хранилища извлекается коллекция статей на английском языке. Часть данных о статье, в том числе текст публикации, может отсутствовать. В качестве источника достоверной информации об авторах и их статьях использовалась Научная электронная библиотека (eLIBRARY.ru). Бóльшая часть информации в ней представлена на русском языке.

Для решения проблемы идентификации авторства коллекции документов из хранилища link.springer.com требуется:

- сопоставить статьи из хранилища link.springer.com со статьями из eLIBRARY.ru;
- произвести разделение набора научных статей, соответствующих одному или нескольким авторам, на набор непересекающихся множеств, где каждому множеству соответствует один автор данных научных публикаций;
- произвести визуализацию полученных результатов.

[1] eLIBRARY.RU – Научная электронная библиотека. URL: https://elibrary.ru/.
[2] Springer – International Publisher Science, Technology, Medicine. URL: https://link.springer.com/.

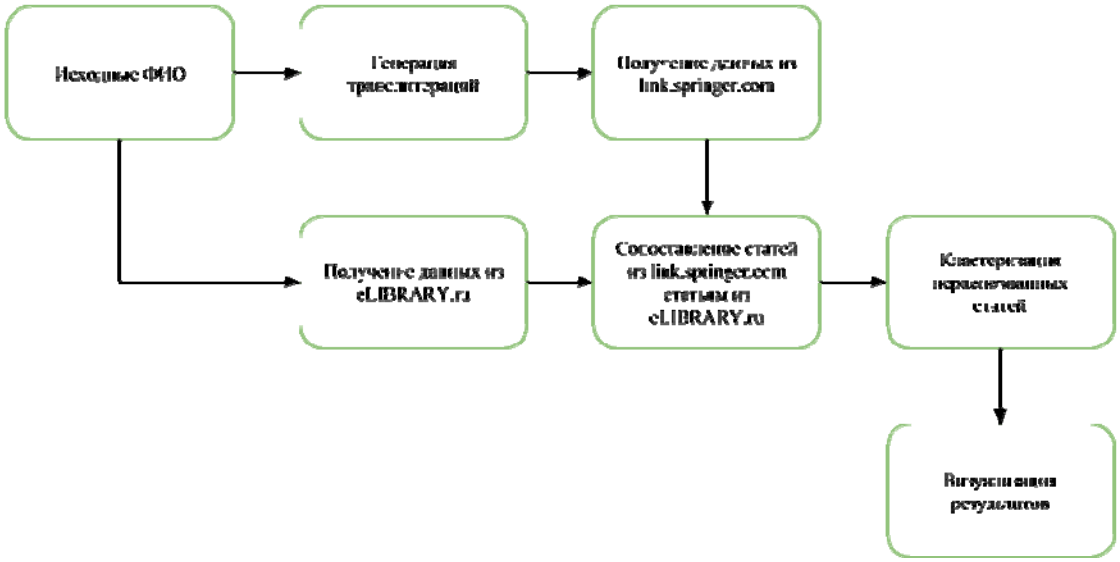


Рис. 1. Схема работы системы

Генерация транслитераций

Как известно, существует проблема неоднозначности транслитерации имен авторов с русского языка на английский язык. Как упоминалось ранее, в системе Scopus хранятся публикации В. А. Непомнящего, отнесенные к разным авторам, имена которых представлены как Nepomniaschy, V.A. Nepomnyashchii, V.A. Nepomnyaschy. Генерация всех возможных транслитераций не представляется возможной, так как реальные данные могут не подчиняться правилам транслитерации букв. Однако чем больше будет покрытие вариантов, тем больше данных будет доступно при поиске. В ранних работах использовались транслитерации, полученные лишь по одному из имеющихся стандартов или по обращению на языковые ресурсы для переводов текста, такие как translate.google.com [3]. Данные подходы покрывают малое количество вариантов транслитерации.

В текущей работе были изучены различные транслитерации букв русского алфавита, используемые в стандартах зарубежных стран и Российской Федерации (ГОСТ 7.79-2000, ГОСТ 16876-71 и пр.) [5], а также транслитерации, используемые в обиходе пользователей сети Интернет [4].

На основании выделенных транслитераций отдельных букв была реализована генерация всех возможных транслитераций имени автора на русском языке. С каждым именем сопоставляются различные варианты сокращений, так как не всегда в хранилище возможно найти статьи по полному имени автора. По всем вариантам транслитерации производится обращение к базе данных Springer. Также осуществляется обращение к eLIBRARY.ru по изначально предложенным экспертом ФИО на русском языке.

Идентификация авторства статей
путем их сопоставления с данными из eLIBRARY.ru

В результате получения входных данных исходными параметрами статей из хранилища link.springer.com являются:

- название статьи;
- список авторов;
- список мест работы для каждого автора;

[3] Google Переводчик. URL: https://translate.google.com/.
[4] ALA-LC Romanization Table for Russian. URL: http://www.loc.gov/catdir/cpso/romanization/russian.pdf

- дата публикации;
- название журнала;
- список тем, затронутых в публикации;
- список ключевых слов;
- текст публикации в формате pdf.

В результате получения входных данных исходными параметрами статей из eLIBRARY.ru являются:

- название статьи;
- список авторов;
- информация об издании, в котором была опубликована статья.

Сопоставление происходит по доступным параметрам следующим образом:

- пусть A – статья из link.springer.com, B – статья из eLIBRARY.ru;
- если название статьи A совпадает с названием статьи B полностью, без учета разделительных символов, регистра и знаков препинания, то считается, что $A = B$;
- иначе производится стемминг названий A и B, и считается коэффициент совпадения названия как доля совпадающих слов в данных названиях;
- коэффициент соавторства данных статей принимается равным доле совпадающих авторов;
- если сумма двух данных коэффициентов превышает пороговое значение, то считается, что $A = B$.

Если название публикации и список авторов указаны в eLIBRARY на русском языке, сравнить их вышеуказанным способом не получится. В таком случае в сравнении использовались результаты машинного перевода названия и списка авторов с русского языка на английский. В качестве инструмента для машинного перевода использовалась система Яндекс.Переводчик.

Иногда среди данных о публикации из link.springer.com доступна информация об издании, в котором был опубликован оригинал статьи на русском языке, включающая в себя номер выпуска, номера страниц и дату публикации. В таком случае производится сравнение этой информации с данными eLIBRARY. В результате такого сопоставления формируются группы статей, которые принадлежат одному автору, найденному в электронной библиотеке eLIBRARY.ru, а также группа статей, которые не были распознаны.

Для оценки качества сопоставления проведены эксперименты на данных сотрудников ИСИ СО РАН. В выборку были включены 25 сотрудников института, чьи публикации содержатся в системе link.springer.com. Средний процент числа публикаций авторов, распознанных системой, составил 79 %, при этом количество публикаций, которые не принадлежат автору, но были отнесены в его группу, близко к нулю. Основной причиной, по которой система не может определить принадлежность статьи ее автору, является неполнота данных. Для улучшения результатов к группе статей, которые не были распознаны, применяется алгоритм подсчета близости и группировки статей, описанный далее.

Алгоритм кластеризации статей

Алгоритм кластеризации статей, не сгруппированных на ранних этапах, заключается в попарном сравнении статей и объединении групп в случае, если коэффициент схожести статей превышает заданный порог. Более формальное описание алгоритма приведено ниже.

Пусть $A = \bigcup_{g_i} A_{g_i}$ – множество статей, полученных после сопоставления публикаций из Springer с публикациями из eLIBRARY.ru, где g_i – номер группы. При этом группа A_{g_i}, где $g_i = -1$ – группа публикаций, для которых не было найдено сопоставление. Тогда применяется следующий алгоритм:

для каждой статьи $s \in A$
    для каждой статьи $t \in A$
        $d := $ *коэффициент сходства* (s,t)
        Если $(d > threshold)$
            Если $(Group(s) = -1 \ \ и \ \ Group(t) = -1)$
                $NewGroup(s,t)$
            Иначе
                $UniteGroups(s,t)$

При объединении групп происходит проверка на то, что обе эти группы не были изначально сформированы на этапе сопоставления со статьями из eLIBRARY.ru. В данном случае объединения не происходит, так как эти группы соответствуют статьям различных авторов, указанным в eLIBRARY.ru.

Асимптотика данного алгоритма $O(N^3)$. Для улучшения данной асимптотики была применена структура данных «Система непересекающихся множеств» [6]. С ее помощью асимптотика операции объединения групп уменьшается до $O(1)$, следовательно, весь алгоритм имеет асимптотику $O(N^2)$.

Подсчет коэффициента сходства статей

Для подсчета близости научных статей из хранилища link.springer.com используются все полученные через API данные, чтобы сократить влияние неполноты данных на результаты идентификации. Сравнение каждого из параметров формирует свой коэффициент, который суммируется в итоговый.

Далее, пусть A и B – различные статьи, полученные из хранилища link.springer.com.

Сравнение названий статей:

- если название статьи A совпадает с названием статьи B полностью, без учета разделительных символов, регистра и знаков препинания, то считается, что коэффициент совпадения названий равен максимальному значению – 1.0;
- иначе производится стемминг названий A и B, и считается коэффициент совпадения названий как доля совпадающих слов в данных названиях.

Сравнение списков авторов публикаций. Коэффициент соавторства статей принимается равным доле совпадающих авторов.

Для *сравнения имен авторов* производятся следующие шаги:

- приведение пары имен к одинаковому формату (например, если одно имя является полным, а во втором отсутствует отчество автора, то из первого удаляется отчество; таким же образом обрабатывается ситуация с сокращениями имен);
- производится сравнение имен с помощью алгоритма сравнения строк.

По результату сравнения двух приведенных к одному формату имен авторов не всегда можно сразу сказать, являются ли эти строки ФИО одного и того же человека. Это обусловлено тем, что транслитерации имени одного человека могут достаточно сильно различаться, либо, наоборот, люди могут являться полными тезками. Для того чтобы уменьшить количество ошибок при сравнении, используется полученная информация о местах работы авторов. В случае если место работы совпадает, коэффициент сравнения имен авторов увеличивается, так как более вероятно, что это один и тот же человек.

Сравнение и формирование коэффициентов схожести тем и ключевых слов статей подсчитывается аналогично коэффициенту соавторства, т. е. они принимаются равными доле совпадающих терминов.

Сравнение даты публикаций. Данный коэффициент является небольшим добавочным коэффициентом и призван улучшить сопоставление документов в соответствии с гипотезой о том, что если между датами публикаций прошло не очень много времени, то вероятность

того, что они принадлежат одному автору выше, чем у тех документов, которые были приняты в печать с довольно продолжительным разрывом во времени:

- если между датами публикаций статьи *A* и статьи *B* разница менее 5 лет, то коэффициент принимается равным 0.1;
- иначе, если между датами публикаций статьи *A* и статьи *B* разница более 25 лет, то коэффициент принимается равным –0.1.

Еще один добавочный коэффициент, основанный на эвристике, – это *сравнение названия журнала*: если названия журналов статей *A* и *B* совпадают, то коэффициент принимается равным 0.1.

Подсчет коэффициента сходства текста публикаций

Для подсчета коэффициента сходства текстов на естественном языке они представляются в виде векторов в многомерном пространстве. Тогда мера близости между ними определяется как косинусное расстояние. Для улучшения качества сравнения текстов на естественном языке, а также уменьшения размерности векторного представления текстов производится их предобработка [7], в которую входят удаление стоп-слов и стемминг.

Для построения векторного представления текстов в ранних работах использовался алгоритм мешка слов (bag of words) с применением TF-IDF меры [8]. TF-IDF – статистическая мера, показывающая важность слова в контексте набора документов. Наибольший показатель будет иметь слово, которое часто встречается в документе, но редко встречается во всей коллекции.

Также были проведены эксперименты по векторизации текстов на естественном языке с применением инструмента word2vec [5]. Это программный инструмент анализа семантики естественных языков, представляющий собой технологию, которая основана на дистрибутивной семантике и векторном представлении слов. Векторное представление слов основывается на контекстной близости: близкие векторы будут иметь слова, имеющие похожий смысл. Векторные репрезентации слов, полученные в результате работы word2vec, обладают следующим свойством: смысл имеют только расстояния между векторами, а не сами векторы. При сложении векторов двух слов получается вектор слова, который показывает нечто общее между исходными. Однако увеличение количества слагаемых быстро приводит к потере какого-либо ценного результата, поэтому нельзя описать основную идею документа простой суммой векторов всех слов, которыми представлен текст.

Одним из вариантов векторного представления текста является представление, в котором каждый элемент соответствует некоторой тематике. Перечислив достаточное количество возможных тематик текста, можно посчитать количество слов в тексте, соответствующих каждой тематике, и получить семантический вектор текста – вектор, каждый элемент которого обозначает отношение данного текста к той или иной тематике.

Таким образом, для построения семантического вектора текста необходимо описать достаточное количество кластеров, отражающих тематику и стиль текста. С помощью алгоритма кластеризации все слова разбиваются на заданное число кластеров, и, если количество кластеров будет достаточно большим, можно ожидать, что каждый кластер будет указывать на достаточно узкую тематику текста, а точнее, на узкий признак тематики или стиля.

Каждое слово имеет отношение ко многим кластерам – к каким-то больше, к каким-то меньше. Поэтому вычисляется семантический вектор слова – вектор, зависящий от расстояния от слова до центра соответствующего кластера в полученном векторном пространстве. После этого, для того чтобы получить семантический вектор текста, необходимо сложить все векторы слов, которые составляют текст. Для улучшения результатов необходимо отбросить все слова-шумы, расстояние от которых до центров кластеров не превышает пороговое значение, а также нормировать полученный семантический вектор текста количеством входящих в него слов.

[5] Word2Vec. URL: https://code.google.com/archive/p/word2vec/.

Сравнение алгоритмов

Для обучения алгоритма word2vec была использована модель, построенная на части дампа сайта wikipedia.org за 2014 г. [6] Данная модель содержит приблизительно двести тысяч векторных представлений слов. На основании этой модели были произведены кластеризация на 100 кластеров с помощью алгоритма k-means, реализованного в библиотеке Accord.Net [7], и подсчет векторного представления текстов по описанному выше алгоритму.

В качестве выборки для тестирования алгоритмов векторного представления текстов были использованы тексты на естественном языке, полученные при обращении в хранилище link.springer.com по имени Быстров Александр Васильевич. В результате получено 10 документов, 2 из которых не содержали текста, поэтому сравнение было произведено по тем 8 документам, которые имели текст публикации.

Ниже представлены матрицы схожести данных текстов, построенные на основании меры TF-IDF и алгоритмов word2vec (табл. 1, 2).

Таблица 1

Матрица смежности,
полученная при сравнении текстовых данных TF-IDF мерой

1.0000	0.0101	0.0073	0.0103	0.1167	0.0084	0.0068	0.0100
0.0101	1.0000	0.0162	0.4791	0.0164	0.1977	0.0327	0.2201
0.0073	0.0162	1.0000	0.0157	0.0206	0.0120	0.0252	0.0204
0.0103	0.4791	0.0157	1.0000	0.0373	0.1679	0.0248	0.2957
0.1167	0.0164	0.0206	0.0373	1.0000	0.0113	0.0344	0.0168
0.0084	0.1977	0.0120	0.1679	0.0113	1.0000	0.0205	0.1296
0.0068	0.0327	0.0252	0.0248	0.0344	0.0205	1.0000	0.0262
0.0100	0.2201	0.0204	0.2957	0.0168	0.1296	0.0262	1.0000

Таблица 2

Матрица смежности,
полученная при сравнении текстовых данных word2vec

1.0000	0.9477	0.9211	0.9426	0.9681	0.9484	0.9448	0.9388
0.9477	1.0000	0.9258	0.9925	0.9613	0.9757	0.9630	0.9768
0.9211	0.9258	1.0000	0.9294	0.9493	0.9216	0.9118	0.9165
0.9426	0.9925	0.9294	1.0000	0.9620	0.9774	0.9633	0.9846
0.9681	0.9613	0.9493	0.9620	1.0000	0.9571	0.9594	0.9534
0.9484	0.9757	0.9216	0.9774	0.9571	1.0000	0.9659	0.9803
0.9448	0.9630	0.9118	0.9633	0.9594	0.9659	1.0000	0.9618
0.9388	0.9768	0.9165	0.9846	0.9534	0.9803	0.9618	1.0000

Как видно из таблиц, результаты, полученные на основании алгоритма word2vec, являются плохо разделимыми. Такое возможно из-за недостаточно точно обученной модели. Для использования более крупных моделей требуется больше вычислительных мощностей и времени, что неприменимо в данной системе, когда эксперту необходимо взаимодействовать с ней и изменять параметры группировки по ходу работы.

[6] Word2vec API. Pretrained models. URL: https://github.com/3Top/word2vec-api/.
[7] Accord.net framework. URL: http://accord-framework.net/.

Результаты тестирования с применением кластеризации

Добавление в систему модуля кластеризации статей, не распознанных на этапе сравнения с публикациями из eLIBRARY, позволило улучшить результат идентификации авторства статей до 92 %. Следует отметить, что получение стопроцентной точности автоматической идентификации представляется маловероятным, при этом экспертный анализ позволяет достичь гораздо более высоких результатов, но отличается высокой трудоемкостью. Таким образом, стоит признать наиболее оптимальным вариант полуавтоматической обработки данных о публикациях с целью установления авторства. При этом необходимо представлять результаты автоматической идентификации в удобном для эксперта формате, чтобы упростить и ускорить процесс экспертного анализа.

Визуализация полученных результатов

Количество документов в коллекции, для которых необходимо произвести атрибуцию, может достигать десятков, а то и сотен. Анализировать полученные результаты в виде текстовых данных затруднительно, эксперт может потратить большое количество времени. Поэтому для упрощения понимания результатов и взаимодействия пользователя с системой используется визуализация информации.

В разработанной системе пользователю предлагается рассмотрение результатов на различных уровнях. Такая методика применяется во многих системах: она позволяет взглянуть на результаты с разных сторон: например, на результаты в целом и на внутреннее представление объектов. Это также позволяет производить более тонкую настройку инструмента пользователем, поскольку он может исключить из рассмотрения ненужные признаки или выделить признаки, вносящие наибольший вклад в целевую функцию.

В главном меню пользователю предлагается ввести ФИО искомого автора и запустить программу. Также есть возможность просмотреть все генерируемые транслитерации и сокращения для данного имени на русском языке. В текстовом поле отображается текущий статус работы системы, ведется логирование всех действий.

Первый уровень – визуализация групп объектов по сущности (автору). Это позволяет сразу взглянуть на итоговые результаты и внести коррективы. В качестве визуализации предлагается круговая диаграмма, в которой каждая доля показывает выделенную алгоритмом анализа группу (рис. 2). Размер долей в круговой диаграмме прямо пропорционален количеству документов из коллекции, которые система определила в данную группу. На этом уровне пользователю предлагается просмотр краткого текстового описания группы документов, которое появляется после нажатия на долю круговой диаграммы. Также доступны тонкие настройки параметров группировки, такие как использование различных параметров в целевой функции и порог целевой функции. При изменении данных параметров система автоматически пересчитывает результаты, что добавляет визуализации интерактивный характер. Также эксперту доступно редактирование полученных результатов. В диалоге (рис. 3) можно изменять группы публикаций с помощью переноса статей из одной группы в другую. При нажатии кнопки «Показать детальнее» открывается следующий уровень визуализации – визуализация отдельной группы статей, а при нажатии кнопки «Сохранить результаты» пользователь может выбрать путь для сохранения данных, а также информации о текущем разбиении.

Следующий уровень представления – внутреннее представление сформированной группы документов (рис. 4). На этом уровне коллекция представлена в виде матрицы смежности документов, попавших в данную группу. Коэффициенты схожести отображены в виде окружностей, радиус которых зависит от веса коэффициента. При нажатии на определенную точку она выделяется красным цветом, появляется текстовое описание пары документов, а также развернутое пояснение полученного коэффициента. В случае если документ был отнесен к данной группе на этапе кросс-языковой идентификации с библиотекой eLIBRARY.ru, окружность изначально имеет зеленый цвет, а информация об авторе, указанном в eLIBRARY.ru, добавляется в краткое текстовое описание.

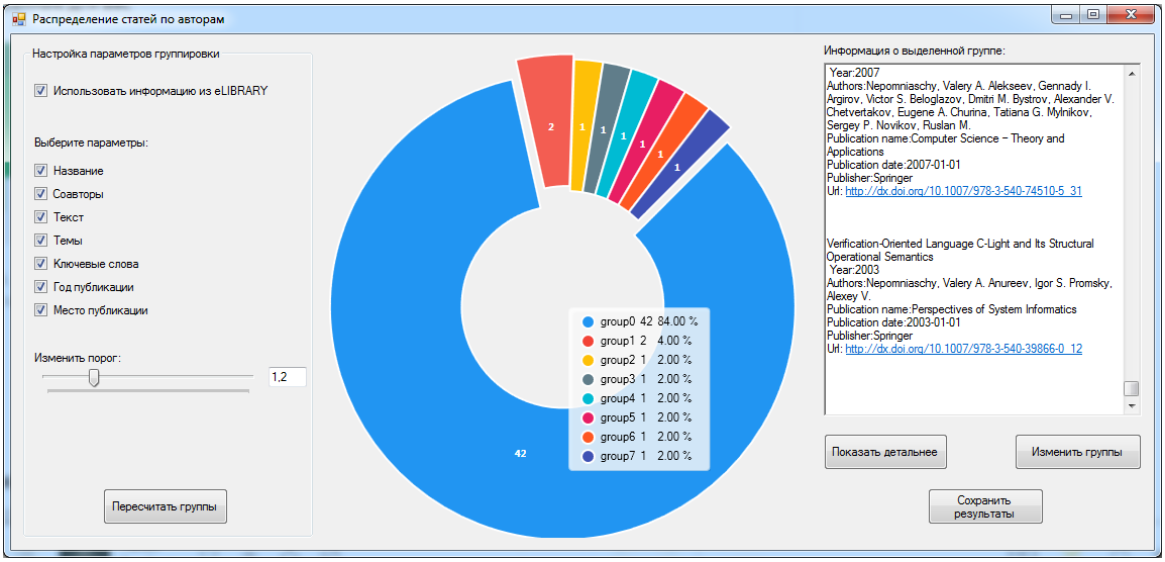


Рис. 2. Представление распределения статей по авторам

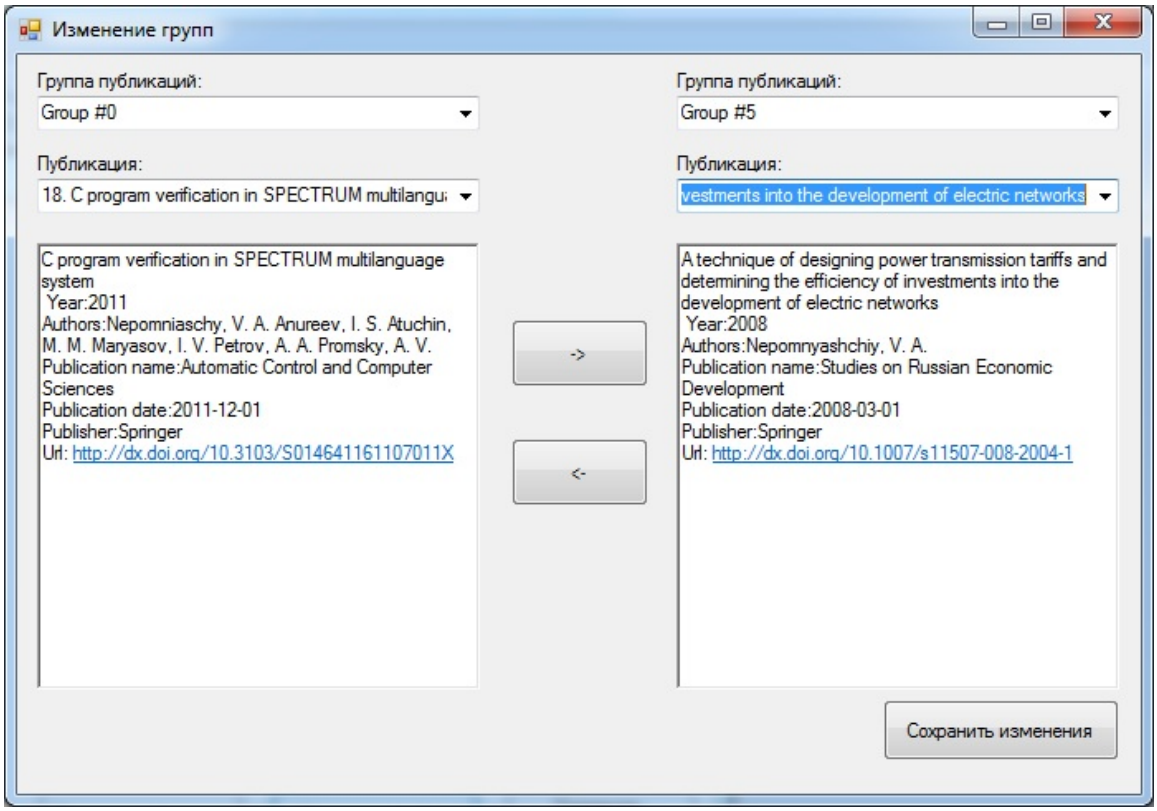


Рис. 3. Диалог для настройки полученных групп

При нажатии кнопки «Соавторство» открывается очередной уровень визуализации, представляющий соавторов научных публикаций в виде матрицы (рис. 5). Данный уровень помогает искать так называемые «выбросы» в группе – такие публикации, которые в действительности не принадлежат данному автору, в отличие от остальных. Например, эксперт точно знает группу ученых, вместе с которыми публиковался данный человек, а значит, может точно определить, что некоторые статьи в этом наборе лишние. Для этого предусмотрено выделение интересующей эксперта статьи, и по нажатию кнопки «Убрать из группы»

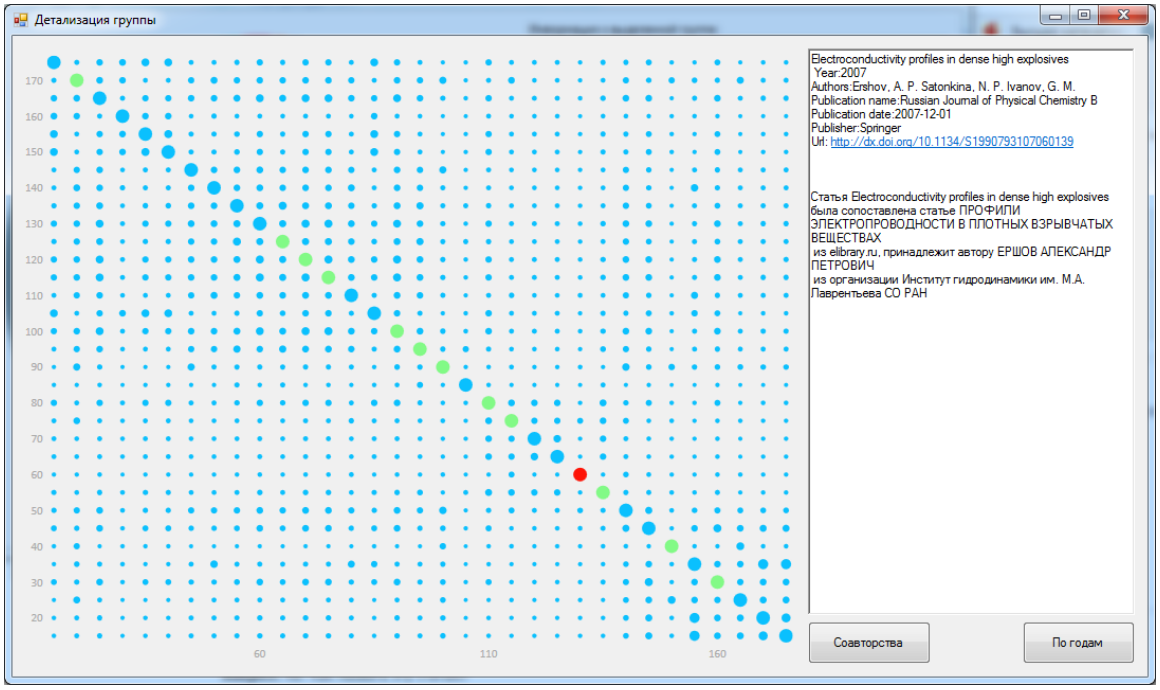


Рис. 4. Результаты в виде матрицы смежности внутри группы публикаций

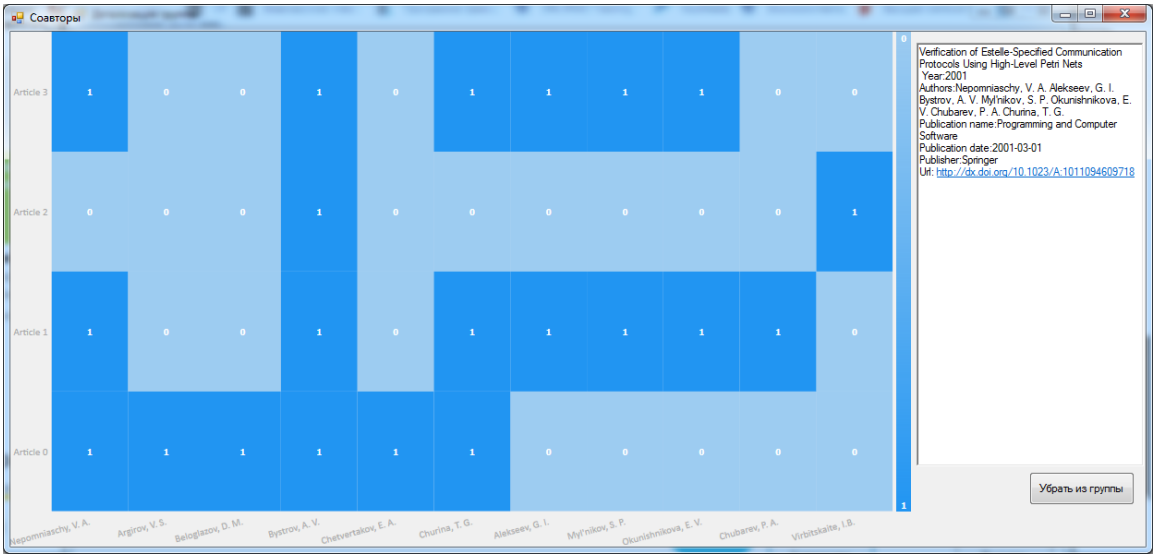


Рис. 5. Таблица соавторства в выделенной группе публикаций

текущая статья будет перемещена из группы. Система либо автоматически распределит эту публикацию в другую группу, либо создаст новую группу, содержащую эту статью.

Помимо перечисленного, пользователю системы предлагается для изучения распределение научных статей автора по году публикации (рис. 6). Оно отображается при нажатии кнопки «По годам» и также может помочь при поиске научных публикаций, не принадлежащих данному автору.

Заключение

В статье представлена система анализа и визуализации для разрешения неоднозначности авторства англоязычных статей хранилища link.springer.com при помощи сопоставления с русскоязычным источником данных elibrary.ru.

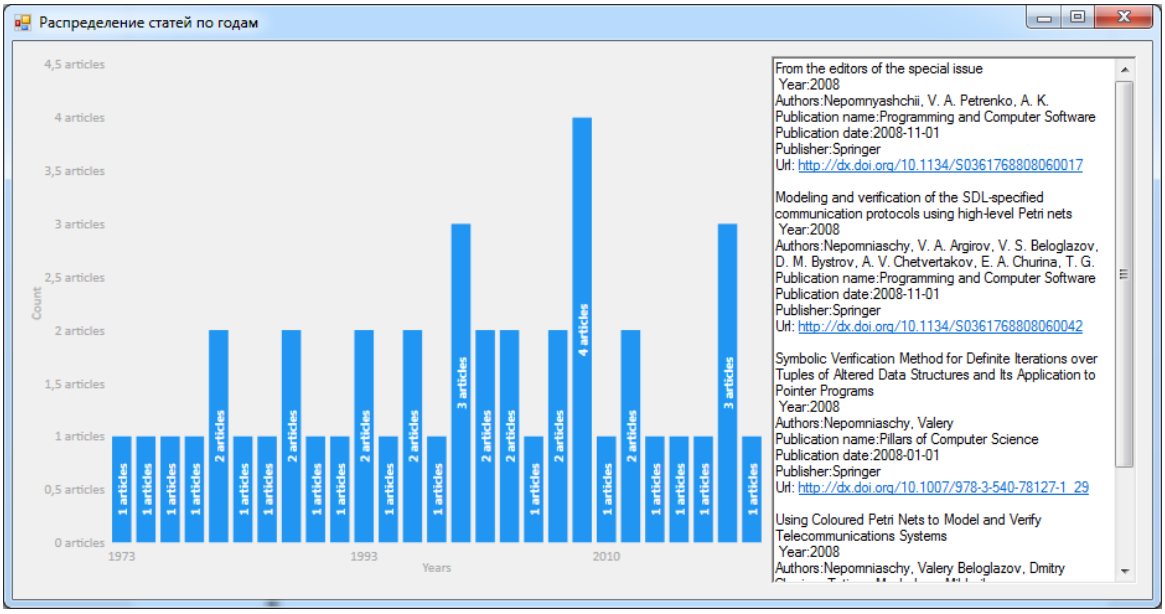


Рис. 6. Распределение статей в группе по году публикации

Реализованная система:

- генерирует множество вариантов транслитераций имен авторов;
- использует в качестве источника достоверных данных на русском и английском языках электронную библиотеку научных публикаций eLIBRARY.ru;
- на основании извлекаемых метаданных и текста публикации идентифицирует авторов исходной коллекции документов;
- показала результат распознавания 92 % (протестирована на выборке авторов из ИСИ СО РАН);
- предоставляет интерактивную визуализацию для упрощения интерпретации полученных результатов и анализа коллекции.

В дальнейшем планируется добавить дополнительные виды визуализации, помогающие не только искать выбросы в полученных группах, но и точнее настраивать алгоритм кластеризации и анализировать полученные группы, например, изменение тематики с течением времени. Также планируется расширить систему для использования различных англо- и русскоязычных баз данных, предоставляющих информацию о публикациях.

Список литературы

1. *Ferreira A. A., Gonçalves M. A., Laender A. H. F.* A brief survey of automatic methods for author name disambiguation // ACM SIGMOD Record. 2012. Vol. 41. No. 2.

2. *Shen Q., Wu T., Yang H., Wu Y., Qu H., Cui W.* Nameclarifier: A visual analytics system for author name disambiguation // IEEE Trans. Vis. Comput. Graph. 2017. Vol. 23. No. 1. P. 141–150.

3. *Apanovich Z. V., Cherepanov D. N., Marchuk A. G.* Cross-language identity resolution and approaches to its solution // Bulletin of the Novosibirsk Computing Center. Series: Computer Science. 2014. P. 41–54.

4. *Apanovich Z. V., Marchuk A. G.* Experiments on Russian-English identity resolution // Proceedings of the ICADL-2015 Conference. Seul, South Korea, LNCS 9469. Springer International Publishing, Switzerland, 2015. P. 12–21.

5. Fifth United Nations Conference on the Standardization of Geographical Names. 1987. Vol. 1: Report of the Conference. P. 40–41.

6. *Cormen T. H., Leiserson C. E., Rivest R. L., Stein C.* Introduction to Algorithms. 3rd ed. MIT Press, 2009.

7. *Большакова Е. И., Клышинский Э. С., Ландэ Д. В., Носков А. А., Пескова О. В., Ягунова Е. В.* Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: Учеб. пособие. М.: МИЭМ, 2011. 272 с.

8. *Salton G., Buckley C.* Term-weighting approaches in automatic text retrieval // Information Processing & Management. 1988. Vol. 24 (5). P. 513–523.

Материал поступил в редколлегию 04.04.2018

V. V. Isachenko [1], **Z. V. Apanovich** [1, 2]

[1] *Novosibirsk State University*
1 Pirogov Str., Novosibirsk, 630090, Russian Federation

[2] *A. P. Ershov Institute of Informatics Systems SB RAS*
6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation

vv.isachenko@gmail.com, apanovich@iis.nsk.su

SYSTEM OF ANALYSIS AND VISUALIZATION FOR CROSS-LANGUAGE IDENTIFICATION OF THE AUTHORS OF SCIENTIFIC PUBLICATIONS

This paper describes a system for disambiguation of authorship of articles in English using Russian-language data sources. The system allows a user to find and correct mistakes in determining the authorship of scientific publications, which can improve the search results for articles by a certain author and calculation of the citation index.

As a source of publications, the link.springer.com database was used. To obtain reliable information about authors and their articles, the eLIBRARY digital library was used.

The system provides interactive visualization of the analysis results and editing facilities to improve the quality of expert analysis. The approaches used in this system are applicable for disambiguation of the authorship of publications from various bibliographic databases.

Keywords: authorship disambiguation, cross-language identity resolution, natural language processing, interactive visualization, clustering.

References

1. Ferreira A. A., Gonçalves M. A., Laender A. H. F. A brief survey of automatic methods for author name disambiguation. *ACM SIGMOD Record*, 2012, vol. 41, no. 2.

2. Shen Q., Wu T., Yang H., Wu Y., Qu H., Cui W. Nameclarifier: A visual analytics system for author name disambiguation. *IEEE Trans. Vis. Comput. Graph*., 2017, vol. 23, no. 1, p. 141–150.

3. Apanovich Z. V., Cherepanov D. N., Marchuk A. G. Cross-language identity resolution and approaches to its solution. *Bulletin of the Novosibirsk Computing Center. Series: Computer Science*, 2014, p. 41–54.

4. Apanovich Z. V., Marchuk A. G. Experiments on Russian-English identity resolution. *Proceedings of the ICADL-2015 Conference. Seul, South Korea, LNCS 9469*. Springer International Publishing, Switzerland, 2015, p. 12–21.

5. *Fifth United Nations Conference on the Standardization of Geographical Names*, 1987, vol. 1: Report of the Conference, p. 40–41.

6. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms. 3rd ed. MIT Press, 2009.

7. Bolshakova E. I., Klyshinskiy E. S., Lande D. V., Noskov A. A., Peskova O. V., Yagunova E. V. Automatic processing of texts in natural language and computer linguistics. Moscow, MIAM Press, 2011, 272 p.

8. Salton G., Buckley C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 1988, vol. 24 (5), p. 513–523.

For citation:

Isachenko V. V., Apanovich Z. V. System of Analysis and Visualization for Cross-Language Identification of the Authors of Scientific Publications. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 49–61. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-49-61

УДК 026.06:004.62.032.2
DOI 10.25205/1818-7900-2018-16-2-62-69

А. А. Князева [1], **О. С. Колобов** [2], **И. Ю. Турчановский** [1], **А. М. Федотов** [1]

[1] *Институт вычислительных технологий СО РАН*
пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия

[2] *Институт сильноточной электроники СО РАН*
пр. Академический, 2/3, Томск, 634055, Россия

aknjazeva@ict.nsc.ru, okolobov@hcei.tsc.ru, tur@hcei.tsc.ru, fedotov@sbras.ru

КОЛЛАБОРАТИВНАЯ ФИЛЬТРАЦИЯ ДЛЯ ПОСТРОЕНИЯ РЕКОМЕНДАЦИЙ НА ОСНОВЕ ДАННЫХ О ЗАКАЗАХ *

Рассматривается возможность применения методов коллаборативной фильтрации в процессе создания рекомендательной системы на основе данных о заказах документов из библиотечного фонда. Приводится сравнительный экспериментальный анализ трех методов коллаборативной фильтрации: на основе документов, на основе пользователей и на основе гибридного метода, являющегося комбинацией первых двух методов.

Ключевые слова: рекомендательная система, коллаборативная фильтрация, унарные данные, бинарные данные.

Введение

Рекомендательные системы открывают новые возможности навигации в процессе информационного поиска. Очевидно, что одной из областей их применения могут быть библиотечные фонды [1]. Учет поведения пользователей для ранжирования документов, с которыми они взаимодействуют, ведет к установлению новых взаимосвязей между этими документами, выходящих за пределы традиционной рубрикации и ключевых слов. Такой учет позволяет связывать документы из смежных областей знания в условиях, когда в них используется различная терминология. В рамках данной работы рассматривалась возможность применения методов коллаборативной фильтрации для создания рекомендательной системы на основе данных о заказах документов в электронном каталоге Научно-технической библиотеки Томского политехнического университета (НТБ ТПУ). Задачи, поставленные в рамках исследования: 1) предварительная оценка качества рекомендаций на основе документов и на основе пользователей по сравнению с базовым методом без персонализации; 2) оценка качества работы гибридной рекомендательной системы, объединяющей описанные выше методы; 3) подбор некоторых параметров будущей системы.

Описание данных

В работе использовались данные о заказах читателей НТБ ТПУ за 2015 г., представленные в виде таблицы из двух столбцов: в первом столбце содержатся идентификаторы пользовате-

* Работа выполнена при частичной поддержке фонда РФФИ (проект № 18-07-01457).

Князева А. А., *Колобов О. С.*, *Турчановский И. Ю.*, *Федотов А. М.* Коллаборативная фильтрация для построения рекомендаций на основе данных о заказах // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 62–69.

ISSN 1818-7900. Вестник НГУ. Серия: Информационные технологии. 2018. Том 16, № 2

лей, зашифрованные с помощью хеш-функции для обеспечения анонимности, а во втором – идентификаторы документов. В качестве документа может выступать любой объект, библиографическое описание которого присутствует в электронном каталоге (книга, статья, цифровой носитель и т. д.). Каждая строка отражает факт заказа читателем документа без указания времени заказа. Строго говоря, описанные данные являются унарными. Это означает, что мы знаем лишь о положительном отклике пользователя: факте заказа. При этом мы не знаем, насколько высоко пользователь оценил данный документ (рейтинги неизвестны), а также не обладаем сведениями об отрицательном отклике. Если пользователь не заказал конкретный документ, то причин может быть несколько:

- данный документ не является релевантным;
- документ релевантен, известен пользователю и, следовательно, не должен быть рекомендован;
- документ релевантен, неизвестен пользователю.

Очевидно, при построении рекомендаций необходимы документы последней группы. Однако выделить их на основе имеющихся данных не представляется возможным. Для создания тестовой выборки при оценке качества работы рекомендательной системы мы вынуждены использовать допущение, что все документы, которые не были заказаны, являются нерелевантными. Технически это означает замену всех неопределенных значений на нули и переход от унарного типа данных к бинарному [3]. Если пользователь заказывал документ, то на пересечении соответствующих строки и столбца стоит единица, в противном случае – ноль. Такой подход позволяет формировать группу нерелевантных документов для проверки без оценки пользователем каждого документа в коллекции, что, как правило, невозможно.

Дополнительно в работе был задействован набор данных под названием MSWeb, предоставляемый в рамках используемого инструментария. Данные получены путем выборочного анализа лог-файлов сайта www.microsoft.com. Они представляют собой записи об обращениях к различным областям сайта анонимных пользователей, выбранных случайным образом, и также приведены к бинарному виду. Временной период: одна неделя в феврале 1998 г. В роли документа выступает область сайта. Набор данных MSWeb является вспомогательным, его использование в данной работе обусловлено стремлением выделить особенности данных о заказах НТБ.

Для того чтобы исключить из работы пользователей и документы, о которых слишком мало информации, были применены следующие фильтры (в указанном порядке):

1) исключение документов, которые были заказаны менее чем 4-мя пользователями;
2) исключение пользователей, которые заказали менее 4-х документов.

Описанная фильтрация позволяет существенно сократить объем данных для работы (табл. 1). Кроме того, она позволяет составлять тестовую выборку из тех пользователей, кто заказал 4 и более документов. Это означает, что мы можем строить рекомендации на основе трех документов и иметь как минимум один документ для проверки.

Таблица 1

Количественное описание данных

Данные	До фильтрации		После фильтрации	
	НТБ	MSWeb	НТБ	MSWeb
Записи о заказах / просмотрах	98 341	98 653	51 513	57 497
Уникальные пользователи	9 619	32 710	4 786	9 544
Уникальные документы	37 718	285	3764	231

Краткое описание инструментария и моделей

В работе была использована библиотека *recommenderlab* [2] для вычислительной среды R project. С помощью данной библиотеки для исходных данных были построены следующие варианты рекомендательных систем:

1) рекомендации по популярности (*Popular*);

2) коллаборативная фильтрация на основе документов (*Item-based collaborative filtering*, *IBCF*);

3) коллаборативная фильтрация на основе пользователей (*User-based collaborative filtering*, *UBCF*);

4) гибридный подход (*Hybrid*).

Первый способ, при котором всем пользователям рекомендуются наиболее популярные документы, был использован в качестве базового метода для сравнения. Рекомендации, полученные с его помощью, не являются персонализированными.

Модели на основе сходства документов используют предположение, что похожие между собой документы будут оцениваться пользователями сходным образом. Таким образом, производится вычисление меры схожести для каждой пары документов, и задействуются те документы, для которых значения меры наибольшие.

Модели на основе пользователей базируются на аналогичной идее: похожие между собой пользователи оценивают документы приблизительно одинаково. Для того чтобы спрогнозировать оценку данным пользователем конкретного документа, можно привлечь оценки других пользователей, похожих на данного пользователя [3]. Количество похожих документов или пользователей может варьироваться. В данной работе оно задается с помощью значения параметра k.

Гибридный метод подразумевает комбинацию двух списков рекомендаций с заданными весовыми коэффициентами. В данной работе комбинировались два варианта коллаборативной фильтрации: на основе документов и на основе пользователей.

Используемые меры схожести

Для оценки того, насколько документы или пользователи похожи между собой, были использованы следующие меры:

1) коэффициент Жаккара [4];
2) мера Дайса [5];
3) косинусная мера [3];
4) коэффициент корреляции Пирсона [3].

Описание экспериментов

Данные, используемые в работе, были случайным образом разбиты на обучающую (70 % пользователей) и тестовую (30 %) выборки. Для пользователей из тестовой выборки, в свою очередь, производилось разделение документов. Для каждого пользователя были выбраны по три документа, на основе которых строились рекомендации. Размер списка рекомендаций описывается параметром N. Полученные рекомендации сравнивались с остальными «скрытыми» документами пользователя. По результатам сравнения были вычислены оценки качества работы системы.

Качество рекомендаций оценивалось с помощью показателей, традиционно используемых для оценки качества информационного поиска: полноты, точности и F-меры [6].

Коллаборативная фильтрация на основе документов

Результаты применения различных мер схожести для построения рекомендаций на основе документов можно проиллюстрировать с помощью так называемых кривых «полнота-точность» (рис. 1).

Как видно из рис. 1, коэффициент Жаккара и мера Дайса дают очень близкие результаты, тогда как коэффициент Пирсона им несколько проигрывает. Значения полноты и точности для косинусной меры настолько малы, что вся кривая выглядит как одна точка рядом с началом координат. Такая аномалия проявляется за счет особенностей реализации построения рекомендаций с помощью косинусной меры в библиотеке *recommenderlab*. Использование

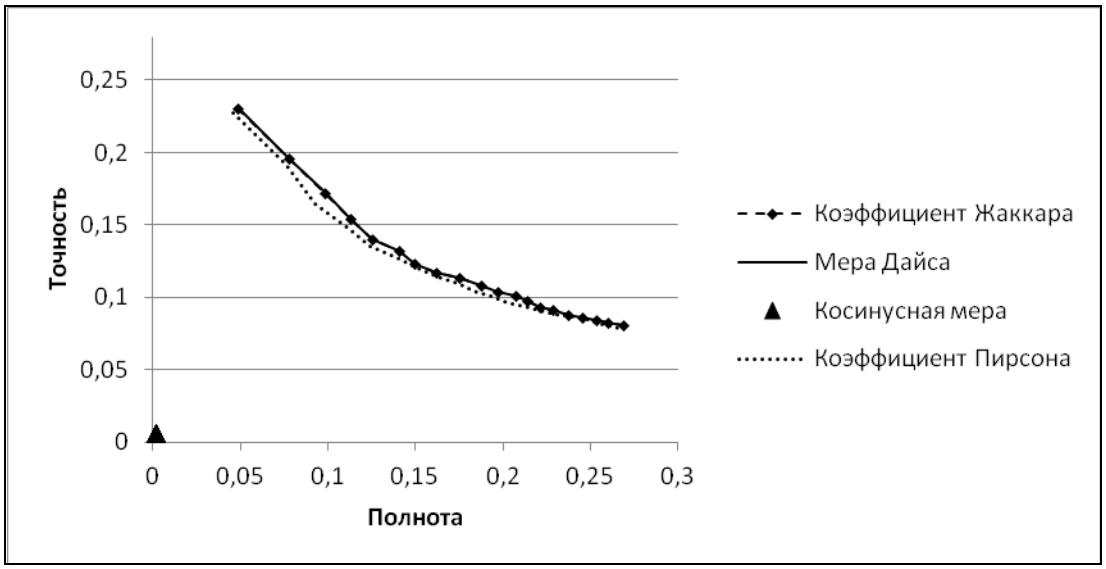


Рис. 1. Кривые «полнота-точность» для рекомендаций на основе документов (IBCF) в зависимости от меры схожести (параметр $k = 30$; количество рекомендаций N изменяется от 1 до 20)

данной меры схожести приводит к тому, что слишком многие документы получают максимально возможное значение меры схожести. В случае, когда для некоторого документа количество максимально схожих с ним документов больше значения параметра k, выбор k ближайших соседей становится проблематичным. Используемый инструментарий в этом случае возвращает пустое множество вместо списка рекомендаций. Таким образом, рекомендации на основе косинусной меры были сформированы менее чем для 4 % пользователей тестовой выборки. Для оставшихся пользователей были приняты нулевые значения показателей полноты и точности.

Коллаборативная фильтрация на основе пользователей

Для вычисления сходства между пользователями использовались уже перечисленные меры схожести (рис. 2).

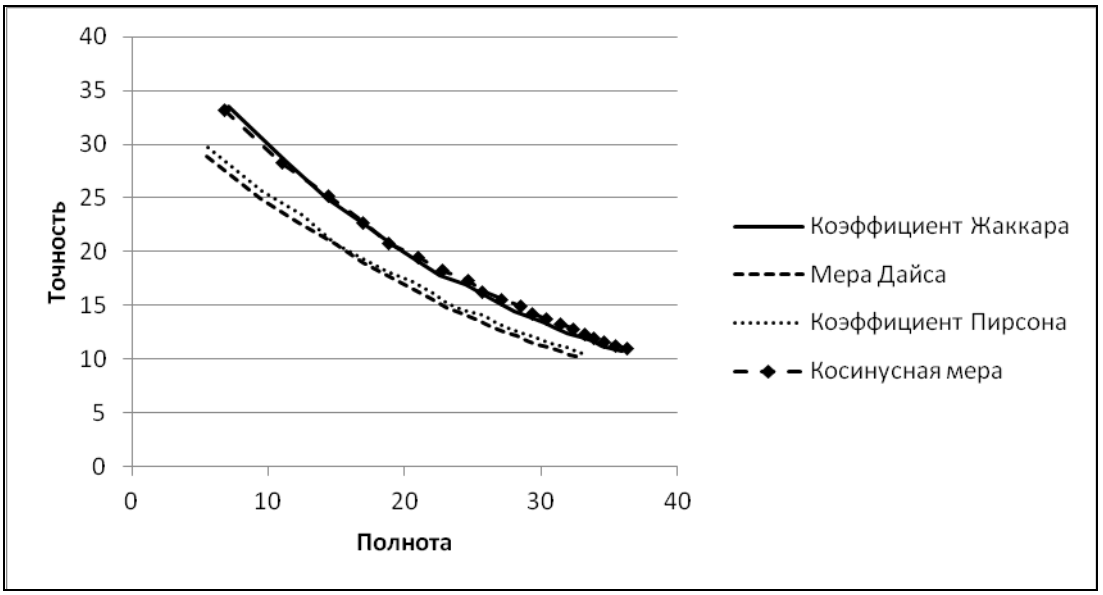


Рис. 2. Кривые «полнота-точность» для рекомендаций на основе пользователей (UBCF) в зависимости от меры схожести (параметр $k = 50$; количество рекомендаций N изменяется от 1 до 20)

Лучшие результаты для данных НТБ ТПУ показала косинусная мера, которой незначительно уступает коэффициент Жаккара. Иллюстрация того, как на качество рекомендаций влияет количество ближайших соседей, приведена на гистограмме (рис. 3). Из рассмотренных значений параметров рекомендательной системы наиболее качественные рекомендации дает использование параметра $k = 50$ в сочетании с косинусной мерой. По результатам аналогичного анализа метода построения рекомендаций на основе документов параметр k был выбран равным 30. В табл. 2 приведены показатели качества для двух вариантов коллаборативной фильтрации.

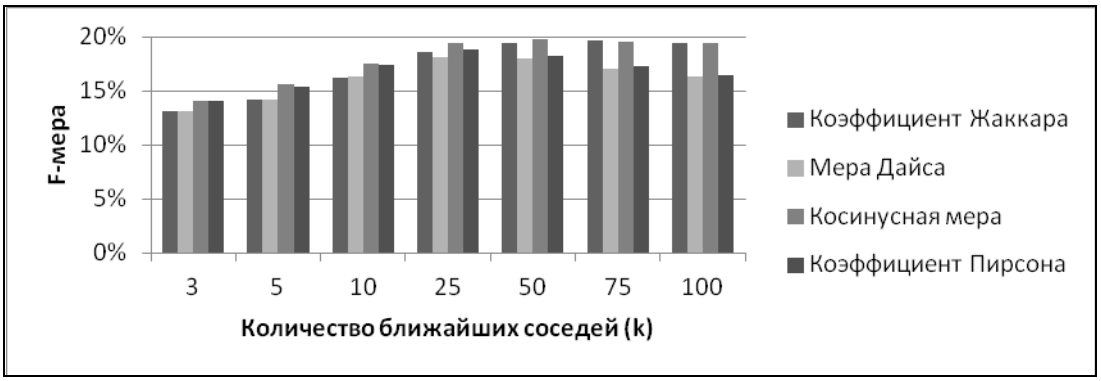


Рис. 3. F-мера для рекомендаций на основе пользователей (UBCF) в зависимости от меры схожести и значения параметра k ($N = 10$)

Таблица 2

Оценки качества (%) для списка из 10 рекомендаций

Мера сходства	Рекомендации					
	основанные на документах (IBCF, $k = 30$)			основанные на пользователях (UBCF, $k = 50$)		
	точность	полнота	F-мера	точность	полнота	*F*-мера
Жаккара	10,85	18,75	13,74	15,15	27,01	19,41
Пирсона	10,46	18,33	13,32	14,53	24,27	18,18
Косинусная	0,65	0,18	0,28	15,58	27,06	19,78
Дайса	10,84	18,73	13,73	14,26	24,28	17,97

В результате проведенных экспериментов для рекомендаций на основе документов был выбран коэффициент Жаккара, а для рекомендаций на основе пользователей – косинусная мера. При этом результаты метода на основе пользователей заметно превосходят качество рекомендаций на основе документов.

Гибридный метод

При создании гибридного подхода были использованы два метода: построение рекомендаций на основе пользователей с применением косинусной меры и на основе документов с использованием коэффициента Жаккара. Пропорция для комбинирования методов задавалась с помощью параметра a:

$$R_{hybrid} = aR_{UBCF} + (1-a)R_{IBCF}.$$

Зависимость *F*-меры от параметра a проиллюстрирована на рис. 4.

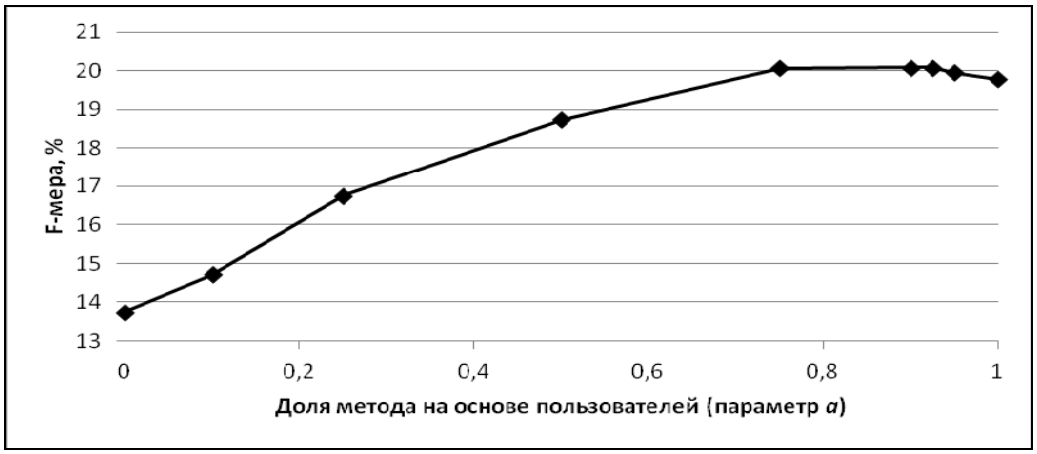


Рис. 4. F-мера для гибридных рекомендаций в зависимости от значений параметра *a* (количество рекомендаций равно 10)

Среди рассмотренных значений лучшим оказалось значение $a = 0{,}925$. При этом достигнутое значение *F*-меры превосходит значение, полученное методом на основе пользователей. Таким образом, комбинация превосходит по качеству каждый метод в отдельности.

Сравнение описанных методов

В табл. 3 приведены оценки качества рекомендаций рассмотренных выше подходов. Для всех методов параметр $N = 10$. Значения показателей заметно различаются для двух наборов. Набор данных MSWeb можно назвать более «предсказуемым», поскольку он позволяет добиться более высоких показателей качества (значение *F*-меры достигает 27,47 %). Набор данных НТБ ТПУ показывает более скромные результаты, но при этом он характеризуется значительной разницей между рекомендациями по популярности и коллаборативной фильтрацией. Выигрыш в *F*-мере для гибридного метода составляет всего 1,5 % от значения для метода на основе пользователей, в то же время он является значительно более трудоемким (табл. 4).

Таблица 3

Сравнение качества рекомендаций для описанных подходов ($N = 10$), %

Показатель качества	По популярности		На основе документов (коэф. Жаккара, $k = 30$)		На основе пользователей (косинусная мера, $k = 50$)		Гибридный метод ($a = 0{,}925$)	
	НТБ	MSWeb	НТБ	MSWeb	НТБ	MSWeb	НТБ	MSWeb
Точность	3,79	16,16	10,85	17,46	15,58	17,27	15,84	18,50
Полнота	4,78	57,63	18,75	64,37	27,06	65,19	27,43	68,60
F-мера	4,23	25,24	13,74	27,47	19,78	27,31	20,08	29,14

Таблица 4

Среднее время на итерацию вычислений

Рекомендации	Время, с		
	моделирование	формирование рекомендаций	Всего
По популярности	0,004	3,59	3,594
На основе документов	2467,38	2,58	2469,96
На основе пользователей	0,007	59,36	59,367
Гибридный метод	13529,17	848,61	14377,79

Метод, основанный на документах, требует значительно больше времени для моделирования. Это связано с особенностями данного подхода, а также с реализацией данного алгоритма в библиотеке *recommenderlab*. Поскольку количество документов в нашем случае значительно, матрица схожести имеет большую размерность, что затрудняет вычисления. При этом время формирования рекомендаций для пользователей значительно меньше, чем для подхода на основе пользователей. Что касается гибридного метода, выигрыш в качестве, который он обеспечивает, вряд ли может компенсировать его временные затраты.

Заключение

Проведенные эксперименты позволяют утверждать о возможности построения рекомендательной системы методами коллаборативной фильтрации на основе данных о заказах НТБ ТПУ. Использование подхода, основанного на пользователях, позволило добиться более качественных рекомендаций по сравнению с базовым методом – рекомендациями по популярности, а также по сравнению с рекомендациями на основе документов. Гибридный подход с использованием двух методов коллаборативной фильтрации позволил несколько улучшить показатели качества, но при этом потребовал значительного времени как на этапе моделирования, так и на этапе формирования рекомендаций. В рамках дальнейшей работы планируется исследовать возможности сбора более качественной информации о предпочтениях пользователей (например, в виде рейтингов документов), а также оценить возможности привлечения информации из библиографических описаний документов, хранящихся в фонде библиотеки.

Список литературы

1. *Карауш А. С.* Рекомендательные системы в публичных библиотеках // Библиосфера. 2009. № 1. С. 41–43.

2. *Hahsler M.* Recommenderlab: A Framework for Developing and testing Recommender Algorithms. URL: https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf (дата обращения 20.09.2017).

3. *Aggarwal C.* Recommender Systems: The Textbook. Springer International Publishing, Switzerland, 2016. 498 p.

4. *Leskovec J., Rajaraman A., Ullman J. D.* Mining of Massive Datasets. 2nd ed. New York: Cambridge University Press, 2014. 476 p.

5. *Dice L.* Measures of the amount of ecologic association between species // Ecology. 1945. Vol. 26 (3). P. 297–302.

6. *Manning C. D.* Introduction to Information. Retrieval. URL: http://www-nlp.stanford.edu/IR-book/ (дата обращения 20.09.2017).

Материал поступил в редколлегию 17.03.2018

A. A. Knyazeva [1], **O. S. Kolobov** [2], **I. Yu. Turchanovsky** [1], **A. M. Fedotov** [1]

[1] *Institute of Computational Technologies SB RAS*
6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation

[2] *Institute of High Current Electronics SB RAS*
2/3 Akademichesky Ave., Tomsk, 634055, Russian Federation

aknjazeva@ict.nsc.ru, okolobov@hcei.tsc.ru, tur@hcei.tsc.ru, fedotov@sbras.ru

COLLABORATIVE FILTERING FOR CREATION OF RECOMMENDATIONS ON BASE OF ORDER DATA

In the article an opportunity of the collaborative filtering methods application in a process of creating a recommender system on the base of order data of documents from library fund is consid-

ered. A comparison experimental analysis of three collaborative filtering methods is provided: item-based, user-based and hybrid method, which is a combination of first two methods.

Keywords: recommender system, collaborative filtering, unary data, binary data.

References

1. Karaush A. S. Recommender system in a public library. *Bibliosphere*, 2009, no. 1, p. 41–43. (in Russ.).

2. Hahsler M. Recommenderlab: A Framework for Developing and testing Recommender Algorithms. 2011. URL: https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf (access: 19.09.2017).

3. Aggarwal C. Recommender Systems: The Textbook. Springer International Publishing, Switzerland, 2016, 498 p.

4. Leskovec J., Rajaraman A., Ullman J. D. Mining of Massive Datasets. 2nd ed. New York, Cambridge University Press, 2014, 476 p.

5. Dice L. Measures of the amount of ecologic association between species. *Ecology*, 1945, vol. 26 (3), p. 297–302.

6. Manning C. D. Introduction to Information Retrieval. URL: http://www-nlp.stanford.edu/IR-book/ (access: 19.09.2017).

For citation:

Knyazeva A. A., Kolobov O. S., Turchanovsky I. Yu., Fedotov A. M. Collaborative Filtering for Creation of Recommendations on Base of Order Data. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 62–69. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-62-69

УДК 004.9
DOI 10.25205/1818-7900-2018-16-2-70-77

А. В. Козодоев, **Е. М. Козодоева**

Институт оптики атмосферы им. В. Е. Зуева СО РАН
пл. Академика Зуева, 1, Томск, 634055, Россия

kav@iao.ru, klen@iao.ru

БИНАРНЫЕ ОПЕРАЦИИ В ИНФОРМАЦИОННОЙ СИСТЕМЕ «МОЛЕКУЛЯРНАЯ СПЕКТРОСКОПИЯ»

Представлен подход, использованный при разработке и реализации модуля, выполняющего бинарные операции над данными в ИС «Молекулярная спектроскопия». Приводится формализация бинарных операций над наборами спектроскопических данных с учетом особенностей предметной области. Описываются алгоритм действий и интерфейс пользователя для проведения бинарных операций – единые для имеющихся баз данных по различным веществам и нескольким типам спектроскопических данных.

Ключевые слова: структуры данных, количественная спектроскопия, бинарные операции, базы данных.

Введение

Основным способом представления широкой общественности результатов научной деятельности являются публикаций в научных изданиях. При небольшом объеме численных данных, полученных в ходе исследования, они публикуются непосредственно в статье в виде таблиц или графиков. В случае если объем численных данных велик для публикации в статье, такие данные публикуют в сети Интернет, размещая файлы, например, на FTP-серверах.

Современные исследования по молекулярной спектроскопии высокого разрешения в области состояний молекул и характеристик спектральных переходов дают постоянно растущие объемы численных данных благодаря совершенствованию как измерительного оборудования, так и вычислительной техники. Таким образом, результаты экспериментальных работ содержат значения параметров от десятков и сотен до десятков тысяч спектральных переходов или состояний молекулы [1; 2]. Теоретические (расчетные) работы могут содержать характеристики нескольких миллиардов переходов [3; 4].

В информационной системе (ИС) «Молекулярная спектроскопия» ведется накопление численных значений данных, публикуемых исследователями в статьях [5–8]. Совокупность извлеченных из опубликованных материалов численных значений спектроскопических данных будем называть набором данных. В отдельных статьях, как правило, представляются данные по свойствам лишь одной молекулы или небольшой группы молекул. Зачастую приводятся только часть набора параметров спектральных линий или исследования в узком спектральном диапазоне. Однако исследователям из прикладных областей, использующим необходимые спектральные параметры, например, в расчетах радиационных или климатических моделей, требуются данные в достаточно широком спектральном диапазоне. Это приводит к необходимости формирования составных наборов данных путем комбинирования данных из различных источников. Среди составных наборов можно выделить экспертные

Козодоев А. В., *Козодоева Е. М.* Бинарные операции в информационной системе «Молекулярная спектроскопия» // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 70–77.

наборы, предоставляющие достоверные, согласованные и наиболее полные данные как по перечню спектральных параметров, так и по спектральному диапазону [9].

Процесс создания таких наборов можно разделить на два вида действий: отбор данных с помощью условий в пределах одного набора данных и объединение отобранных данных в новый составной набор данных. Формирование составных наборов данных – процесс трудоемкий, но часть действий можно автоматизировать с помощью информационных технологий. Действия по выборке части данных из конкретного источника, удовлетворяющих наложенным ограничениям, были автоматизированы созданным в рамках ИС «Молекулярная спектроскопия» модулем «унарные операции» [10]. Для создания составного экспертного источника данных из отобранных частей исходных источников необходим механизм их объединения, позволяющий учитывать как формализуемые ограничения предметной области, так и принимаемые экспертами решения, не поддающиеся алгоритмизации.

Определение бинарных операций в молекулярной спектроскопии

Подробно структура данных молекулярной спектроскопии, над наборами которых производятся манипуляции, была рассмотрена нами в работе [10]. Здесь лишь коротко напомним, что набор данных представим в виде таблицы независимо от способа хранения, и имеются *ограничения на операции*, продиктованные особенностями предметной области.

1. Во всех наборах данных есть две обязательные части: набор квантовых чисел и набор некоторых физических величин. В зависимости от спектроскопической задачи обязательной физической величиной является уровень энергии или частота спектрального перехода. Остальные физические величины могут отсутствовать совсем либо иметь «пустоты» в отдельных строках.

2. Значения квантовых чисел в каждой строке таблицы является уникальным идентификатором, который однозначно определяет, к какому уровню энергии или спектральному переходу относятся значения набора физических величин, представленных в этой строке таблицы, согласно используемой модели молекулы. Сравнение строк производится только по набору значений квантовых чисел, который мы рассматриваем как единый элемент (идентификатор). Два идентификатора равны, если равны между собой все соответствующие квантовые числа. Остальные физические характеристики имеют приближенные значения в силу неточности самих моделей или измерений в эксперименте. Таким образом, все строки в одном наборе данных должны быть с различными (уникальными) идентификаторами как в исходных наборах, так и в результирующем наборе.

3. Операндами в бинарных операциях являются только канонические наборы данных, в которых квантовые числа удовлетворяют ограничениям на состояние и правилам отбора (определяют допустимые сочетания квантовых чисел верхнего и нижнего состояния). Операции производятся над данными, относящимися к одному типу спектроскопических задач (прямая / обратная) и одному веществу [11].

4. В исходных и результирующем наборах данных может быть только по одной колонке с данными по конкретной спектральной характеристике (физической величине).

Рассмотрим три операции над парой наборов данных по аналогии с теорией множеств: объединение, пересечение и разность. Эти операции являются манипуляциями с данными, т. е. действиями, не изменяющими сами значения данных. *Операция разности* наборов данных дает множество строк одного набора данных, не имеющих пары по набору значений квантовых чисел во втором наборе данных. *Операция объединения* наборов данных, позволяет соединять в единый набор строки из двух разных наборов независимо от наличия совпадений по набору значений квантовых чисел. С помощью *операции пересечения* наборов данных, можно выбрать строки из двух наборов с совпадающими наборами значений квантовых чисел.

Специфика предметной области приводит к особому процессу выполнения некоторых операций. Так, в операции разности результат однозначен и может быть получен автоматически, так как среди отобранных строк набора данных не может быть строк с одинаковыми наборами значений квантовых чисел. В то время как в операциях объединения и пересечения

имеет место *промежуточный результат операции*, где могут быть строки с совпадающими наборами значений квантовых чисел, но из разных наборов данных. Согласно принятым нами правилам результирующий набор данных должен содержать только канонические данные [11], т. е. среди прочих условий не иметь строк с совпадающими наборами значений квантовых чисел. Выбор строк, которые попадут в результирующий набор данных, может произвести только пользователь системы (эксперт) на основании имеющихся у него неформализуемых знаний. Это и есть неавтоматизированная часть операций манипулирования наборами данных. Таким образом, в операциях объединения и пересечения автоматически выполняется формирование промежуточного результата с выбранными из разных источников строками и создание нового набора данных с сохранением в него строк, отобранных экспертом.

Формализация бинарных операций

Для описания сути бинарных операций нам удобно использовать термины теории множеств. Обозначим набор данных по конкретному веществу, как $ND^j = \{K_i\}$, где $j \in$ {спектральные переходы, профили спектральных линий, уровни энергии молекулы} – тип спектроскопической задачи, а $\{K_i\}$ – это множество строк набора данных. Согласно теории множеств при выполнении операций (пересечения, объединения и разности) производится сравнение элементов по их значениям, однако в нашей предметной области строка в наборе данных не является элементарным объектом с одним значением. Каждая строка K_i набора данных имеет сложную структуру, для описания которой можно применить кортежи из алгебры кортежей [12].

Тогда элемент множества (набора данных) K_i можно представить кортежем, элементы которого разделены по смыслу на две части, как упоминалось ранее:

$$K_i = \left(i, S_i^j\right),$$

где $i = \left(qn_1, ..., qn_M\right)$ – кортеж, содержащий значения квантовых чисел из предметной области, уникальная комбинация которых (идентификатор) в пределах набора данных однозначно идентифицирует K_i; $S_i^j = \left(s_{i,1}^j, ..., s_{i,N}^j\right)$ – кортеж, содержащий численные значения физических величин, набор которых зависит от спектроскопической задачи *j*, описывающих свойства *i*-го спектрального перехода или состояния молекулы, *N* – число физических характеристик.

Пересечение

Операция «пересечение» двух наборов данных производится построчно на основе сравнения идентификаторов. Выбираются строки с одинаковыми идентификаторами из обоих наборов данных, прочие строки в результат операции не попадают. Промежуточный результат может содержать по два значения одной и той же физической величины, соответствующих одному идентификатору. Так как в нашей информационной системе принято, что одному идентификатору должно соответствовать по одному значению каждой физической величины, то далее пользователь должен выбрать, какое значение физической величины включать в результат.

Пусть $ND1^j = \left\{\left(i, S_i^j\right)\right\}$ и $ND2^j = \left\{\left(p, S_p^j\right)\right\}$ – два исходных набора данных. Их пересечение можно расписать следующим образом:

$$ND1^j \bigcap ND2^j = ND3^j = \left\{\left(r, S_r^j\right)\right\},$$

где $r \in \{i\}$ и $r \in \{p\}$, а $S_r^j = \left(\left[s_{i,1}^j \middle| s_{p,1}^j\right], ..., \left[s_{i,N}^j \middle| s_{p,N}^j\right]\right)$.

С помощью записи $\left[s_{i,n}^{j} \middle| s_{p,n}^{j} \right]$ мы обозначили, что эксперт, производящий операцию, делает выбор между $s_{i,n}^{j}$ и $s_{i,p}^{j}$, основываясь на имеющихся у него неформализуемых знаниях.

Объединение

Операция «объединение» двух наборов данных проводится, как и операция пересечения, на основе сопоставления идентификаторов строк. Выбираются все строки из первого набора данных, а затем к ним добавляются строки из второго набора данных. Из получившегося промежуточного результата в итоговый набор данных попадают строки с несовпадающими идентификаторами. Значения физических величин в строках с совпадающими идентификаторами могут быть включены в результирующий набор данных в зависимости от выбора пользователя либо все из первого набора данных, либо все из второго набора данных. Допускается объединение наборов данных независимо от заполненности строк, т. е. с возможностью образования пропусков в данных (отсутствие значений физических величин).

Пусть имеется два набора данных:

$$ND1^{j} = \left\{ \left(i, S_{i}^{j} \right) \right\}, \; ND2^{j} = \left\{ \left(p, S_{p}^{j} \right) \right\}.$$

Тогда их объединение можно записать так:

$$ND1^{j} \bigcup ND2^{j} = ND3^{j} = \left\{ \left(r, S_{r}^{j} \right) \right\},$$

где S_{r}^{j} содержит строки, удовлетворяющие следующим условиям:

$$S_{r}^{j} = \left(\left[s_{i,1}^{j} \middle| s_{p,1}^{j} \right], ..., \left[s_{i,N}^{j} \middle| s_{p,N}^{j} \right] \right) \text{ при } r \in \{i\} \text{ и } r \in \{p\},$$
$$S_{r}^{j} = \left(s_{i,1}^{j}, ..., s_{i,N}^{j} \right) \text{ при } r \in \{i\} \text{ и } r \notin \{p\},$$
$$S_{r}^{j} = \left(s_{p,1}^{j}, ..., s_{p,N}^{j} \right) \text{ при } r \notin \{i\} \text{ и } r \in \{p\}.$$

Разность

Операция «разность» наборов данных $ND1^{j}$ и $ND2^{j}$ аналогична разности (дополнению) в теории множеств. Так же как и операция пересечения, она выполняется на основе сравнения идентификаторов. Разность множества $ND1^{j}$ и множества $ND2^{j}$ – это множество, содержащее в себе элементы множества $ND1^{j}$, но не входящие в $ND2^{j}$. Обозначается: $ND1^{j}$ без $ND2^{j}$, $ND1^{j} \setminus ND2^{j}$ или $ND1^{j} - ND2^{j}$. Мы будем использовать последний вариант обозначения для записи операции разности наборов данных.

В результате может получиться пустой набор данных, часть $ND1^{j}$ или полностью $ND1^{j}$.

$$ND1^{j} = \left\{ \left(i, S_{i}^{j} \right) \right\}, \; ND2^{j} = \left\{ \left(p, S_{p}^{j} \right) \right\},$$
$$ND1^{j} - ND2^{j} = ND3^{j} = \left\{ \left(r, S_{r}^{j} \right) \right\},$$

где $r \in \{i\}$ и $r \notin \{p\}$.

Обзор интерфейса пользователя модуля «бинарные операции»

Обзор интерфейсов в информационной системе «Молекулярная спектроскопия» для проведения автоматических операций (не требующих участия пользователя) был представлен

в работах [10; 13]. Здесь рассмотрим интерфейс той части ИС, которая относится к выполнению неавтоматизированного этапа операций «объединение» и «пересечение».

После выбора двух источников данных, над которыми будут производиться манипуляции, пользователь делает выбор операции и параметров ее выполнения (рис. 1). В качестве дополнительной информации при принятии решения эксперт может воспользоваться имеющейся в системе характеристикой разупорядочения между выбранными наборами данных.

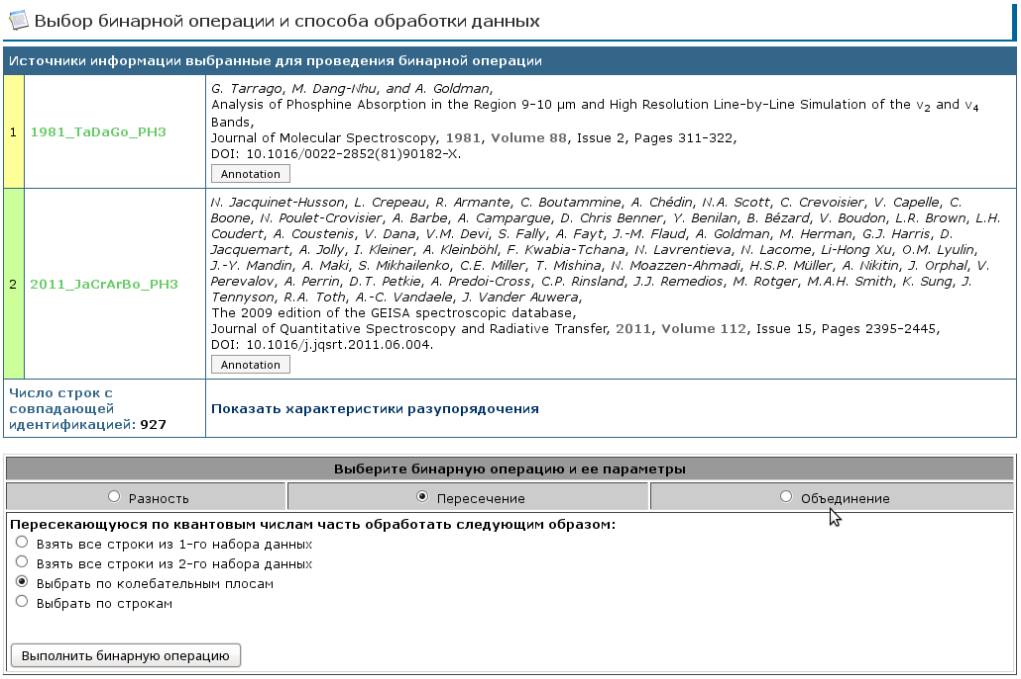


Рис. 1. Интерфейс выбора операции и способа обработки пересекающейся части данных

В операции *объединения* выбор эксперта сводится к указанию, что сделать со всеми строками с совпадающими наборами значений квантовых чисел. Их можно все взять из одного или другого набора данных либо не брать их в результирующий набор данных вообще. Поэтому промежуточный результат операции не отображается, а сразу выдается окончательный результат.

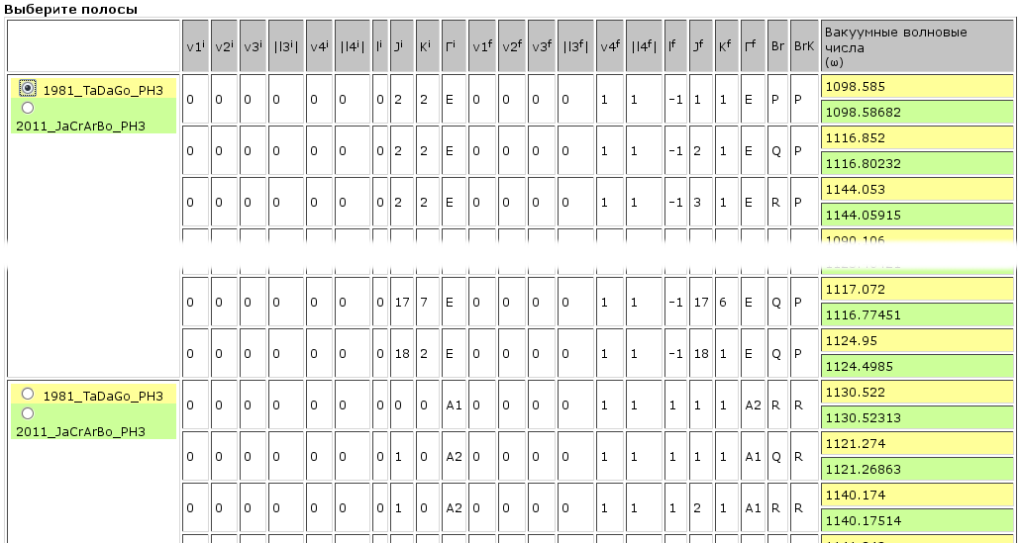
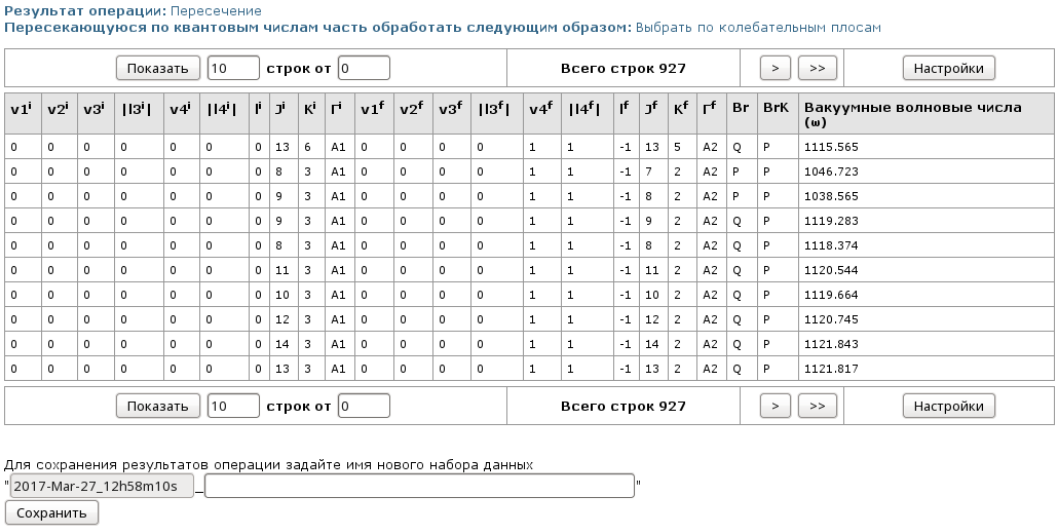


Рис. 2. Интерфейс обработки промежуточного результата по полосам

В операции *пересечения* эксперт имеет возможность детальной обработки промежуточного результата. Он может указать, из какого источника брать данные по конкретной спектральной линии или спектральной полосе в целом (рис. 2). Название источника данных и значения конкретных параметров спектральной линии, взятых из него, подсвечиваются одним цветом. Полученный результат сохраняется как новый набор данных (рис. 3).



$v1^i$	$v2^i$	$v3^i$	$\|l3^i\|$	$v4^i$	$\|l4^i\|$	I^i	J^i	K^i	Γ^i	$v1^f$	$v2^f$	$v3^f$	$\|l3^f\|$	$v4^f$	$\|l4^f\|$	I^f	J^f	K^f	Γ^f	Br	BrK	Вакуумные волновые числа (ω)
0	0	0	0	0	0	0	13	6	A1	0	0	0	0	1	1	-1	13	5	A2	Q	P	1115.565
0	0	0	0	0	0	0	8	3	A1	0	0	0	0	1	1	-1	7	2	A2	P	P	1046.723
0	0	0	0	0	0	0	9	3	A1	0	0	0	0	1	1	-1	8	2	A2	P	P	1038.565
0	0	0	0	0	0	0	9	3	A1	0	0	0	0	1	1	-1	9	2	A2	Q	P	1119.283
0	0	0	0	0	0	0	8	3	A1	0	0	0	0	1	1	-1	8	2	A2	Q	P	1118.374
0	0	0	0	0	0	0	11	3	A1	0	0	0	0	1	1	-1	11	2	A2	Q	P	1120.544
0	0	0	0	0	0	0	10	3	A1	0	0	0	0	1	1	-1	10	2	A2	Q	P	1119.664
0	0	0	0	0	0	0	12	3	A1	0	0	0	0	1	1	-1	12	2	A2	Q	P	1120.745
0	0	0	0	0	0	0	14	3	A1	0	0	0	0	1	1	-1	14	2	A2	Q	P	1121.843
0	0	0	0	0	0	0	13	3	A1	0	0	0	0	1	1	-1	13	2	A2	Q	P	1121.817

Рис. 3. Интерфейс сохранения результата операции в новый набор данных

Такая реализация позволяет сохранять результат детальной обработки в отдельном наборе данных и использовать его потом в других операциях, тем самым уменьшается количество потенциальных ошибок, когда пользователь в при первой обработке промежуточного результата выбрал одно, а при повторной выбрал другое. Детальная обработка пересекающейся части (по строкам и по спектральным полосам) имеется только в операции пересечения.

Заключение

В рамках информационной системы «Молекулярная спектроскопия» создан модуль позволяющий формировать составные наборы данных, путем их комбинирования из различных источников. Это позволяет создавать составные экспертные наборы данных, учитывая как формализуемые ограничения предметной области, так и принимаемые экспертами решения, не поддающиеся алгоритмизации.

Одно из направлений дальнейших исследований – определение критериев для автоматического формирования рекомендаций для эксперта по выбору окончательного результата полуавтоматической части операций.

Список литературы

1. *Сердюков В. И., Синица Л. Н., Быков А. Д., Щербаков А. П.* Уширение и сдвиг спектральных линий метана в области 11000–11400 $см^{-1}$ // Оптика атмосферы и океана. 2017. Т. 30, № 12. С. 1023–1026.

2. *Лукашевская А. А., Люлин О. М., Perrin A., Перевалов В. И.* Глобальное моделирование центров спектральных линий молекулы NO_2 // Оптика атмосферы и океана. 2015. Т. 28, № 01. С. 12–27.

3. *Yurchenko S. N., Barber R. J., Tennyson J.* A variationally computed line list for hot NH_3 // Monthly Notices of the Royal Astronomical Society. 2011. Vol. 413. Iss. 3. P. 1828–1834. https://doi.org/10.1111/j.1365-2966.2011.18261.x

4. *Yurchenko S. N., Tennyson J.* ExoMol line lists IV: The rotation-vibration spectrum of methane up to 1500K. arXiv:1401.4852v1 [astro-ph.EP] 20 Jan 2014

5. *Козодоев А. В.* Система загрузки данных в распределенной информационной системе «Молекулярная спектроскопия» // Материаловедение, технологии и экология в 3-м тысячелетии: Материалы IV Всерос. конф. молодых ученых. Томск: Изд-во Ин-та оптики атмосферы СО РАН, 2009. С. 587–591.

6. *Ахлестин А. Ю., Воронина С. С., Лаврентьев Н. А., Фазлиев А. З.* Информационные ресурсы по спектроскопии в ИОА СО РАН // Оптика атмосферы и океана. 2015. Т. 28, № 05. С. 480–488.

7. *Ахлестин А. Ю., Воронина С. С., Науменко О. В., Половцева Е. Р., Фазлиев А. З.* Информационная система для решения задач молекулярной спектроскопии. 6. Систематизация спектроскопических данных по дейтерозамещенным изотопологам молекулы сероводорода // Оптика атмосферы и океана. 2016. Т. 29, № 05. С. 386–396.

8. *Ахлестин А. Ю., Воронина С. С., Привезенцев А. И., Родимова О. Б., Фазлиев А. З.* Информационная система для решения задач молекулярной спектроскопии. 7. Систематизация информационных ресурсов по поглощению для основного изотополога молекулы метанола // Оптика атмосферы и океана. 2016. Т. 29, № 10. С. 876–887.

9. *Лаврентьев Н. А., Макогон М. М., Фазлиев А. З.* Сравнение спектральных массивов данных HITRAN и GEISA с учетом ограничения на опубликование спектральных данных // Оптика атмосферы и океана. 2011. Т. 24, № 4. С. 279–292.

10. *Козодоев А. В., Козодоева Е. М.* Универсальный модуль «унарные операции» в ИС «Молекулярная спектроскопия» // Вестн. НГУ. Серия: Информационные технологии. 2015. Т. 13, № 1. С. 46–54.

11. *Быков А. Д., Науменко О. В., Родимова О. Б., Синица Л. Н., Творогов С. Д., Тонков М. В., Фазлиев А. З., Филиппов Н. Н.* Информационные аспекты молекулярной спектроскопии. Томск: Изд-во Ин-та оптики атмосферы СО РАН, 2008. 360 с.

12. *Кулик Б. А., Зуенко А. А., Фридман А. Я.* Алгебраический подход к интеллектуальной обработке данных и знаний. СПб.: Изд-во Политехн. ун-та, 2010. 235 с.

13. *Козодоев А. В., Козодоева Е. М.* Формирование наборов данных в ИС «Молекулярная спектроскопия» с использованием бинарных операций // Оптика атмосферы и океана. Физика атмосферы: Материалы XXII Междунар. симп. Томск: Изд-во ИОА СО РАН, 2016. A209-2012.

Материал поступил в редколлегию 03.04.2018

A. V. Kozodoev, **E. M. Kozodoeva**

V. E. Zuev Institute of Atmospheric Optics SB RAS
1 Academician Zuev Square, Tomsk, 634055, Russian Federation

kav@iao.ru, *klen@iao.ru*

THE BINARY OPERATIONS IN THE INFORMATION SYSTEM «MOLECULAR SPECTROSCOPY»

The paper describes an approach to the development and implementation of functions for performing binary operations on data in the IS «Molecular Spectroscopy». The formalization of binary operations over sets of spectroscopic data is made, taking into account the features of the subject area. Describes an action algorithm and a user interface for performing binary operations, one for databases for various substances and several types of spectroscopic data.

Keywords: data structure, quantitative spectroscopy, binary operations, database.

References

1. Serdyukov V. I., Sinitsa L. N., Bykov A. D., Shcherbakov A. P. Broadening and shift of the methane absorption lines in the 11000–11400 cm^{-1} region. *Optika Atmosfery i Okeana*, 2017, vol. 30, no. 12, p. 1023–1026. (in Russ.)

2. Lukashevskaya A. A., Lyulin O. M., Perrin A., Perevalov V. I. Global modeling of NO_2 central line positions. *Optika Atmosfery i Okeana*, 2015, vol. 28, no. 01, p. 12–27. (in Russ.)

3. Yurchenko S. N., Barber R. J., Tennyson J. A variationally computed line list for hot NH_3. *Monthly Notices of the Royal Astronomical Society*, 2011, vol. 413, iss. 3, 21 p. 1828–1834. https://doi.org/10.1111/j.1365-2966.2011.18261.x

4. Yurchenko S. N., Tennyson J. ExoMol line lists IV: The rotation-vibration spectrum of methane up to 1500K. arXiv:1401.4852v1 [astro-ph.EP] 20 Jan 2014

5. Kozodoev A. V. Data loading system in the distributed information system «Molecular spectroscopy». *Materials of the IV all-Russian conference of young scientists materials. Science, technology and ecology in the 3rd Millennium*". Tomsk, Publishing House of the Institute of Atmospheric Optics SB RAS, 2009, p. 587–591. (in Russ.)

6. Akhlyostin A. Yu., Voronina S. S., Lavrent'ev N. A., Fazliev A. Z. Spectroscopic information resources at Institute of Atmospheric Optics SB RAS. *Optika Atmosfery i Okeana*, 2015, vol. 28, no. 05, p. 480–488. (in Russ.)

7. Akhlyostin A. Yu., Voronina S. S., Naumenko O. V., Polovtseva E. R., Fazliev A. Z. Information system for molecular spectroscopy. 6. Systematization of spectral data on deutero-substituted isotopologues of hydrogen sulfide molecule. *Optika Atmosfery i Okeana*, 2016, vol. 29, no. 05, p. 386–396. (in Russ.)

8. Akhlyostin A. Yu., Voronina S. S., Privezentsev A. I., Rodimova O. B., Fazliev A. Z. Information system for molecular spectroscopy. 7. Systematization of information resources on the main isotopologue of methanol molecule. *Optika Atmosfery i Okeana*, 2016, vol. 29, no. 10, p. 876–887. (in Russ.)

9. Lavrent'ev N. A., Makogon M. M., Fazliev A. Z. Comparison of HITRAN and GEISA spectral data, based on taking into account the existent constraints. *Optika Atmosfery i Okeana*, 2011, vol. 24, no. 04, p. 279–292. (in Russ.)

10. Kozodoyev A. V., Kozodoyeva E. M. Extensible module «Unary operations» in the Information system «Molecular spectroscopy». *Vestnik NSU. Series: Information Technologies*, 2015, vol. 13, no. 1, p. 46–54. ISSN 1818-7900. (in Russ.)

11. Bykov A. D., Naumenko O. V., Rodimova O. B., Sinitsa L. N., Tvorogov S. D., Tonkov M. V., Fazliev A. Z., Filippov N. N. Information aspects of molecular spectroscopy. Tomsk, Publishing House of the Institute of Atmospheric Optics SB RAS, 2008, 360 p. (in Russ.)

12. Kulik B. A., Zuenko A. A., Fridman A. Ya. Algebraic approach to intelligent processing of data and knowledge. St. Petersburg Polytechnic University Publishing House, 2010, 235 p. (in Russ.)

13. Kozodoev A. V., Kozodoeva E. M. Building of data sets in is «Molecular spectroscopy» using binary operations. *Atmospheric and ocean Optics. Atmospheric physics: Proceedings of the XXII International Symposium*. Tomsk, Publishing House of the Institute of Atmospheric Optics SB RAS, 2016, A209-2012. (in Russ.)

For citation:

Kozodoev A. V., Kozodoeva E. M. The Binary Operations in the Information System «Molecular Spectroscopy». *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 70–77. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-70-77

УДК 004.4'422
DOI 10.25205/1818-7900-2018-16-2-78-85

А. Е. Малых

Новосибирский государственный университет
ул. Пирогова, 1, Новосибирск, 630090, Россия

ООО НЦИТ «УНИПРО»
ул. Ляпунова, 2, Новосибирск, 630090, Россия

awa149@rambler.ru

РАЗРАБОТКА И РЕАЛИЗАЦИЯ АЛГОРИТМОВ РАЗРЕШЕНИЯ КОНФЛИКТОВ ПО ДОСТУПУ К ПАМЯТИ В ДИНАМИЧЕСКОМ КОМПИЛЯТОРЕ JAVA ДЛЯ ПРОЦЕССОРА «ЭЛЬБРУС»

Рассмотрены особенности разработки алгоритмов разрешения конфликтов по доступу к памяти и их реализация в динамическом компиляторе Java для отечественной платформы «Эльбрус». Эти алгоритмы позволяют существенно расширить возможности планировщика инструкций – ключевой оптимизации VLIW-процессоров. В работе исследуются статические и динамические подходы к анализу зависимостей по памяти, и приводится сравнение эффективности реализованных алгоритмов на основе стандартной тестовой сюиты SpecJVM2008.

Ключевые слова: Эльбрус, Java, JIT-компилятор, планировщик инструкций, оптимизации компилятора.

Постановка задачи

В настоящее время идет активное развитие Российской электроники, в том числе и процессоров. Так, процессоры «Эльбрус», разработанные на основе архитектуры Very Long Instruction Word (VLIW), призваны заменить зарубежные аналоги в отраслях, где ключевым критерием является безопасность, – например, в работе государственных организаций.

Одна из важных задач, которая всегда встает перед разработчиками нового процессора, – это задача адаптации уже разработанного программного обеспечения под новый процессор. Для ее решения необходимо разработать для этого процессора компиляторы и виртуальные машины популярных языков программирования. Одним из примеров таких языков является Java – кросс-платформенный язык, в котором программы преобразуются в архитектурно независимый байт-код, а затем исполняются при помощи виртуальной Java-машины (JVM). Идея работы JVM заключается в том, чтобы интерпретировать методы, которые используются в программе не очень часто, и компилировать в машинный код часто встречающиеся методы. При этом компилятор должен обеспечивать качество кода с помощью встроенных методов оптимизации.

Как и на любом процессоре с архитектурой VLIW, важной оптимизирующей компонентой компилятора является планировщик инструкций. В текущей версии компилятора Java для процессора «Эльбрус» используется суперблоковый планировщик инструкций [1], позволяющий выбрать сразу несколько базовых блоков и запланировать операции внутри них так, как если бы планирование происходило в одном блоке. Зачастую при планировании возникает ситуация, когда одна инструкция должна быть исполнена раньше другой, в таких случаях речь идет о *зависимости* между инструкциями. Если не получается точно определить, есть

Малых А. Е. Разработка и реализация алгоритмов разрешения конфликтов по доступу к памяти в динамическом компиляторе Java для процессора «Эльбрус» // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 78–85.

ли зависимость между какой-либо парой инструкций, то необходимо предполагать ее наличие, чтобы не допустить ошибку выполнения программы. Одним из видов зависимости является *зависимость по памяти*, когда операции «загрузка-запись» и «запись-запись» могут указывать на один и тот же участок памяти в программе. В предыдущей реализации JVM для процессора «Эльбрус» любая пара вышеупомянутых обращений к памяти считалась зависимой, что отрицательно сказывалось на возможностях планирования [2]. Целью данной работы являлся анализ алгоритмов разрешения конфликтов по доступу к памяти и их реализация в JVM на процессоре «Эльбрус».

В работе анализируются существующие в настоящее время алгоритмы и способы их адаптации к JVM, а также описываются новые алгоритмы, разработанные с учетом особенностей JVM и процессора «Эльбрус». В заключение приводится сравнение производительности реализованных алгоритмов.

Обзор предметной области

Виртуальная Java-машина. Java – это типизированный объектно-ориентированный язык, разработанный компанией Sun Microsystems. Исходные коды на Java транслируются в .class-файлы, содержащие специальный байт-код, благодаря которому обеспечивается кросс-платформенность, – такие файлы могут быть исполнены на любой архитектуре, для которой реализована JVM.

Для исполнения байт-кода JVM на процессоре «Эльбрус» включает в себя интерпретатор и JIT-компилятор. Интерпретатор исполняет один байт-код за другим, не применяя при этом сложных оптимизаций, он используется для реализации методов, исполняющихся не очень часто. JIT-компилятор, в свою очередь, анализирует и оптимизирует сразу весь метод, в результате чего получается более оптимальный код. Так как сам процесс компиляции является ресурсоемким, он обычно применяется к часто исполняющимся методам [3].

Аппаратные возможности процессора «Эльбрус». Ключевая особенность процессоров «Эльбрус» состоит в том, что они построены на VLIW-архитектуре, позволяющей за один такт процессора выполнять сразу несколько инструкций. Эти инструкции исполняются параллельно, а распределение работ между ними задается во время компиляции. Такой подход существенно упрощает устройство процессора и позволяет увеличить количество вычислительных модулей, однако усложняет работу компилятора [4].

За один такт процессор выполняет одну *широкую команду*. Такая команда может содержать несколько операций сложения, вычитания, умножения, деления, операций загрузки и записи и др.

Планировщик инструкций. Задача планирования заключается в том, чтобы выбрать последовательность выполнения инструкций, минимизировав при этом сумму времени исполнения операций и времени простоя [5]. Для ее решения нужно разместить все инструкции в наименьшее количество широких команд, соблюдая при этом минимальные задержки по готовности результатов.

Ключевым понятием при планировании является определение *базового блока*. Базовый блок – это последовательность инструкций, имеющая одну точку входа, одну точку выхода и не содержащая инструкций передачи управления ранее точки выхода [6].

Один из вариантов реализации планировщика заключается в том, чтобы производить планирование инструкций только внутри базовых блоков. Проблема такого подхода заключается в том, что зачастую код содержит небольшие блоки, внутри которых недостаточное количество инструкций, для того чтобы выполнить эффективную упаковку кода.

Один из способов решения этой проблемы заключается в том, чтобы производить *суперблоковое планирование*. *Суперблок* – это последовательность базовых блоков, содержащая только одну точку входа и сколько угодно точек выхода [7]. Используемый в JIT-компиляторе суперблоковый планировщик был разработан с учетом особенностей JVM для процессора «Эльбрус» [1].

Суперблоковый планировщик – это итеративный алгоритм, каждая итерация которого выглядит следующим образом:

- выбрать еще не запланированный суперблок;
- построить граф зависимостей между инструкциями в суперблоке;

- выбрать порядок выполнения инструкций и разместить их по широким командам.

Построение графа зависимостей. Одним из ключевых понятий алгоритмов планирования инструкций является граф зависимостей – это взвешенный ориентированный граф, описывающий зависимости между инструкциями [8]. Множество вершин такого графа совпадает со множеством инструкций, для которых производится планирование. Ребро (u, v) с весом w в графе указывает, что инструкция u должна быть исполнена раньше инструкции v хотя бы на w тактов. Если w равно 0, значит, инструкции u и v могут быть исполнены в одной широкой команде. Тем не менее надо учитывать, что в некоторых случаях порядок инструкций в широкой команде важен, и инструкция u все равно должна быть исполнена раньше инструкции v.

Можно выделить три ключевых вида зависимостей между инструкциями:

- *по данным* (возникают, когда операции чтения и записи значения в регистр должны следовать друг за другом в определенном порядке);
- *по памяти* (возникают, когда операции загрузки и записи в память должны следовать друг за другом в определенном порядке);
- *по управлению* (возникают, когда необходимо выполнить какую-то инструкцию гарантированно до или после перехода в другой блок).

Чем меньше ребер окажется в итоговом графе зависимостей, тем больше будет в результате возможностей для планирования.

Зависимости по данным определяются однозначно и не могут быть упрощены. Количество зависимостей по управлению может быть уменьшено путем вставки дополнительного кода, восстанавливающего состояния регистров на выходах из суперблока. На данный момент эта оптимизация уже реализована в JVM для «Эльбруса».

Таким образом, с точки зрения уменьшения количества зависимостей наибольший интерес представляют зависимости по памяти. Если удастся доказать, что две инструкции обращаются к разным участкам памяти, то можно говорить о том, что они независимы [9].

Алгоритмы разрешения конфликтов по доступу к памяти

Рассмотрим два основных подхода к анализу зависимостей по доступу к памяти: *статический* и *динамический*. Статический подход заключается в том, чтобы производить проверку зависимости двух операций обращения к памяти во время компиляции метода, и в соответствии с результатом проверки либо добавлять ребро в графе зависимостей, либо не добавлять. Идея динамического подхода, в свою очередь, заключается в том, чтобы производить такую проверку во время исполнения метода. В таком случае нам необходимо генерировать код для обоих исходов проверки [10].

Статический анализ по типам. Одной из ключевых особенностей языка Java является отсутствие указателей, благодаря чему представляется возможным выполнить анализ по типам – статический подход к анализу зависимостей по памяти при помощи сравнения типов загружаемых и записываемых элементов [11].

В высокоуровневом представлении компилятора хранится дополнительная информация об объектах, в частности о классах, к которым эти объекты относятся. Здесь важно отметить, что из-за наличия в языке Java виртуального полиморфизма на этапе компиляции мы не можем достоверно знать, принадлежит в действительности объект к данном классу или к одному из его наследников.

Передав информацию о классах объектов в низкоуровневое представление, мы получаем возможность определять, к каким классам относятся загружаемые и записываемые элементы на этапе построения графа зависимостей (Здесь неважно, является эта операция обращением к полю некоторого объекта или обращением к индексу массива. В первом случае речь будет идти о классе, к полю которого мы обращаемся, во втором – о классе массива.) Если класс объекта, из которого идет загрузка, является наследником или предком класса объекта, в который производится запись, то возможно, что эти объекты совпадают, и соответствующие операции могут указывать на один и тот же адрес. В этом случае нам необходимо использовать другие методы анализа зависимостей. Если же это не так, т. е. ни один из этих классов

не является наследником другого, то можно утверждать, что операции обращения к памяти являются независимыми, и не добавлять между ними ребро в графе зависимостей.

Также анализ особенностей языка Java показал, что в результате наследования классов типы полей объектов не могут быть изменены. Тем самым мы можем анализировать по типам операции чтения и записи не только на основе объектов, из которых происходит загрузка, но и на основе типов непосредственно загружаемых объектов (например, классов полей при обращении к полю объекта). В этом случае нам даже не нужно смотреть на иерархию классов, достаточно просто сравнивать, совпадает ли класс загружаемого объекта с классом записываемого. Если классы отличаются между собой, то операции гарантированно являются независимыми.

Динамический анализ внутри циклов. Самый простой способ сгенерировать динамическую проверку – это добавить блок, в котором производится сравнение на равенство двух адресов чтения и записи, и создать код для обоих возможных случаев выполнения программы – один для случая равенства адресов и другой для случая неравенства. При таком подходе надо учитывать, что проверка будет производиться во время исполнения программы, поэтому генерировать такие проверки для всех пар операций слишком ресурсоемко. Обычно проверки создаются только для наиболее «горячих» участков кода, а именно – для циклов. Если адреса в операциях обращения к памяти не зависят от номера итерации цикла, то можно сделать блок с проверкой перед циклом и создать два варианта цикла – оптимизированный, в котором все операции чтения и записи между собой не пересекаются, и неоптимизированный, в котором операции чтения и записи считаются пересекающимися, если обратное не было доказано при помощи статических алгоритмов разрешения конфликтов по доступу к памяти. Если в цикле есть несколько операций чтения и записи, то нужно добавить проверку для всех таких пар и делать переход на оптимизированную версию цикла только в случае, если никакие из них не являются пересекающимися.

Как правило, в реальных программах адреса операций чтения и записи в цикле зависят от номера итерации, поэтому для эффективной работы динамической проверки нужен более сложный анализ, для которого необходимо, чтобы цикл имел канонический вид, а именно:

- содержал ровно одну индуктивную переменную;
- индуктивная переменная изменялась на каждом шаге на некоторую константу, известную на этапе компиляции;
- не имел других точек входа кроме головы цикла;
- не имел других точек выхода кроме хвоста цикла.

В таких циклах инструкции обращения к памяти в массивах можно представить в виде

$$адрес = база + множитель \times индуктивная\ переменная + смещение.$$

Для получения такого представления необходимо определить, каким образом получается адрес обращения к памяти. Для этого мы находим предыдущую операцию, в которой происходило определение регистра, соответствующего адресу, и в зависимости от типа операции изменяем базу, множитель и смещение. Далее процедура повторяется для базы, пока мы не дойдем до операции, из которой нельзя будет однозначно выразить новые значения базы и смещения (это может быть, например, загрузка из памяти или результат вызова функции). Когда такое представление будет получено для каждой инструкции обращения к памяти в цикле, мы можем проводить статический анализ для соответствующих операций, а именно: если для каких-то двух операций базы и множители совпадают, а смещения различаются, то мы можем утверждать, что они указывают на разную память. Несовершенство такого подхода заключается в том, что часто базы, с которых происходит загрузка, различаются (например, два массива передаются в функцию в качестве аргументов), и в этом случае мы ничего не можем сказать о зависимости между ними. Для преодоления этой проблемы перед циклом выполняется динамическая проверка. Мы составляем список используемых в цикле баз массивов, а перед циклом вставляем дополнительный проверочный блок. В этом блоке сравниваются между собой базы используемых в цикле массивов, и если базы всех массивов отличаются друг от друга, то делается переход на предварительно сгенерированную оптимизированную версию цикла. В противном случае делается переход на неоптимизированную

версию [12]. Преимущество описанного подхода состоит в том, что при планировании нашей оптимизированной версии цикла мы можем полагать, что никакие базы между собой не пересекаются, и, соответственно, если адреса двух инструкций определяются через две разных базы, то они гарантированно являются независимыми.

Динамический анализ с использованием аппаратных возможностей процессора «Эльбрус». Архитектура процессора «Эльбрус» предоставляет специальное устройство для динамического разрешения зависимостей по памяти. Принцип работы устройства основывается на разбиении начальной инструкции чтения на две: *preload* и *check*. Обе инструкции содержат три аргумента, два из которых определяют адрес загрузки, а третий обозначает регистр, в который будет записан результат чтения. Инструкция *preload* выполняет обычную загрузку из памяти и добавляет адрес, с которого произошла загрузка, в специальную таблицу процессора. Любая последующая операция записи с этого адреса удаляет соответствующую строчку из таблицы. Инструкция *check* проверяет наличие данного адреса в таблице. Если адрес в таблице есть, то значение загруженного ранее регистра актуально, и указанный в качестве аргумента адрес просто удаляется из таблицы. Если же такого адреса в таблице нет, значит, данные по нему были перезаписаны, в таком случае загрузка производится заново, как при обычной операции чтения. Кроме того, в последнем случае выставляется специальный флаг, по которому впоследствии это можно сделать. Благодаря этому мы можем выносить за инструкции записи не только загрузку, но и зависимые от нее операции, и если эти зависимые инструкции были произведены с «неправильным» значением загруженного регистра, то мы можем исполнить их снова. Такие зависимые операции называются *компенсационным кодом*.

Для эффективного использования соответствующих команд «Эльбруса» необходимо внести определенные изменения в вышеприведенный алгоритм суперблокового планирования. Модифицированная версия планировщика выглядит следующим образом [13]:

- выбрать еще не запланированный суперблок;
- разбить каждую инструкцию чтения на *preload* и *check*;
- построить граф зависимостей, полагая инструкции *preload* независимыми по отношению к инструкциям записи;
- спланировать инструкции в выбранном суперблоке, убрав лишние инструкции *check*;
- добавить компенсационный код.

Несмотря на то что идея алгоритма достаточно простая, есть несколько важных моментов, которые нужно учитывать при реализации алгоритма. Особенность инструкции *check* на «Эльбрусе» заключается в том, что она не может быть спекулятивной, а значит, в результате должна остаться в исходном блоке, что необходимо отдельно учитывать при построении графа зависимостей. Кроме того, в JVM на «Эльбрусе» используются *неявные исключения*, т. е. может быть сделана загрузка с нулевого адреса и только потом произведена проверка на то, удалось ли сделать такую загрузку. Чтобы избежать загрузки с нулевого адреса, неявная проверка переносится между инструкциями *preload* и *check*. Также определяемый в инструкции *preload* регистр добавляется в *check* в качестве дополнительного аргумента, чтобы при распределении регистров между ними на то же место не был назначен другой регистр.

Была разработана эвристика, которая не дает выносить инструкции *preload* слишком высоко от начального места, это позволяет уменьшить количество срабатываний инструкции *check* и сократить давление на регистры, чтобы избежать ситуации, когда нам приходится выгружать часть регистрового файла на стек.

Чтобы корректно сгенерировать компенсационный код, для каждой операции *preload* необходимо поддерживать набор регистров, значения которых от нее зависят. Важно заметить, что нельзя переносить зависимости от использования к определению регистра за инструкцию *check*, так как в этом случае не получится восстановить начальные значения регистров для того, чтобы исполнить их снова.

Результаты

Алгоритмы, описанные в этой статье, были реализованы в динамическом компиляторе с языка Java для VLIW-процессора «Эльбрус». Компилятор с реализованными оптимизациями был проверен на следующих стандартных тестах:

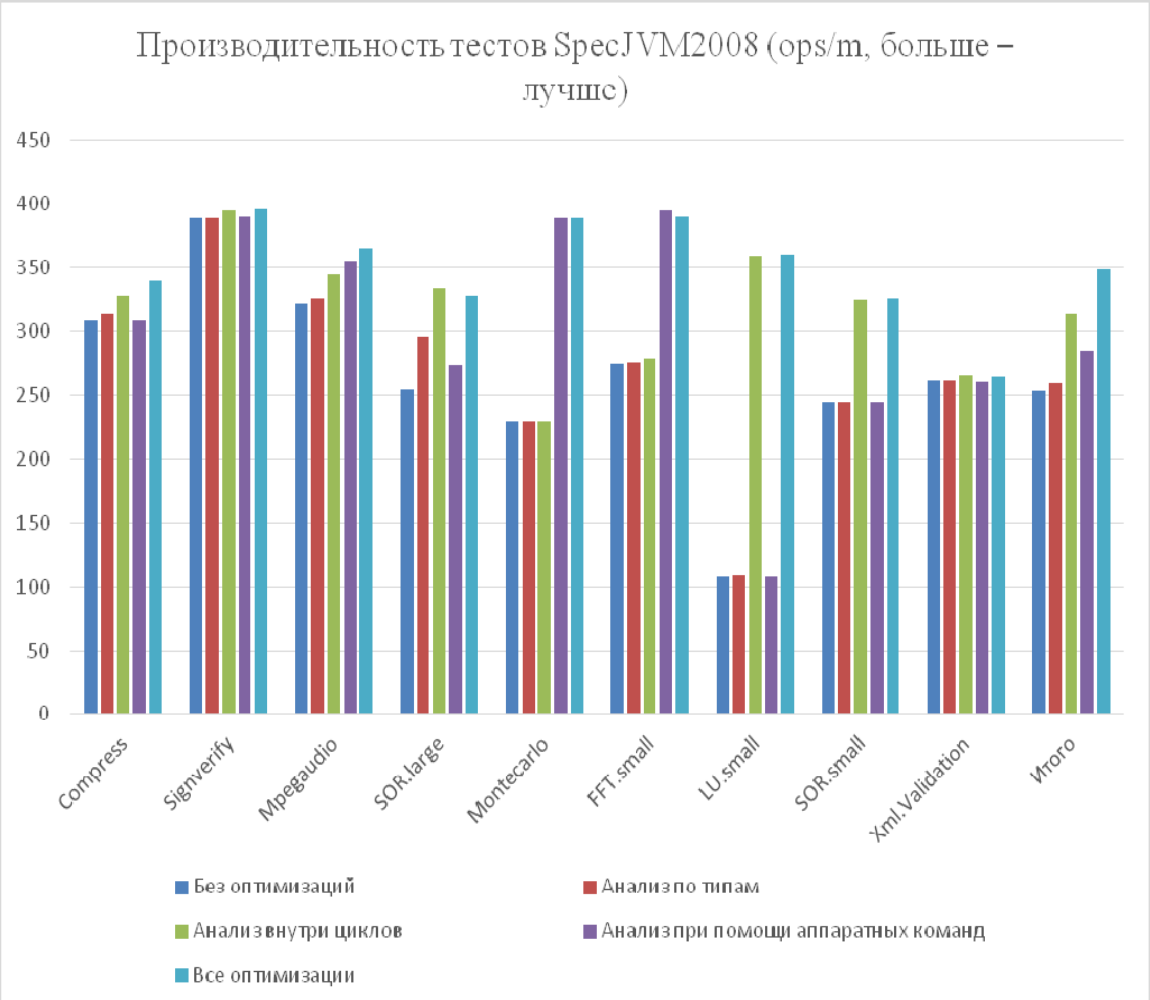
- SpecJVM98;
- SpecJVM2008;
- SpecJBB2005;
- Dacapo;
- JCK;
- JCTF.

Все тесты были пройдены успешно.

Также была произведена оценка полученного ускорения за счет реализованных оптимизаций на стандартной тестовой сюите SpecJVM2008. Тесты SpecJVM2008 включают в себя следующие бенчмарки:

- Compress – алгоритм сжатия LZW;
- Signverify – алгоритм подписи и проверки на основе протоколов MD5withRSA, SHA1withRSA, SHA1withDSA и SHA256withRSA;
- Mpegaudio – декодирование формата mp3;
- Scimrak SOR – метод релаксации Якоби;
- Scimrak Monte Carlo – интегрирование методом Монте-Карло;
- Scimrak LU – LU-разложение матрицы;
- Xml.Validation – валидация xml-документов по xml-схеме.

Следует заметить, что тесты Scimark (кроме Monte Carlo) в SpecJVM2008 включены в small и large версии. В первом случае все данные, которыми оперирует программа, помещаются в кеш процессора, во втором – размеры матриц и массивов подобраны таким образом, что их необходимо загружать из оперативной памяти. Результаты сравнения производительности приведены на рисунке:



Заключение

Статья является логическим продолжением работы [1] над улучшением показателей упаковки кода и, как следствие, ускорением работы JIT-компилятора Java на процессоре «Эльбрус». Разработанные и описанные в статье оптимизации открывают большие возможности для добавления ряда других важных улучшений компилятора. В частности, это конвейеризация циклов и динамическая проверка существования исключений в цикле, которые невозможны без реализации рассмотренных выше алгоритмов. Кроме того, планируется перенести описанные алгоритмы и на другие JIT-компиляторы для процессора «Эльбрус», а именно, C# и JavaScript.

Список литературы

1. *Андреенко С. А.* Межблоковое планирование инструкций в динамическом компиляторе java для VLIW-процессора. Новосибирск, 2016.

2. *Ghosh, Soumyadeep, Yongjun Park, and Arun Raman.* Enabling efficient alias speculation // ACM SIGPLAN Notices. 2015. Vol. 50. No. 5.

3. *Paleczny M., Vick C., Click C.* The Java HotSpot Server Compiler, Sun Microsystems // Java Virtual Machine Researchand Technology Symposium (JVM '01), 2001.

4. Микросхема интегральная 1891ВМ7Я (Система команд): Руководство программиста. ТВГИ.00742-01 33 01-1. Ч. 1. Общие сведения.

5. *Hwu, Wen-Mei W. et al.* The superblock: an effective technique for VLIW and superscalar compilation // Instruction-Level Parallelism. Springer, Boston, MA, 1993. P. 229–248.

6. *Mahlke, Scott A. et al.* Effective compiler support for predicated execution using the hyperblock // ACM SIGMICRO Newsletter. IEEE Computer Society Press, 1992. Vol. 23. No. 1–2.

7. *Faraboschi P., Fisher J. A., Young C.* Instruction scheduling for instruction level parallel processors // Proceedings of the IEEE. 2001. Vol. 89.11. P. 1638–1659.

8. *Yang T., Gerasoulis A.* List scheduling with and without communication delays // Parallel Computing. 1993. Vol. 19.12. P. 1321–1344.

9. *Huang A. S., Slavenburg G., Shen J. P.* Speculative disambiguation: A compilation technique for dynamic memory disambiguation // ACM SIGARCH Computer Architecture News. IEEE Computer Society Press, 1994. Vol. 22. No. 2.

10. *Maalej M. et al.* Pointer disambiguation via strict inequalities // Proceedings of the 2017 International Symposium on Code Generation and Optimization. IEEE Press, 2017.

11. Shpeisman T., Lueh G.-Y., Adl-Tabatabai A-R. Just-in-time Java compilation for the Itanium/spl reg/processor // Parallel Architectures and Compilation Techniques. Proceedings International Conference. IEEE, 2002.

12. *Péricles A. et al.* Runtime pointer disambiguation // ACM SIGPLAN Notices. 2015. Vol. 50. No. 10.

13. *Gallagher D., Chen W., Mahlke S., Gyllenhaal J., Hwu W.* Dynamic Memory Disambiguation Using the Memory ConflictBuffer // ASPLOS-VI Proceedings. Center for Reliable and High-Performance Computing, University of Illinois, 1994.

Материал поступил в редколлегию 17.03.2018

A. E. Malykh

Novosibirsk State University
1 Pirogov Str., Novosibirsk, 630090, Russian Federation

UNIPRO Ltd.
2 Lyapunov Str., Novosibirsk, 630090, Russian Federation

awa149@rambler.ru

DEVELOPMENT AND IMPLEMENTATION OF MEMORY DISAMBIGUATION ALGORITHMS IN DYNAMIC JAVA COMPILER FOR ELBRUS PROCESSOR

This article describes algorithms for memory disambiguation in dynamic Java compiler for Russian processor Elbrus with their implementation. These algorithms let significantly improve possibilities for instruction scheduler which is a key optimization for VLIW-processors. Static and dynamic approaches for memory disambiguation are described in this work. All implemented algorithms' efficiency is shown based on popular testing suite SpecJVM2008.

Keywords: Elbrus, Java, JIT-compiler, instruction scheduler, compiler optimizations.

References

1. Andreenko S. A. Superblock instruction scheduling in dynamic java compiler for VLIW processor. Novosibirsk, 2016. (in Russ.)

2. Ghosh, Soumyadeep, Yongjun Park, and Arun Raman. Enabling efficient alias speculation. *ACM SIGPLAN Notices*, 2015, vol. 50, no. 5.

3. Paleczny M., Vick C., Click C. The Java HotSpot Server Compiler, Sun Microsystems. *Java Virtual Machine Researchand Technology Symposium (JVM '01)*, 2001.

4. Integrated circuit 1891VM7Y, Programmers manual, (Instruction set), TVGI.00742-01 33 01-1. Part 1. General information. (in Russ.)

5. Hwu, Wen-Mei W. et al. The superblock: an effective technique for VLIW and superscalar compilation. *Instruction-Level Parallelism*. Springer, Boston, MA, 1993, p. 229–248.

6. Mahlke, Scott A. et al. Effective compiler support for predicated execution using the hyperblock. *ACM SIGMICRO Newsletter*. IEEE Computer Society Press, 1992, vol. 23, no. 1–2.

7. Faraboschi P., Fisher J. A., Young C. Instruction scheduling for instruction level parallel processors. *Proceedings of the IEEE*, 2001, vol. 89.11, p. 1638–1659.

8. Yang T., Gerasoulis A. List scheduling with and without communication delays. *Parallel Computing*, 1993, vol. 19.12, p. 1321–1344.

9. Huang A. S., Slavenburg G., Shen J. P. Speculative disambiguation: A compilation technique for dynamic memory disambiguation. *ACM SIGARCH Computer Architecture News*. IEEE Computer Society Press, 1994, vol. 22, no. 2.

10. Maalej M. et al. Pointer disambiguation via strict inequalities. *Proceedings of the 2017 International Symposium on Code Generation and Optimization*. IEEE Press, 2017.

11. Shpeisman T., Lueh G.-Y., Adl-Tabatabai A-R. Just-in-time Java compilation for the Itanium/spl reg/processor. *Parallel Architectures and Compilation Techniques. Proceedings International Conference*. IEEE, 2002.

12. Péricles A. et al. Runtime pointer disambiguation. *ACM SIGPLAN Notices*, 2015, vol. 50, no. 10.

13. Gallagher D., Chen W., Mahlke S., Gyllenhaal J., Hwu W. Dynamic Memory Disambiguation Using the Memory ConflictBuffer. *ASPLOS-VI Proceedings*. Center for Reliable and High-Performance Computing, University of Illinois, 1994.

For citation:
Malykh A. E. Development and Implementation of Memory Disambiguation Algorithms in Dynamic Java Compiler for Elbrus Processor. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 78–85. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-78-85

УДК 004.652; 004.9
DOI 10.25205/1818-7900-2018-16-2-86-94

С. Н. Трошков

Новосибирский государственный университет
ул. Пирогова, 1, Новосибирск, 630090, Россия

kamronis@yandex.ru

ОБ ОПЫТЕ МИГРАЦИИ ПРИЛОЖЕНИЙ НА СВОБОДНО РАСПРОСТРАНЯЕМОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ С ОТКРЫТЫМ КОДОМ

Важной проблемой современного программирования является поддержка и сопровождение наследственного программного обеспечения (ПО). Функциональность приложений, написанных в старых окружениях, ценна и попрежнему актуальна. Устаревшее ПО не позволяет использовать их на современных машинах и развивать в дальнейшем. В работе описан опыт миграции на примере двух приложений – Архива академика А. П. Ершова и системы «Библиотека», которые используются в ИСИ СО РАН не один десяток лет. В качестве платформы с открытым кодом для создания новых приложений был выбран CMF Drupal, который значительно облегчает разработку и перенос модели данных. Миграция включает в себя реинжиниринг приложения с сохранением бизнес-логики, модели данных, а также перенос самих данных.

Ключевые слова: миграция, база данных, модель данных, реинжиниринг, drupal, ms sql server, fox pro, mysql, postgresql.

Введение

Многие научные организации столкнулись с необходимостью поддержки большого количества информационных систем, разработанных в течение последних двух десятков лет с использованием проприетарного программного обеспечения. Такие системы, как правило, создавались с помощью грантов или с использованием спонсорской помощи компаний, предоставляющих программное обеспечение (ПО) окружения – серверные платформы, СУБД, системы программирования, интернет-серверы. Гранты закончились, и ресурсов на обновление ПО окружения больше нет, а приложения используются по сей день и работают на устаревшем оборудовании, в устаревшем окружении, в то время как своей актуальности они не потеряли и требуют дальнейшего развития и сопровождения.

В связи с этим становится актуальной задача миграции с проприетарного ПО на свободное. В данной статье будет описан метод такой миграции на опыте двух проектов – Архива академика А. П. Ершова и библиотечной системы «Библиотека».

Что такое миграция приложений и для чего она нужна?

Определим, что мы будем понимать под миграцией приложений. «Мигра́ция (от англ. *migration*) приложений – процедура перевода программных продуктов (исходного кода и структуры базы данных) с одной платформы (технологии) на другую (чаще всего из уста-

Трошков С. Н. Об опыте миграции приложений на свободно распространяемое программное обеспечение с открытым кодом // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 86–94.

ревшей в более современную)» [1]. В противоположность миграции данных из одного хранилища в другое (с одной аппаратной платформы на другую, из одной СУБД в другую), в данной статье речь пойдет именно о миграции приложения в целом.

Причин миграции может быть множество.

1. Экономическая причина. Стоимость лицензии от мировых лидеров – производителей самых популярных СУБД Oracle и Microsoft велика. А в связи с новой экономической политикой компаний продавать лицензии только на год, причем указывается стоимость лицензии на один процессор, т. е. в случае 4-процессорного сервера (обычная конфигурация) стоимость увеличивается в 4 раза, платить такую цену за СУБД под силу только крупным компаниям. Но, как показывает опыт, крупные компании тоже не всегда идут на это. Так, например, компания «Яндекс» в период с 2012 по 2016 г. планомерно мигрировала все свои сервисы на свободно распространяемую СУБД PostgreSQL [2].

Заметим, что ПО с открытым кодом не всегда бывает бесплатным – компании, предоставляющие собственные сборки, строят свой бизнес на продаже и поддержке собственных сборок. И все же стоимость такой поддержки не сравнима с ценами, установившимися в связи с новой лицензионной политики мировых брендов ПО.

2. Политическая причина. Зачастую именно политическая причина играет важную роль в принятии решения о миграции, если речь идет о приложениях, имеющих национальное значение. Уже упомянутая компания «Яндекс» приняла решение о миграции с Oracle еще в 2012 г., до объявления санкций и связанной с ними политикой импортозамещения [3]. Политическая ситуация была названа руководством одной из причин миграции.

3. Вопросы безопасности. Как ни странно, даже компании, предоставляющие решения в области безопасности, предпочитают использовать свободно распространяемое ПО с открытым кодом. Разработчиками ПО в open source (контрибуторами) являются и отдельные энтузиасты, и целые группы разработчиков, и компании. Политика же контрибуции модулей при разработке открытого ПО обычно такова, что модуль при добавлении подвергается многократному аудиту и ревизии, и если в нем присутствовал бы вредоносный код или просто уязвимость, они были бы выявлены на ранней стадии из-за широкого применения пользователями, которые одновременно являются и разработчиками.

Обнаруженные в процессе эксплуатации уязвимости исправляются очень быстро выпуском обновлений безопасности, зачастую в течение нескольких часов с момента обнаружения, в то время как в случае проприетарного ПО известны случаи, когда уязвимость не устранялась в течение длительного времени, пока о ней не становилось известно большому количеству пользователей.

Что касается ОС для серверных платформ, выбор открытого ПО (Ubuntu, CentOS, Debian, Fedora, FreeBSD) давно признан предпочтительным с точки зрения безопасности.

4. Выход свободного ПО на качественно новый уровень. Еще совсем недавно свободно распространяемые продукты, такие, например, как операционные системы Linux, использовались либо исключительно разработчиками ПО, либо только на серверных платформах. Сейчас благодаря Wine, Open Office, Linux Mint пользователи, не являющиеся специалистами в области IT, могут даже не заметить, что на их персональном компьютере установлена Linux, а не Windows.

Что касается платформ для разработки веб-приложений, многие системы, начинавшиеся двадцать лет назад как системы управления содержимым сайтов (CMS – Content Management System) переросли в платформы (frameworks) для разработки CMS. За двадцать лет они проделали путь от нехитрых конструкторов для построения сайтов до фреймворков, содержащих десятки тысяч библиотечных модулей с функциональностью, которая может потребоваться при разработке собственных CMS и веб-приложений.

5. Поддержка. При правильном выборе свободного ПО пользователь получает не только само ПО, но и бесплатную круглосуточную поддержку от членов сообщества, работающих с этим ПО. Поскольку круг пользователей очень широк, вопрос, который может возникнуть

[1] URL: http://wp.wiki-wiki.ru/wp/index.php/Миграция_приложений/.
[2] URL: http://www.tadviser.ru/index.php/Проект:Яндекс_(миграция_с_Oracle_на_PostgreSQL)/.
[3] *Друзягин Р.* Теория и практика миграции веб-систем на PostgreSQL. 2015. URL: https://habr.com/post/254667/.

у одного из них, скорее всего, уже возникал не у одного десятка пользователей, и на форумах сообщества найдется не только формулировка вопроса, но и решение проблемы.

6. Рефакторинг кода. Неотъемлемым свойством жизненного цикла приложения является развитие. По мере добавления все новых и новых возможностей, а также роста объема данных первоначально спроектированная архитектура приложения перестает удовлетворять современным требованиям. Приложение начинает работать медленно, понижается его отказоустойчивость. Миграция приложения является хорошим поводом для пересмотра функциональных модулей системы, его архитектуры и рефакторинга кода.

Миграция – прихоть или вынужденная мера?

Миграция приложений, безусловно, требует затрат и человеческих ресурсов. Даже если речь идет о миграции на другую СУБД с сохранением кода приложения и структуры базы данных, это может стать долгим и трудоемким процессом [4], поскольку оказывается, что SQL-запросы и хранимые процедуры не переносятся «один в один» из СУБД в СУБД. Более того, даже если речь идет о миграции от версии к версии одной СУБД, задача может оказаться нелегкой и потребовать изучения бизнес-логики самого приложения и переписывания части его кода.

Если же в рамках миграции требуется также и полное переписывание кода приложения в другую технологию, процесс может растянуться на годы. Поэтому миграция приложения – это, прежде всего, вынужденная мера, на которую приходится идти, чтобы сохранить возможность дальнейшей эксплуатации приложения.

Но, как мы покажем далее, даже миграция с полным переписыванием кода вручную требует значительно меньших трудозатрат, чем требовалось на этапе создания исходного приложения. Очевидно, объясняется это тем, что при миграции мы имеем дело с уже сформулированной бизнес-логикой приложения и готовой моделью данных, к тому же проверенными годами эксплуатации. Кроме того, тщательный подход к выбору платформы для миграции позволяет использовать богатый арсенал готовых библиотечных модулей для воссоздания функциональности приложения на новой платформе. И здесь самое время рассказать о выборе платформы Drupal.

Платформа Drupal

Для осуществления миграции была выбрана свободно распространяемая платформа Drupal [5], которая активно развивается с 2001 г. Первоначально развиваемая как CMS, впоследствии Drupal стала позиционироваться как CMF (framework – платформа). По данным на январь 2018 г., Drupal включает в себя более 39 000 модулей, что сильно ускоряет разработку приложений. Drupal поддерживает все популярные СУБД, отлично справляется с большими проектами, хорошо документирован. Платформа Drupal поддерживается широким сообществом разработчиков в мире (и в России в том числе).

Кроме того, выбор Drupal был также обусловлен тем, что в Институте систем информатики имени А. П. Ершова СО РАН уже был накоплен значительный опыт работы с этой платформой. В период с 2005 г. по настоящее время сотрудниками института реализовано около 30 веб-проектов на этой платформе.

Из недостатков можно отметить подход Drupal к структуре базы данных. В базе данных Drupal хранятся не только данные приложения, но и сама система управления приложением, поэтому создавать структуру базы данных произвольно нельзя – она формируется модулями ядра Drupal и модулями третьих сторон. Работать с базой данных напрямую достаточно сложно (и не рекомендуется).

[4] *Бородин В.* История успеха Яндекс-почты. URL: https://pgday.ru/files/papers/61/2016.07.08 История успеха Яндекс.Почты.pdf

[5] Официальный сайт сообщества Drupal: https://www.drupal.org/.

Другим недостатком Drupal является то, что, несмотря на жесткую модерацию модулей, некоторые из них могут вести себя некорректно при определенных условиях либо больше не поддерживаются разработчиками.

Постановка задачи миграции

В рамках изучения методов миграции на свободное ПО требовалось выполнить миграцию приложений, разработанных несколько десятилетий назад в ИСИ СО РАН, а именно Архива академика А. П. Ершова [1; 2] и системы «Библиотека» [3]. Миграция включает в себя следующие работы, которые могут быть выполнены как поэтапно, так и параллельно:

1) воссоздание структуры приложения с сохранением (по возможности) модели данных;
2) воссоздание бизнес-логики приложения для различных ролей пользователей, включая неавторизованных пользователей;
3) организация рабочих мест для авторизованных пользователей;
4) разработка дополнительных возможностей для повышения удобства использования приложений;
5) тестирование и отладка на тестовом пуле данных;
6) перенос данных на новое приложение.

Электронный архив академика А. П. Ершова

Электронный архив академика А. П. Ершова [4] – проект ИСИ СО РАН, выполненный при поддержке Microsoft Research. Проект был начат в 2000 г. и реализован в технологиях Microsoft (MS SQL Server, .Net, Microsoft Windows Server, IIS).

После кончины академика А. П. Ершова остался уникальный архив [5]. Это более 500 папок с документами, отражающими жизненный путь академика и историю развития информатики в России [6–9]. В настоящее время архив включает в себя следующие коллекции документов: архив А. П. Ершова, архив С. С. Лаврова, архив ИСИ СО РАН, архив ВНТК «Старт».

В архиве содержится следующая информация:

- документы – 42 386;
- изображения документов – 156 033;
- описанные персоналии – 6 431;
- сведения об организациях – 3 047.

В архиве реализованы два вида просмотра документов: папки и листы, как собирал их сам А. П. Ершов, и разбиение по темам и группам, как распределили их архивисты (рис. 1, 2).

Система «Библиотека»

Система «Библиотека» разработана Я. М. Курляндчиком в начале 1980-х гг. на БЭСМ-6, затем была перенесена на ПК с использованием средств MS DOS и FoxPro [10]. Система используется до настоящего времени в Мемориальной библиотеке А. П. Ершова для хранения, управления фондами, обработки и публикации новых поступлений в библиотеку. Система «Библиотека» представляет собой десктопное однопользовательское приложение [3]. И база данных, и само приложение находятся на одном компьютере, который представляет собой и рабочее место библиотекаря, и рабочее место читателя. Система написана не в архитектуре клиент-сервер. Это означает, в частности, что она не предоставляет возможности доступа к приложению с другого компьютера. Фактически, база данных представляет собой набор файлов, управляемых специально разработанной системой [11].

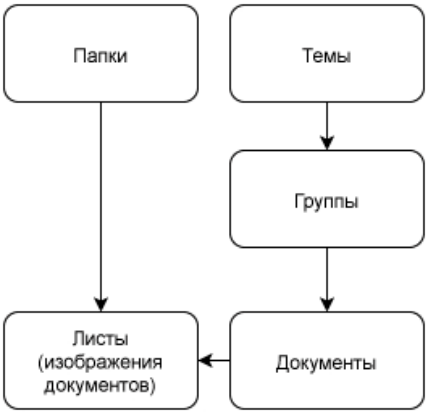


Рис. 1. Способы просмотра документов в архиве А. П. Ершова

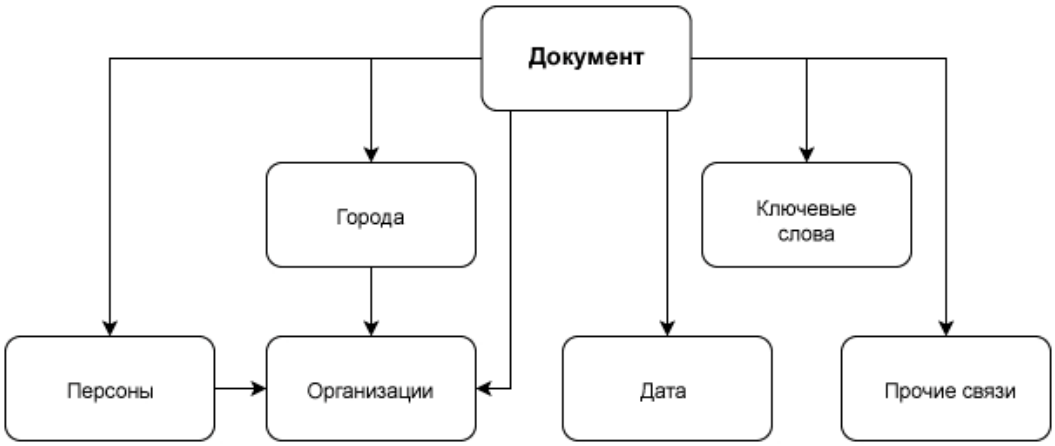


Рис. 2. Архив А. П. Ершова: связь с документами

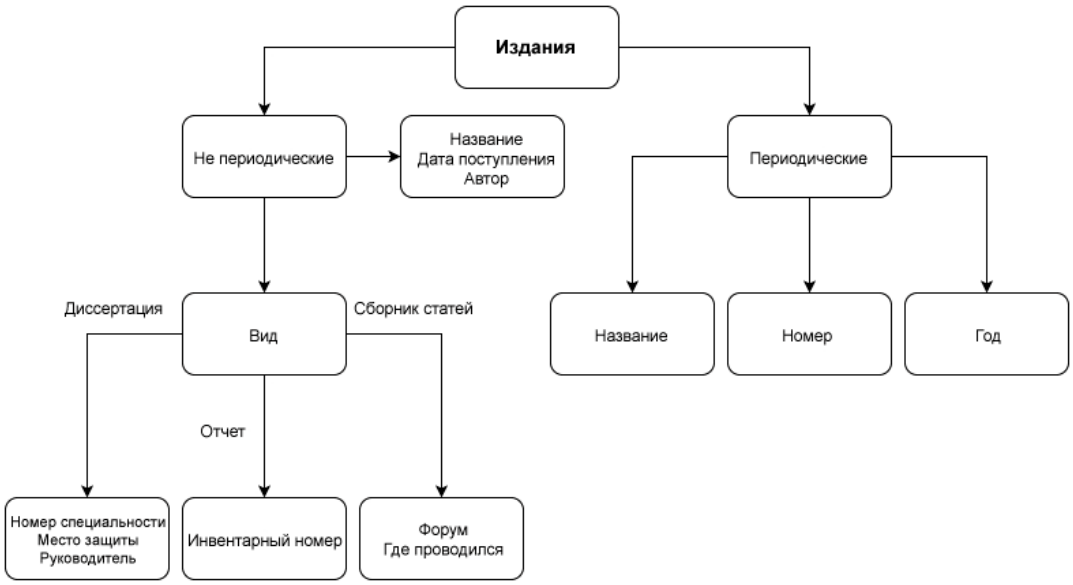


Рис. 3. Система «Библиотека»: модель данных

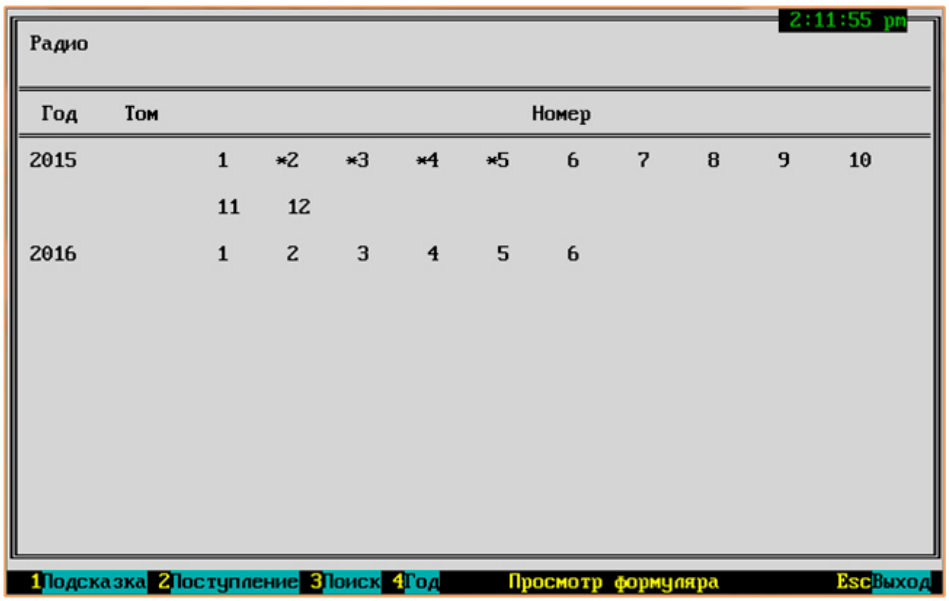


Рис. 4. Таблица номеров журнала «Радио»

В системе содержатся следующие данные:
- журналы – 722 (15 955 номеров);
- описания – 54 646;
- персоналии читателей – 114.

Объекты, с которыми работает Библиотека, – периодические (журналы) и непериодические (описания) издания (рис. 3). Каталог периодических изданий представляет собой список названий журналов. Выпуски журналов представлены таблицей, символом «*» помечены номера, находящиеся на руках (рис. 4). Описания хранятся в массивах по 1 000 элементов. Каждый массив – это файл с таблицей FoxPro. В описании указываются вид, язык, авторы, УДК, ББК, редактор, редакция, серия, год, название, количество страниц, источник, выпуск, том, ISSN и другие свойства описываемого фонда библиотеки.

Разработка приложения

Перед тем как приступить к миграции приложения «Библиотека», был изучен и обобщен опыт построения библиотечных систем [6].

Для сохранности данных приложения миграцию не следует проводить из работающего приложения. Непосредственно перед осуществлением миграции приложения пришлось осуществить аппаратную миграцию работающего приложения «как есть», а именно создать виртуальную машину с окружением, необходимым для запуска исходного приложения, затем скопировать данные приложения и запустить (эмулировать) приложение в современном окружении. С этой целью для приложения «Библиотека» была создана виртуальная машина, что позволило осуществить удаленный доступ к копии исходного приложения.

Для приложения «Архив академика А. П. Ершова» была скопирована база данных на виртуальную машину с СУБД MS SQL SERVER. Только после этого стало возможным приступить к первому этапу разработки приложения – воссозданию модели данных.

Требовалось проанализировать исходную модель данных и полностью воссоздать ее на платформе Drupal. Работа состояла из нескольких этапов:

1) создание типов сущностей;
2) создание словарей таксономии;
3) настройка связей между сущностями и терминами таксономий;
4) разграничение уровня доступа для пользователей.

В случае с системой «Библиотека» требовалось не просто воссоздать модель данных, но и формализовать ее, поскольку все данные в системе были строковыми и хранились нерационально.

Миграция данных в новое приложение

Как говорилось ранее, Drupal не позволяет напрямую работать с базой данных. При миграции каждой сущности для Drupal нужно указывать сразу все связи этой сущности с другими, после чего Drupal распределит их по таблицам в базе данных. Это очень удобно для пользователей приложений, чтобы заполнять данные о конкретной сущности, но вызывает сложности при миграции. По этой причине очень полезным оказался фреймворк Migrate, который позволяет писать собственные модули для миграции (рис. 5). Migrate поддерживает миграцию из всех популярных CMS и БД в Drupal, а также между версиями Drupal. Каждый модуль – это расширение класса Migrate на языке PHP.

Для каждого типа сущности был написан модуль, который преобразует извлеченные данные и записывает их в соответствующие поля сущностей Drupal. Миграции запускаются поочередно, итеративно, в зависимости от модели данных. Есть возможность тестирования миграций: включение миграции на небольшом объеме данных, остановка миграции, откат изменений, вызванных конкретной миграцией.

[6] Проект ELIBCONSULT: создание электронной библиотеки от проекта до реализации (http://www.elibconsult.ru/).

Рис. 5. Схема работы модуля Migrate

Подводные камни миграции

Во время миграции приложений пришлось столкнуться с определенными сложностями, которые были вызваны структурой исходных данных, а также особенностями самого метода миграции.

- В архиве А. П. Ершова данные, относящиеся к одной сущности, зачастую хранились в разных таблицах, такие таблицы приходилось отыскивать для каждой сущности и соединять операцией JOIN.
- В системе «Библиотека» все данные были строковые, т. е. не подразумевали использование словарей, типизации. Пришлось определить типы данных и на их основе составить словари для связей.
- В системе «Библиотека» после миграции возникло больше количество дубликатов сущностей. Дублировались сущности, которые библиотекарь вводил с ошибками или используя разные варианты написания и сокращения. После миграции были написаны скрипты на языке PHP, которые позволили объединить дубликаты сущностей и восстановить связи между ними.
- Платформа Drupal может работать только с кодировкой UTF-8. Архив А. П. Ершова работал в среде MS SQL, где использовалась стандартная кодировка Windows CP1251. В системе «Библиотека» использовалась DOS кодировка CP866. Были написаны функции, декодирующие данные из исходных кодировок в UTF-8 перед записью в базу данных.

Заключение

По итогам работы предложенный метод миграции был исследован и рекомендуется для использования в дальнейшем.

Благодаря поддержке словарей таксономии на уровне ядра Drupal отлично подходит для систем со сложной моделью данных, с большим количеством сущностей и связей между ними (архивы, библиотеки, каталоги). Ввиду высоких требований Drupal к аппаратным ресурсам следует с осторожностью принимать решение о применении предложенного метода для миграции высоконагруженных систем с большим количеством одновременных пользователей.

Предложенный метод показал весомое снижение трудозатрат на написание приложения. Исходное приложение «Архив академика А. П. Ершова» разрабатывалось командой из четырех разработчиков на протяжении нескольких лет, в то время как разработка на Drupal заняла несколько месяцев у одного разработчика. Дальнейшее ведение проектов архивист (либо библиотекарь) может осуществлять самостоятельно, не прибегая к помощи разработчика.

Поскольку платформа Drupal широко распространена и поддерживается мощным сообществом, в том числе в России, найти разработчика для поддержки, сопровождения и дальнейшего развития проекта не составляет проблемы.

В рамках миграции по каждому приложению были выполнены следующие задачи:

- воссоздана модель данных исходного приложения в окружении Drupal;
- разработаны веб-приложения с набором тестовых данных, создан интерфейс для пользователей и редакторов, настроены права доступа для пользователей;
- проведено тестирование новых приложений на выборке данных;
- разработаны механизмы для миграции данных исходных приложений в БД новых приложений с воспроизведением модели данных;
- осуществлен ряд итераций для миграции данных;
- приложение «Архив академика А. П. Ершова» находится в работе два года, доступно по адресу http://ershov.iis.nsk.su/;
- приложение «Библиотека» в настоящий момент находится в стадии тестовых испытаний.

Список литературы

1. *Филиппов В. Э. и др.* Электронный архив академика А. П. Ершова – методика создания и научной интерпретации // Информационные технологии в гуманитарных исследованиях. Новосибирск, 2006. Вып. 11. С. 51–56.

2. *Филиппов В. Э., Крайнева И. А., Филиппова М. Я., Черемных Н. А.* Интернет-технологии как средство сохранения и публикации материалов научного, культурного и исторического наследия на примере электронного архива академика А. П. Ершова. Улан-Удэ, 2003. С. 257–261.

3. *Курляндчик Я. М.* Система «БИБЛИОТЕКА». Руководство по использованию. Новосибирск, 1991.

4. *Antiufeev S., Boulyonkova A., Kraineva I., Nemov A.* Crreation and Scientific interliretation of Acad. Ershov's Electronic Archive // The Fourth SEEDI Conference «Digitization of Cultural and Scientific Heritage». Belgrade, Serbia, 2008. Book of Abstracts, li.11.

5. *Крайнева И. А., Черемных Н. А.* Личный архив академика А. П. Ершова в Интернете // Отечественные архивы. 2001. № 5. С. 53–55.

6. *Крайнева И. А., Черемных Н. А.* Научное наследие академика А. П. Ершова // Проблемы культурного наследия в области инженерной деятельности / Под ред. Г. Г. Григоряна. М.: Информ-Знание, 2007. Вып. 5. С. 140–172.

7. *Крайнева И. А.* Архив академика А. П. Ершова как источник по социальной истории научного сообщества // Документ в парадигме междисциплинарного подхода: Материалы II Всерос. науч.-практ. конф. Томск: ТГУ, 2006. С. 146–149.

8. *Крайнева И. А., Черемных Н. А.* Академик А. П. Ершов и его архив // Развитие вычислительной техники в России и странах бывшего СССР: история и перспективы (SORUCOM 2006): Материалы Междунар. конф. Петрозаводск, 2006. Ч. 2. С. 50–56.

9. *Крайнева И. А.* Научная биография академика А. П. Ершова: Автореф. дис. … канд. ист. наук: 07.00.10. Томск, 2008. 32 с.

10. *Курляндчик Г. В.* Судьба семьи в эпоху компьютеров // Sorucom-2014: Материалы Междунар. конф. Казань, 2014.

11. Курляндчик Я. М. Система управления файлами. М., 1980. 20 с. (Препринт / ИТМ и ВТ АН СССР; № 13).

Материал поступил в редколлегию 10.04.2018

S. N. Troshkov

Novosibirsk State University
1 Pirogov Str., Novosibirsk, 630090, Russian Federation

kamronis@yandex.ru

ON EXPERIENCE IN MIGRATING APPLICATIONS TO THE FREELY DISTRIBUTABLE OPEN SOURCE SOFTWARE

An important problem of modern programming is the support and maintenance of legacy software. The functionality of applications written in older environments is valuable and still relevant.

Outdated software environments do not allow to use the applications on modern machines and prevents their further development.

The paper describes the experience of migration to open source software. The migration was performed for two applications: the Archive of academician A. P. Ershov and the Library system. These applications work and have been used in Ershov Institute of Informatics Systems of SB RAS for a number of years. CMF Drupal has been chosen as a free and open source platform for creating new applications. The advantages of CMF Drupal facilitates significantly the development of new application and migration of the data model. Migration included reengineering the application while preserving business logic, the data model, and migrating the data itself.

Keywords: migration, database, data model, reengineering, drupal, ms sql server, fox pro, mysql, postgresql.

References

1. Fillipov V. E. et al. Electronic archive of academician A. P. Ershov as the methodology of creation and scientific interpretation. *Information technologies in humanitarian researches*. Novosibirsk, 2006, vol. 11, p. 51–56. (in Russ.)

2. Filippov V. E., Kraineva I. A., Filippova M. Ya., Cheremnykh N. A. Internet technologies as means of preservation and publication of materials of scientific, cultural and historical heritage by the example of academician A. P. Ershov's electronic archive. Ulan-Ude, 2003, p. 257–261. (in Russ.)

3. Kurlyandchik Ya. M. «LIBRARY» system. User's Guide. Novosibirsk, 1991. (in Russ.)

4. Antiufeev S., Boulyonkova A., Kraineva I., Nemov A. Crreation and Scientific interliretation of Acad. Ershov’s Electronic Archive. *The Fourth SEEDI Conference «Digitization of Cultural and Scientific Heritage»*. Belgrade, Serbia, 2008. Book of Abstracts, li.11.

5. Kraineva I. A., Cheremnykh N. A. Personal archive of academician A. P. Ershov on the Internet. *National archives*, 2001, no. 5, p. 53–55. (in Russ.)

6. Kraineva I. A., Cheremnykh N. A. Scientific heritage of academician A. P. Ershov. *Problems of cultural heritage in the field of engineering*. Ed. by G. G. Grigoryan. Moscow, Inform-Znanie, 2007, vol. 5, p. 140–172. (in Russ.)

7. Kraineva I. A., Cheremnykh N. A. Archive of academician A. P. Ershov as a source on social history of scientific community. *Second Russian scientific-practical conference «Document in the paradigm of interdisciplinary approach».* Tomsk, TSU Press, 2006, p. 146–149. (in Russ.)

8. Krayneva I. A., Cheremnikh N. A. Academician A. P. Ershov and his archive. *Proceedings of the international conference «Development of computer technology in Russia and the former Soviet Union: history and prospects (SORUCOM 2006)»*. Petrozavodsk, 2006, pt 2, p. 50–56. (in Russ.)

9. Kraineva I. A. Scientific library of academician A. P. Ershov. Dis. … candidate of Historical Sciences: 07.00.10. Tomsk, 2008, 32 p. (in Russ.)

10. Kurlyandchik G. V. The fate of the family in the era of computers. *Proceedings of the international conference Sorucom*. Kazan, 2014. (in Russ.)

11. Kurlyandchik Ya. M. File management system. Moscow, 1980, 20 p. (Preprint / IPMCE AS USSR; № 13). (in Russ.)

For citation:

Troshkov S. N. On Expirience in Migrating Applications to the Freely Distributable Open Source Software. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 86–94. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-86-94

УДК 004.91:577.2.01
DOI 10.25205/1818-7900-2018-16-2-95-103

А. А. Цхай [1, 2], **С. В. Мурзинцев** [3]

[1] *Институт водных и экологических проблем СО РАН*
ул. Молодежная, *1*, *Барнаул*, *656038*, *Россия*

[2] *Алтайский государственный технический университет*
пр. Ленина, *46*, *Барнаул*, *656038*, *Россия*

[3] *Алтайский государственный университет*
пр. Ленина, *61*, *Барнаул*, *656049*, *Россия*

taa1956@mail.ru, *o.l00@yandex.ru*

ИСПОЛЬЗОВАНИЕ ГОРИЗОНТАЛЬНО МАСШТАБИРУЕМОЙ ИНФРАСТРУКТУРЫ ПРИ ПОИСКЕ СХОДСТВА В ГЕНОМНЫХ ДАННЫХ ЭКОСИСТЕМ

Рассмотрена проблема выявления генетического сходства при анализе баз данных (БД) геномов организмов. Такая проблема возникает с развитием методов метагеномики, сравнительной геномики, технологий высокопроизводительного секвенирования ДНК, а также инструментов оценки и прогнозирования состояния экосистем. Для быстрого сравнения геномов с целью выявления повторяющихся наборов нуклеотидов разработана специализированная компьютерная система. Из-за большого объема данных, возникающих при обработке исходной информации, осуществлен переход к нереляционным БД, как к более гибким и масштабируемым. В качестве основы подхода использованы распределенная нереляционная БД MongoDB и алгоритм обработки данных Winnowing. При использовании нереляционной БД для выявления генетического сходства предложен вариант представления отпечатков структурных вариаций геномов в виде «ключ – значение». Выполнена программная реализация разработанной модели. Проведены вычислительные эксперименты: 1) загрузка данных в БД с использованием одной и трех шард (серверов, где хранятся данные и осуществляются поиск и обработка информации); 2) поиск совпадений выбранных наборов нуклеотидов с БД геномов с использованием одной и трех шард; 3) расчет скорости поиска геномов в БД; 4) расчет скорости загрузки геномов в БД. Результатом экспериментов стало подтверждение возможности использования предложенного способа поиска генетического сходства. Продолжение работы может быть в направлениях: 1) решения задачи об определении момента, когда необходимо добавлять узел к кластеру при возрастании рассматриваемого количества выбранных наборов нуклеотидов и увеличении числа геномов в БД организмов; 2) практического наполнения создаваемой БД как можно большим количеством реальных геномов организмов; 3) исследования геномных нарушений с целью оценки вероятности генетических отклонений на этапе распознавания потенциально возможного неблагоприятного развития организма.

Ключевые слова: сравнение геномов, большие данные, нереляционные базы данных, алгоритмы поиска повторений, биоинформатика.

Введение

Изучение структурно-функциональной организации живых организмов продолжает оставаться актуальным направлением, развивающимся на стыке биологии и информатики [1]. В этой связи использование компьютерных технологий при исследовании геномов (см., например, [2–5]) получило широкое распространение в мире.

Цхай А. А., *Мурзинцев С. В.* Использование горизонтально масштабируемой инфраструктуры при поиске сходства в геномных данных экосистем // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 95–103.

В числе современных крупных геномных проектов - несколько БД: ENSEMBL [1] – совместный проект организаций EMBL-EBI (Германия) и Sanger Centre (Великобритания) с целью создания программной системы для автоматического создания аннотаций геномов эукариотов. Проект ENSEMBL ориентирован на соответствие следующим критериям: точный, автоматический анализ данных генома; аннотации, основанные на текущих, своевременно обновляемых данных; доступность полученных данных через Интернет. GenBank [2] – БД нуклеотидных последовательностей, которая поддерживается NCBI (Национальным центром биотехнологической информации США). Крупнейшая интегрированная поисковая система ENTREZ, которая создана и поддерживается NCBI, используется для анализа нуклеотидных и аминокислотных последовательностей, библиографии (PubMed), полных геномов (Genomes), а также трехмерных структур белков (MMDB). При этом поиск данных о ДНК и белках не ограничивается только ресурсами GenBank, но распространяется и на другие доступные по сети хранилища информации. International Nucleotide Sequence Database Collaboration объединяет три крупнейшие коллекции нуклеотидных последовательностей: EMBL-EBI и GenBank (NCBI) и DDBJ (Япония). Информационный ресурс KEGG (Kyoto Encyclopedia of Genes and Genomes) [3] создается Институтом химических исследований (Kyoto University, Japan). Эта база знаний имеет обширные возможности для работы со всеми крупными мировыми информационными ресурсами. Обновление KEGG происходит ежедневно.

Существует ряд геномных браузеров, в том числе: NCBI, UCSC Genome Browser [4], ENSEMBL. Эти браузеры используются для получения и визуализации детальной справочной информации о геномах. Разработанная же авторами специализированная система предназначена для обработки справочной информации, а именно для быстрого сравнения геномов организмов с целью выявления повторяющихся наборов нуклеотидов.

В связи с развитием технологий высокопроизводительного секвенирования ДНК продолжается рост объема геномной информации в мире. Таким образом, несмотря на развитие международных БД и браузеров, остается важной задача сравнения протяженных геномных последовательностей, процессинга данных, поиска совпадений.

К настоящему времени предложены специальные форматы геномных данных (например, FASTA [5]), для поиска сходств в определенных классах геномных последовательностей разработаны компьютерные средства (например, BLAST [6]), идет работа над переносом рабочих процессов аппаратной оптимизации быстрого поиска геномных данных на виртуальные мощности облаков [6].

Нельзя не упомянуть вклад российских специалистов в разработку компьютерных методов решения задач метагеномики, сравнительной геномики, определения полиморфизмов, скрининга мутаций, транскриптомного профилирования и т. д. (например, [7–9]).

Данное исследование посвящено решению задачи сравнения выбранных наборов нуклеотидов с геномами организмов в реально существующем и постоянно актуализирующемся информационном ресурсе. Для этого необходимо сравнить содержащиеся в БД геномы организмов {G} с выбранными наборами нуклеотидов {N} в виде символьных последовательностей произвольной длины.

В работе источником элементов используемой БД являлась KEGG GENOME, включающая расшифрованные представления (около пяти тысяч организмов), находящиеся в свободном доступе. В настоящее время объем данных этого информационного ресурса можно оценить приблизительно в пять ТБ, что представляет технический вызов для существующих вычислительных мощностей.

В связи с большим объемом данных, возникающим при обработке исходной информации, был осуществлен переход от реляционных БД к нереляционным как к более гибким и масс-

[1] ENSEMBL. URL: http://www.ensembl.org (дата обращения 23.04.2018).
[2] GenBank. URL: http://www.ncbi.nlm.nih.gov/GenBank/ GenBankOverview.html (дата обращения 23.04.2018).
[3] KEGG GENOME. URL: http://www.genome.jp/kegg/catalog/org_list.html (дата обращения 23.04.2018).
[4] UCSC Genome Browser. URL: https://genome.ucsc.edu/ (дата обращения 23.04.2018).
[5] What is FASTA format? URL: https://zhanglab.ccmb.med.umich.edu/FASTA/ (дата обращения 23.04.2018).
[6] Basic Local Alignement Search Tool (BLAST). URL: https://blast.ncbi.nlm.nih.gov/Blast.cgi/ (дата обращения 23.04.2018).

штабируемым. Такой путь решения подобных проблем был обоснован в работе [10], в том числе для поиска нуклеотидных полиморфизмов и сходства геномных последовательностей [11]. В этом случае необходим инструментарий, позволяющий осуществлять поиск в горизонтально масштабируемой информационной системе (далее ИС) с возрастающими ресурсами в процессе развития. Данная ИС должна характеризоваться достаточной «эластичностью», т. е. способностью к расширению, без существенных инвестиций в инфраструктуру ИС, а также доработку программного обеспечения и алгоритмов.

Один из вариантов решения данной задачи – это использование структуры хранения данных и разработка поискового механизма на основе средств нереляционной распределенной БД MongoDB и алгоритма Winnowing [12–14]. В статье представлены результаты, ориентированные на реализацию такого подхода для сравнения геномов и поиска отклонений в их структуре.

Схема распределенной информационной системы

Распределенная ИС, созданная в работе на основе использования семи серверов, включает нереляционную БД MongoDB. Три сервера (UbuntuSlaveA, UbuntuSlaveB, UbuntuSlaveC) отвечают за хранение данных. На серверах находятся две распределенные БД: база данных нуклеотидных последовательностей и БД организмов. Три сервера: UbuntuSlaveD, UbuntuSlaveE, UbuntuSlaveF – это управляющие сервера, отвечающие за запись данных и их хранение на серверах UbuntuSlaveA, UbuntuSlaveB, UbuntuSlaveC. Сервер UbuntuMaster – это управляющий сервер, который предназначен для управления всем кластером, созданным с использованием технологии MongoDB.

На рис. 1 представлен персональный компьютер пользователя, с которого осуществляются первоначальная загрузка данных для формирования БД геномов организмов и БД нуклеотидных последовательностей, а также последующие запросы к созданным БД.

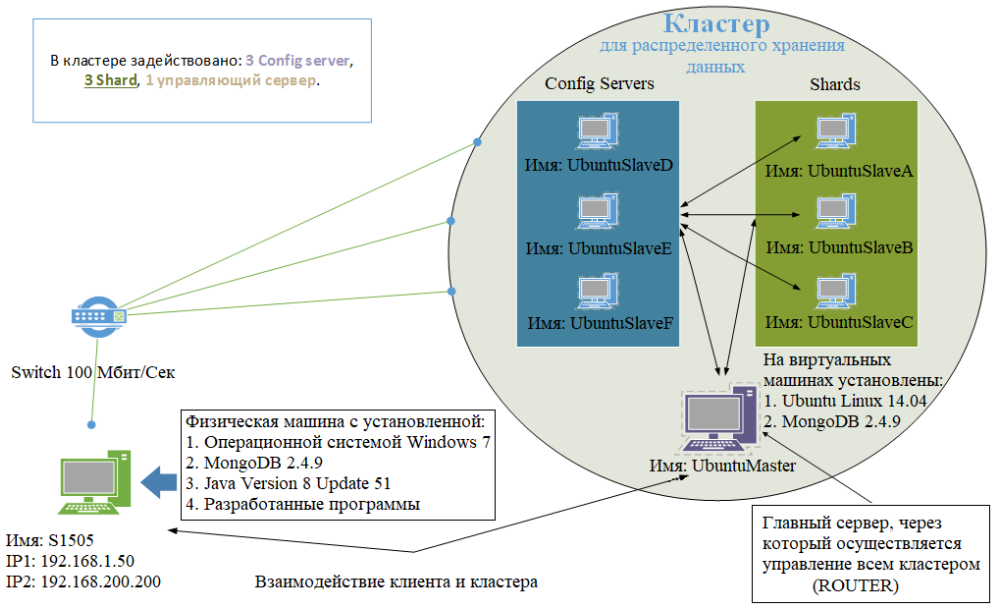


Рис. 1. Схема распределенной информационной системы для поиска выбранных наборов нуклеотидов

Данные о геномах, полученные из ENSEMBL, представляли собой наборы нуклеотидов, например, для мыши типа [CTAAAGTATA TATGAGTAAA CTTGGTCTGA CAGTTACCAA TGCTTAATCA GTGAGGCACC...] и т. п. В этом же виде они переносились в создаваемую в работе «вторичную» БД, со структурой, соответствующей описанной выше информационной системе.

Модель хранения геномов и их отклонений в нереляционной базе данных MongoDB

В исследованиях жизнедеятельности организмов, входящих в экосистемы различного уровня, используется представление о сходстве объектов (например, [15]). В литературе встречается достаточно много вариантов поиска сходства текстов, которые основаны на *n*-граммах (например, [16]). Методы поиска, использующие *n*-граммы, основаны на создании отпечатков документов, которые позволяют идентифицировать части при попарном сравнении символьных последовательностей. Подробный обзор сравнения алгоритмов с точки зрения производительности выполнен в работе [13].

Одним из алгоритмов, который основан на применении *n*-грамм, является алгоритм Winnowing [12]. Пример применения алгоритма Winnowing – для решения задачи сравнения текстов - рассмотрен в работе [14]. Показано, что выбранная *n*-грамма помещается в таблицу БД в виде триады *<хеш> – <документ> – <позиция хеша в документе>* и представляется записями как минимум в двух реляционно-связанных таблицах. Выполненный анализ демонстрирует высокую производительность при росте количества сравниваемых документов, но, как следствие, возникают скоростные ограничения в процессе использования реляционной модели.

В качестве решения проблемы производительности в работе предложена нереляционная модель типа *ключ – значение*, где хеш подстроки (*n*-граммы) документа выступает в качестве ключа, а значение представляет собой двумерную таблицу, содержащую значения *<документ> – <позиция хеша в документе>* для всех документов, имеющих совпадающую символьную последовательность. Таким образом, для поиска используется только одно ключевое значение, представленное значением хеш-функции от *n*-граммы. Благодаря этому возникает возможность выполнения параллельных запросов к нескольким узлам кластера одновременно.

Описанная модель была положена в основу структуры БД и протестирована в ряде экспериментов, результаты которых представлены ниже.

Экспериментальные результаты

Проведены четыре эксперимента по оценке следующих характеристик:

1) скорость загрузки геномов организмов и выбранных наборов нуклеотидов в БД MongoDB на одном компьютере в сравнении со случаем, когда БД распределена между тремя компьютерами в кластере;

2) зависимость скорости загрузки от размера выбранного набора нуклеотидов;

3) скорость загрузки геномов организмов на один компьютер в сравнении со случаем, когда БД распределена между тремя компьютерами в кластере;

4) скорость сравнения геномов организмов с БД выбранных наборов нуклеотидов на одном компьютере в сравнении со случаем, когда БД распределена между тремя компьютерами в кластере.

Применимость подхода оценивалась через два параметра. Первый параметр – это масштабируемость инфраструктуры, а второй – скорость работы. Скорость работы распределенной ИС оценивалась при сравнении выбранного набора нуклеотидов с геномами организмов по времени загрузки последних в БД. Объем данных, использованных в экспериментах, составлял порядка 20 Гб.

Оценка масштабирования инфраструктуры выполнялась при построении прототипа ИС с использованием различных конфигураций. Были протестированы конфигурации без разделения записей по индексу, а также с разделением записи по индексу между одной шардой и тремя. Шарды – это сервера, где хранятся данные и осуществляются поиск и обработка информации.

При тестировании ПО на всех конфигурациях не потребовалось изменения программного обеспечения, что позволяет утверждать, что прототип ИС может работать при разделении БД между неограниченным количеством рабочих станций. Данное свойство придает распределенной ИС эластичность при росте объема данных.

Оценка скорости загрузки

Первый эксперимент был предназначен для оценки скорости загрузки массива геномов организмов в БД. Данная операция не является критической для поисковой системы. Средняя скорость загрузки для различных конфигураций (мс/КБ текста): без шард – 76,2; 3 шарды – 17,9. Зависимость скорости загрузки от размера геномов организмов показана на графиках, которые представлены на верхней панели рис. 2. При загрузке данных производится обновление индексов. Производительность операции добавления данных зависит от объема уже загруженной информации. Из этих графиков видно, что время загрузки документов уменьшается в зависимости от количества шард в кластере.

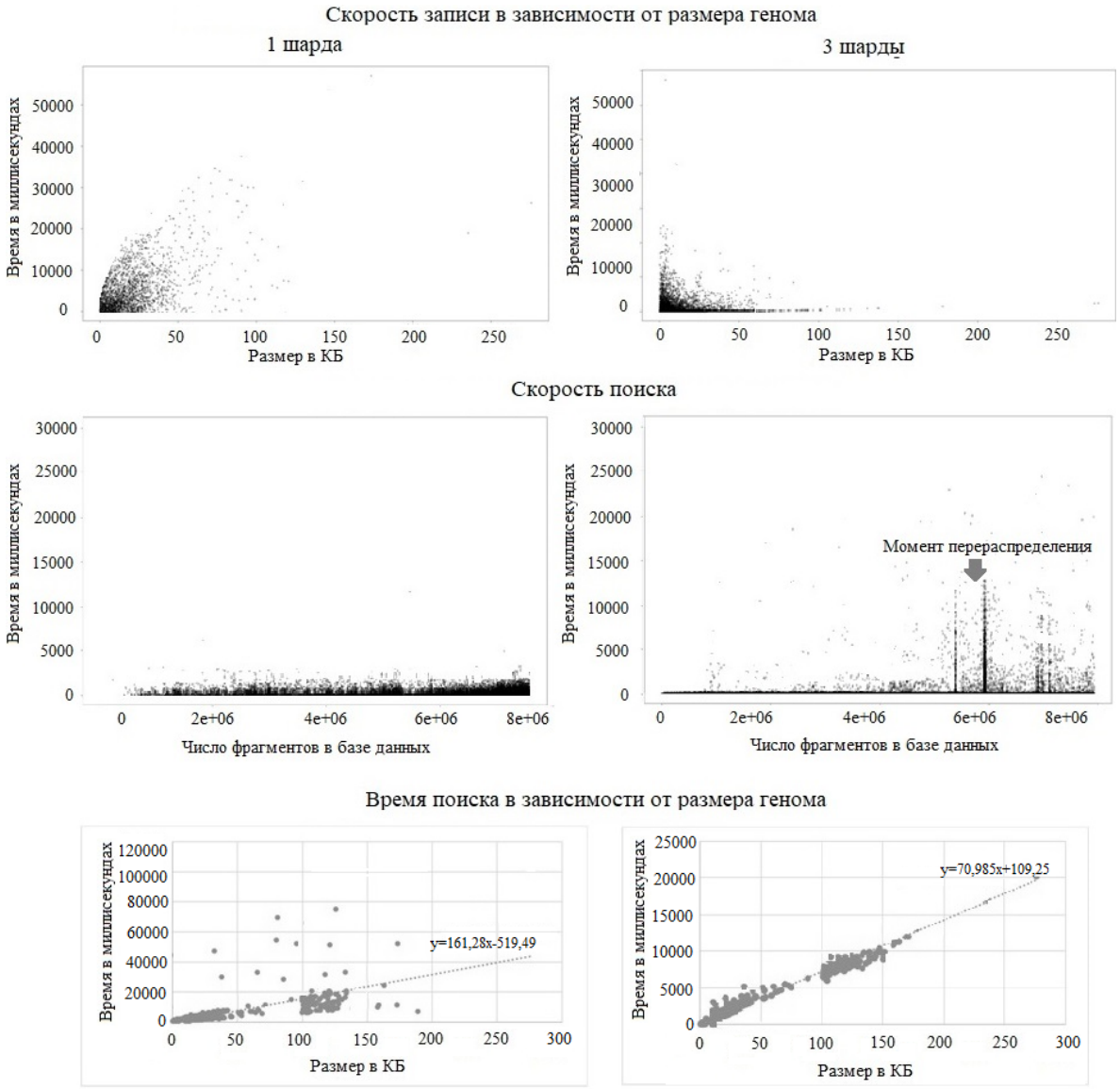


Рис. 2. Скорость поиска информации (одна и три шарды)

Из графиков нижней панели рис. 2 следует, что зависимость относительного увеличения времени, требуемого на загрузку, от объема уже загруженных геномов организмов близка к линейной. Также видно, что скорость поиска уменьшается. Отсюда можно сделать вывод, что скорость работы информационной системы с тремя шардами выше, чем с одной. Увеличение количества узлов хранения с одного до трех дает возможность оценки относительной

скорости загрузки, показывающей меньший рост при увеличении числа узлов хранения. Это позволяет утверждать, что с точки зрения загрузки данных система является горизонтально масштабируемой.

На графике, расположенном в средней панели рис. 2, видно, что в момент перераспределения индекса по шардам скорость загрузки геномов организмов в БД уменьшается (стрелкой показан момент перераспределения).

Оценка скорости поиска отклонений в геномах организмов

Целью второго эксперимента была оценка скорости поиска по БД геномов организмов. Для проведения оценки массив из геномов организмов был загружен в БД MongoDB, которая была установлена на загруженные ранее архитектуры. При осуществлении поиска случайным образом из списка всех рассматриваемых наборов нуклеотидов были выбраны некоторые с определенными нарушениями в порядке нуклеотидов и выполнено сравнение отобранных вариантов с геномами заполненной БД. Результатом поиска было совпадение набора хешей.

Эксперимент оценивал не качество, а скорость поиска. Графики зависимости времени поиска от размера приведены на рис. 2 (нижняя часть). Средняя скорость поиска (мс/КБ) составила: без шард – 121,8; 3 шарды – 72,3. Из этого можно сделать вывод о том, что средняя скорость загрузки без использования распределенной инфраструктуры как минимум в два раза выше, чем при хранении данных в распределенных узлах.

Результаты экспериментов свидетельствуют о принципиальной пригодности разработанной модели для поиска сходства в созданной БД, заполненной реальной информацией о геномах. Оценка скорости работы созданной модели, говорит о ее приемлемости с точки зрения производительности для поиска сходства геномных последовательностей организмов и, как следствие, дает возможность выявлять отклонения в развитии на ранних этапах диагностики.

Алгоритм, основанный на использовании *n*-грамм, в процессе экспериментов с созданной программной платформой показал достаточно хорошие результаты при поиске сходства наборов нуклеотидов в БД геномов.

Разработанная программная модель позволила протестировать алгоритм Winnowing и распределить «отпечатки» геномов организмов и выбранных наборов нуклеотидов по кластеру в нереляционной БД MongoDB. Разработанный программный комплекс позволил провести эксперименты, которые способствуют выработке новых стратегий и алгоритмов по улучшению поиска выбранных наборов нуклеотидов в геномах организмов.

Заключение

Оценим объем данных, которые возникнут в геномике при развитии постгеномных технологий секвенирования. Международный проект «1 000 геномов» уже привел к секвенированию порядка 100 000 индивидуальных геномов, и рост продолжается. Если оценить размер генома человека в 3 ГБ (без вспомогательной информации и аннотации), и население планеты в 7 миллиардов, то получим порядка 210 тысяч петабайт информации.

Рост геномной информации по секвенированию геномов лабораторных животных – крыс и мышей – как результат экспериментов в биомедицине, при тестировании фармпрепаратов, приводит к росту БД различных организмов. Так, например, следует отметить важность охарактеризованных ресурсов и системы быстрого поиска в них для развития средств мониторинга сообществ гидробионтов в водных экосистемах. Модели водных экосистем нового пятого поколения содержат в качестве основополагающих внутренних параметров геномные характеристики видов планктона [17].

Вышесказанное делает актуальным продолжение работы в нескольких направлениях.

Первое направление – это решение задачи об определении момента, когда необходимо добавлять узел к кластеру при возрастании рассматриваемого количества выбранных наборов нуклеотидов и увеличении числа геномов в БД организмов.

Второе – это практическое наполнение БД как можно большим количеством реальных геномов организмов. Использование полученных результатов в междисциплинарных геномных системных исследованиях позволило бы говорить о детализации и развитии модели в перспективном плане.

Третье - это исследование геномных нарушений с целью оценки вероятности генетических отклонений на этапе распознавания потенциально возможного неблагоприятного развития организма.

При индустриальном использовании БД геномов человека, животных и растений, в сотрудничестве со специалистами-генетиками, вышесказанное выглядит все реальнее с учетом продолжающегося «бума» исследований в данной области [18].

Благодарность. Авторы благодарны коллективу, выпускающему журнал, за внимание к работе и ценные советы.

Список литературы

1. Биоразнообразие и динамика экосистем: информационные технологии и моделирование / Отв. ред. В. К. Шумный, Ю. И. Шокин, Н. А. Колчанов, А. М. Федотов. Новосибирск: Изд-во СО РАН, 2006. 648 с.

2. *Lesk A. M.* Introduction to Genomics. 3rd ed. New York: Oxford University Press, 2017. 544 p.

3. *Dankar F. K., Ptitsyn A., Dankar S. K.* The development of large-scale de-identified biomedical databases in the age of genomics-principles and challenges // Hum. Genomics. 2018. Vol. 12 (1). P. 19. DOI 10.1186/s40246-018-0147-5.

4. *Langmead B., Nellore A.* Cloud computing for genomic data analysis and collaboration // Nat. Rev. Genet. 2018. Vol. 19 (4). P. 208–219. DOI 10.1038/nrg.2017.113.

5. *Nakagawa H., Fujita M.* Whole genome sequencing analysis for cancer genomics and precision medicine // Cancer Sci. 2018. Vol. 109 (3). P. 513–522. DOI 10.1111/cas.13505.

6. *Hong D., Rhie A., Park S. S., Lee J., Ju Y. S., Kim S. et al.* FX: an RNA-Seq. analysis tool on the cloud // Bioinformatics. 2012. Vol. 28. P. 721–723.

7. *Орлов Ю. Л., Брагин А. О., Медведева И. В., Гунбин И. В., Деменков П. С., Вишневский О. В., Левицкий В. Г., Ощепков В. Г., Подколодный Н. Л., Афонников Д. А., Гроссе И., Колчанов Н. А.* ICGenomics: программный комплекс анализа символьных последовательностей геномики // Вавиловский журнал генетики и селекции. 2012. Т. 16 (4/1). С. 732–741.

8. *Boekhorst R., Naumenko F. M., Orlova N. G., Galieva E. R., Spitsina A. M., Chadaeva I. V., Orlov Y. L., Abnizova I. I.* Computational problems of analysis of short next generation sequencing reads // Вавиловский журнал генетики и селекции. 2016. Т. 20 (6). С. 746–755. DOI 10.18699/VJ16.191.

9. *Спицина А. М., Орлов Ю. Л., Подколодная Н. Н., Свичкарев А. В., Дергилев А. И., Чен М., Кучин Н. В., Черных И. Г., Глинский Б. М.* Суперкомпьютерный анализ геномных и транскриптомных данных, полученных с помощью технологий высокопроизводительного секвенирования ДНК // Программные системы: теория и приложения. 2015. Т. 1 (24). С. 157–174.

10. Вычислительные методы, алгоритмы и аппаратурно-программный инструментарий параллельного моделирования природных процессов / М. Г. Курносов [и др.]; отв. ред. В. Г. Хорошевский; Рос. акад. наук, Сиб. отд-ние, Ин-т физики полупроводников им. А. В. Ржанова [и др.]. Новосибирск: Изд-во СО РАН, 2012. 335 с. (Интеграционные проекты СО РАН; вып. 33).

11. *Peise E., Fabregat-Traver D., Aulchenko Yu., Bientinesi P.* Algorithms for Large-scale Whole Genome Association Analysis. 2013. DOI 10.1145/2488551.2488577.

12. *Schleimer S., Wilkerson D., Aiken A.* Winnowing: Local Algorithms for Document Fingerprinting // International Conference on Management of Data (ACM SIGMOD. Proceedings). San Diego, 2003. P. 76–85.

13. *Faro S., Lecroq T.* The exact online string matching problem: A review of the most recent results // ACM Computing Surveys. 2013. Vol. 45, № 13. P. 42–50. http://dx.doi.org/10.1145/2431211.2431212.

14. *Цхай А. А., Бутаков С. В., Мурзинцев С. В., Ким Л. С.* Обнаружение плагиата с использованием нереляционных баз данных // Вестн. алтайской науки. 2015. № 1. С. 280–285.

15. *Федотов А. М., Чураев Р. Н.* О подходах к построению мер сходства между объектами // Математические модели эволюции и селекции: Сб. ст. Новосибирск, 1977. С. 120–131.

16. *Дягилев В. В., Цхай А. А., Бутаков С. В.* Архитектура сервиса определения плагиата, исключающая возможность нарушения авторских прав // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2011. Т. 9, № 3. С. 23–29.

17. *Jørgensen S. E.* Structurally dynamic models: a new promising model type // Environmental Earth Sciences. 2015. № 74. DOI 10.1007/s12665-015-4735-6.

18. *Park S. T., Kim J.* Trends in Next-Generation Sequencing and a New Era for Whole Genome Sequencing. // Int. Neurourol. J. 2016. Vol. 20, № 2. P. 76–83. http://doi.org/10.5213/inj.1632742.371

Материал поступил в редколлегию 28.02.2018

A. A. Tskhai [1, 2], **S. V. Murzintsev** [3]

[1] *Institute for Water and Environmental Sciences*
1 Molodezhnaya Str., *Barnaul*, *656038*, *Russian Federation*

[2] *I. I. Polzunov Altai State Technical University*
46 Lenin Ave., *Barnaul*, *656038*, *Russian Federation*

[3] *Altai State University*
61 Lenin Ave., *Barnaul*, *656049*, *Russian Federation*

taa1956@mail.ru, *o.l00@yandex.ru*

THE USE OF A HORIZONTALLY SCALABLE INFRASTRUCTURE IN THE SEARCH FOR GENETIC SIMILARITY IN BIODIVERSITY

The problem of rapid detection of genetic similarity in the analysis of databases (DB) of genomes of individuals of ecosystems at various levels is considered. The distributed non-relational DB MongoDB and the Winnowing data processing algorithm are used as the basis for creating the information system. Using a non-relational database to identify genetic similarity, a variant of representing the prints of the structural variations of the genomes in the form of «key-value» was proposed, a program implementation of the developed model was carried out, and computational experiments were carried out, which confirmed the possibility of using the proposed method of genetic similarity search, for example, in a personified analysis of deviations in the gene level.

Keywords: similarity of genomes, large data, nonrelational databases, search algorithms for repetitions.

References

1. Biodiversity and ecosystem dynamics: information technology and modeling / answering. Ed. V. C. Shumny, Yu. I. Shokin, N. A. Kolchanov, A. M. Fedotov. Novosibirsk, SB RAS Publishing House, 2006, 648 p. (in Russ.)

2. Lesk A. M. Introduction to Genomics. 3rd ed. New York: Oxford University Press, 2017, 544 p.

3. Dankar F. K., Ptitsyn A., Dankar S. K. The development of large-scale de-identified biomedical databases in the age of genomics-principles and challenges. *Hum. Genomics*, 2018, vol. 12 (1), p. 19. DOI 10.1186/s40246-018-0147-5.

4. Langmead B., Nellore A. Cloud computing for genomic data analysis and collaboration. *Nat. Rev. Genet.*, 2018, vol. 19 (4), p. 208–219. DOI 10.1038/nrg.2017.113.

5. Nakagawa H., Fujita M. Whole genome sequencing analysis for cancer genomics and precision medicine. *Cancer Sci*., 2018, vol. 109 (3), p. 513–522. DOI 10.1111/cas.13505.

6. Hong D., Rhie A., Park S. S., Lee J., Ju Y. S., Kim S. et al. FX: an RNA-Seq. analysis tool on the cloud. *Bioinformatics*, 2012, vol. 28, p. 721–723.

7. Orlov Yu. L., Bragin A. O., Medvedeva I. V., Gunbin I. V., Demenkov P. S., Vishnevsky O. V., Levitsky V. G., Oshchepkov D. G., Podkolodny N. L., Afonnikov D. A., Grosse I., Kolchanov N. A. ICGenomics: a program complex for analysis of symbol sequences in genomics. *Vavilov Journal of Genetics and Breeding*, 2012, vol. 16, p. 732–741 (in Russ.)

8. Boekhorst R., Naumenko F. M., Orlova N. G., Galieva E. R., Spitsina A. M., Chadaeva I. V., Orlov Y. L., Abnizova I. I. Computational problems of analysis of short next generation sequencing reads. *Vavilov Journal of Genetics and Breeding*, 2016, vol. 20 (6), p. 746–755. DOI 10.18699/VJ16.191.

9. Spitsina A. M., Orlov Yu. L., Svichkarev A. V., Dergilev A. I., Ming C., Kuchin N. V., Chernykh I. G., Glinskij B. M. Supercomputer analysis of genomics and transcriptomics data revealed by high-throughput DNA sequencing. *Program Systems: Theory and Applications*, 2015, vol. 6, p. 157–174. DOI 10.25209/2079-3316-2015-6-1-157-174. (in Russ.)

10. Computational methods, algorithms and hardware-software tools for parallel modeling of natural processes / M. G. Kurnosov [et al]; Resp. edited by V. G. Khoroshevsky; Russ. Akad. Sciences, Sib. Branch, In-t semiconductor physics named A. V. Rzhanov [et al.]. Novosibirsk, Publishing house of SB RAS, 2012,335 p. (Integration projects SB RAS; issue. 33) (in Russ.)

11. Peise E., Fabregat-Traver D., Aulchenko Yu., Bientinesi P. Algorithms for Large-scale Whole Genome Association Analysis. 2013. DOI 10.1145/2488551.2488577.

12. Schleimer S., Wilkerson D., Aiken A. Winnowing: Local Algorithms for Document Fingerprinting. *International Conference on Management of Data (ACM SIGMOD. Proceedings)*. San Diego, 2003, p. 76–85.

13. Faro S., Lecroq T. The exact online string matching problem: A review of the most recent results. *ACM Computing Surveys*, 2013, vol. 45, № 13, p. 42–50. http://dx.doi.org/10.1145/2431211.2431212.

14. Tskhai A. A., Butakov S. V., Murzincev S. V., Kim L. S. Obnaruzhenie plagiata s ispol'zovaniem nerelyatsionnykh baz dannykh [Plagiarism detection using non-relational databases]. *Vestnik altajskoj nauki*, 2015, no. 1, p. 280–285. (in Russ.)

15. Fedotov A. M., Churaev R. N. On approaches to the construction of measures of similarity between objects. *Mathematical models of evolution and selection*. Novosibirsk, 1977, p. 120–131. (in Russ.)

16. Dyagilev V. V., Tskhai A. A., Butakov S. V. Arkhitektura servisa opredeleniya plagiata, isklyuchayushhaya vozmozhnost' narusheniya avtorskikh prav [The architecture of the service definition of plagiarism, which excludes the possibility of copyright infringement]. Vestnik NSU. Series: Information Technologies, 2011, vol. 9, no. 3, p. 23–29. (in Russ.)

17. Jørgensen S. E. Structurally dynamic models: a new promising model type. *Environmental Earth Sciences*, 2015, no. 74. DOI 10.1007/s12665-015-4735-6.

18. Park S. T., Kim J. Trends in Next-Generation Sequencing and a New Era for Whole Genome Sequencing. *Int. Neurourol. J*., 2016, vol. 20, № 2, p. 76–83. http://doi.org/10.5213/inj.1632742.371

For citation:

Tskhai A. A., Murzintsev S. V. The Use of a Horizontally Scalable Infrastructure in the Search for Genetic Similarity in Biodiversity. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 95–103. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-95-103

УДК 550.37
DOI 10.25205/1818-7900-2018-16-2-104-112

М. В. Чубаров [1], **А. А. Власов** [2]

[1] *Новосибирский государственный университет*
ул. Пирогова, 1, Новосибирск, 630090, Россия

[2] *Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН*
пр. Академика Коптюга, 3, Новосибирск, 630090, Россия

m.chubarov@g.nsu.ru, vlasovaa@ipgg.sbras.ru

АВТОМАТИЗАЦИЯ ПОСТРОЕНИЯ ТРЕХМЕРНЫХ ГЕОЭЛЕКТРИЧЕСКИХ МОДЕЛЕЙ ДЛЯ МЕТОДА ЗОНДИРОВАНИЯ СТАНОВЛЕНИЕМ ПОЛЯ В БЛИЖНЕЙ ЗОНЕ НА ОСНОВЕ РЕЗУЛЬТАТОВ ОДНОМЕРНОЙ ИНВЕРСИИ

Предложен алгоритм автоматизации построения трехмерных геоэлектрических моделей для метода зондирования становлением поля в ближней зоне на основе результатов одномерной инверсии с целью расчета синтетических сигналов для трехмерных моделей, а также ускорения получения качественной оценки полевых материалов и сведения к минимуму ошибок интерпретации. Важной частью алгоритма является автоматическое формирование трехмерных расчетных сетей, необходимых для расчета синтетических сигналов в моделях. Результаты работы алгоритма представляют собой подготовленные трехмерные модели изучаемой среды с рассчитанным синтетическим электромагнитным сигналом. Алгоритм апробирован на данных электромагнитного мониторинга последствий землетрясения, произошедшего в 2003 г. в Республике Алтай.

Ключевые слова: ВЭЗ, ЗСБ, GMSH, Modem3D, Geo3dBuilder, построение трехмерных моделей, трехмерная модель, геоэлектрическая модель, зондирование становлением поля в ближней зоне, диаграммы Вороного, CGAL, EMS, HTCondor, Condor.

Введение

В настоящее время в области наземной геоэлектрики происходит усложнение объектов исследования, для решения научных и производственных задач недостаточно применять только одномерное моделирование сигналов становления электромагнитного поля в исследуемых средах. Причиной тому является возрастающее число необъяснимых или неверно истолкованных сигналов, в связи с чем возникла необходимость использовать более достоверный инструмент – трехмерное моделирование сигналов становления электромагнитного поля, позволяющий отслеживать влияние трехмерных эффектов.

В настоящее время уже существуют программные средства имитации сигналов в трехмерных средах – Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН использует для расчетов Modem3D [1], но остается непростой задача построения реалистичных трехмерных моделей. Для автоматизации построения таких моделей требуется хорошее стартовое приближение, в качестве которого в работе выступает трехмерная модель для выбранного пикета, соответствующая результатам одномерного моделирования для того же пикета. Процесс построения таких трехмерных моделей итеративный и занимает много времени, а также вызывает большое количество ошибок.

Чубаров М. В., *Власов А. А.* Автоматизация построения трехмерных геоэлектрических моделей для метода зондирования становлением поля в ближней зоне на основе результатов одномерной инверсии // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 104–112.

Целью работы является уменьшение времени построения сложных трехмерных моделей для метода зондирования становлением поля в ближней зоне и уменьшение ошибок построения путем автоматизации расчетов моделирования и исключения человеческого фактора.

Зондирование становлением поля в ближней зоне

Наземная электроразведка изучает геологическое строение земли, объединяя различные методы исследований объектов на основе их электрических и магнитных свойств. Сфера применения методов варьируется от поиска полезных ископаемых и картирования геологических разрезов до строительства, мониторинга дамб, а также изучения земных катастроф. На поверхности земли устанавливается аппаратура, регистрирующая изменения электрического и электромагнитного полей объектов. Полученные данные обрабатываются и интерпретируются в одномерную модель среды в каждой точке измерения.

Наиболее популярными методами наземной электроразведки являются вертикальное электрическое зондирование (ВЭЗ) и зондирование становлением поля в ближней зоне (ЗСБ). Исследования направлены на изучение геологических объектов, а также отслеживание их изменений в течение времени.

Метод ВЭЗ основан на измерении напряжения электрического поля, созданного путем установки разнесенных электродов, питающих это поле. Построение трехмерных моделей для данного метода было предложено в работе А. А. Сафиуллиной [2].

Другим ведущим методом наземной электроразведки является метод зондирования становлением поля в ближней зоне (ЗСБ). Установка (рис. 1) состоит из источника поля и приемников, представляющих собой незаземленные проволочные контуры, – генераторная и измерительная петли. При достаточно большом разнесении петель глубина исследования может достигать 10 км, что позволяет использовать метод при картировании местности, поиске рудных месторождений, газа и нефти, мониторинге геологических объектов.

Петли установки, используемые в исследовании, имеют форму квадрата и расположены соосно (совпадают центы координат петель). Измерительная петля находится внутри генераторной. Размеры петель зависят от размера исследуемой области. Чем больше размер генераторной петли, тем больший на нее подается ток, что напрямую влияет на увеличение глубины исследования. Кроме того, увеличение тока на генераторных петлях провоцирует увеличение количества вихревых токов Фуко, которые, в свою очередь, влияют на наводку измерительных петель. Все существующие факторы приводят к одной глобальной проблеме – недостаточной точности измерения сигналов. На генераторную петлю подается ток, затем выключается источник. В земле образуются вихревые токи Фуко, которые регистрируются измерительной петлей – зависимость напряжения от времени затухания. Полученные данные являются основой для трехмерного моделирования.

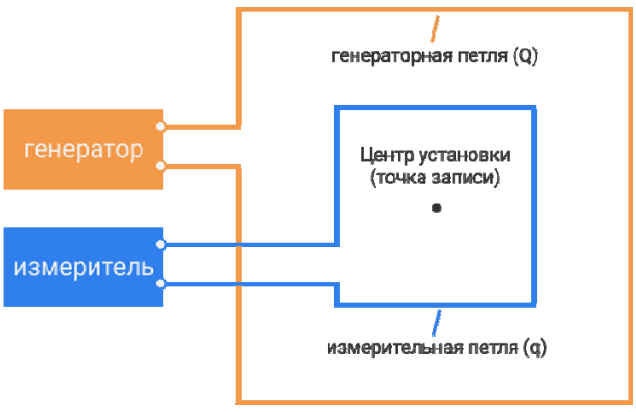


Рис. 1. Схема установки

Для метода ЗСБ предложен алгоритм автоматизации построения трехмерной расчетной сети с целью ускорения получения первого приближения моделей, сведения ошибок интерпретации к минимуму, а также расчета синтетических сигналов и их верификации.

Алгоритм построения трехмерных геоэлектрических моделей

Разработанный алгоритм для метода ЗСБ (рис. 2) является развитием алгоритма, предложенного в работе А. А. Сафиуллиной, А. А. Власова [2], и позволяет решить три принципиальные задачи: автоматическое построение первого приближения трехмерных моделей, составление для полученных моделей расчетной сети и оценка полученных результатов моделирования.

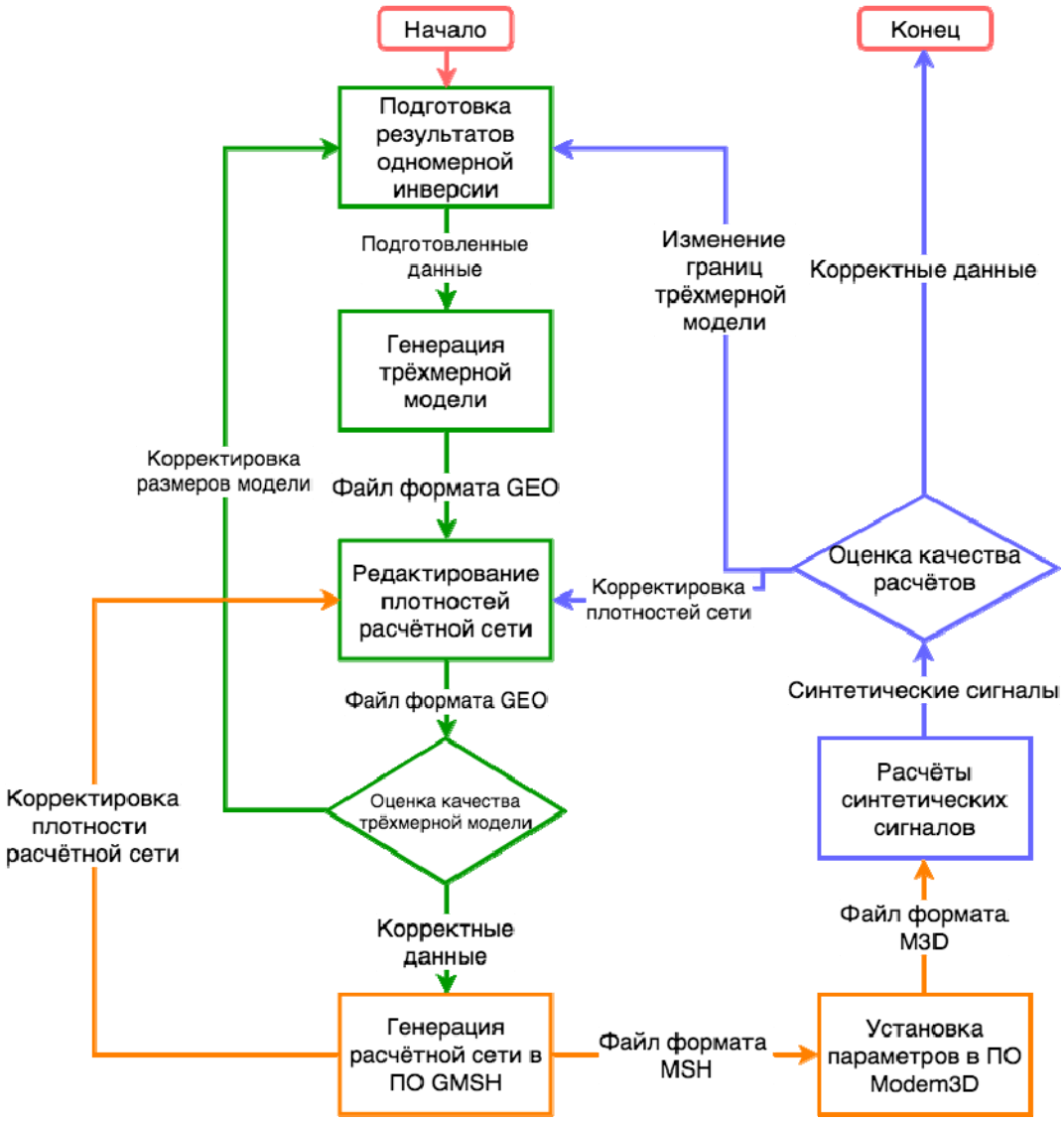


Рис. 2. Алгоритм автоматической генерации трехмерных моделей

Автоматическое построение трехмерных геоэлектрических моделей является важным этапом истолкования зарегистрированных сигналов ЗСБ, так как подготовка входных данных и построение из них трехмерных геоэлектрических моделей занимает у интерпретатора много времени. Разработанное программное средство Geo3DBuilder, использующее предложен-

ный алгоритм, позволяет уменьшить время построения геоэлектрических трехмерных моделей, а также предотвращает ошибки, появляющиеся во время их ручного создания.

Алгоритм апробирован на данных электромагнитного мониторинга последствий землетрясения, произошедшего в 2003 г. в Республике Алтай. Данные брались с установок в районе села Мухор-Тархата. обрабатывались и приводились в структурированный формат, описывающий будущую трехмерную модель.

Подготовленный файл является основой для работы алгоритма. Данные из алгоритма считываются и переводятся во внутреннее представление, на основе которого происходит автоматическая генерация трехмерных моделей. Их обрабатывает генератор трехмерных моделей, который создает двумерную модель, отбрасывая координаты z. Данная модель необходима для того, чтобы построить вокруг каждого пикета диаграммы Вороного [3] – области, ближайшие к этому пикету относительно остальных и принадлежащие ему, что позволяет облегчить дальнейшие расчеты синтетического сигнала, при этом не сильно проигрывая в точности. В работе использовались наработки библиотеки CGAL [1] – средства построения двумерных диаграмм Вороного.

Для удобства использования трехмерной модели, а также для дальнейших расчетов трехмерной сети в алгоритме используется Geo-разметка программного продукта GMSH [2]. Программный продукт GMSH позволяет использовать готовый синтаксис описания трехмерных моделей, а также содержит в себе средства генерации двумерных и трехмерных расчетных сетей, необходимых для дальнейшего моделирования синтетического сигнала. Алгоритм сначала формирует трехмерные точки, каждой присваивается уникальный индекс и координаты (x, y, z). Следует учесть, что точки с одинаковыми координатами объединяются в одну. Это необходимо для единой связности модели. В случае несвязности модели сигнал, проходящий в радиусе одного пикета, не зависит от сигнала другого пикета, что ничем не отличается от одномерного моделирования – моделирования, зависящего только от координат мощностей слоев. Затем из точек строятся линии, содержащие координаты начальной и конечной точек.

Полученная модель, состоящая из точек и линий, уже является трехмерной, но необходимо сформировать физические объемы, на основе которых будут производиться вычисления синтетических сигналов. Из полученных линий собираются контуры – набор линий, идущих одна за другой в определенном порядке и замыкающих друг друга. Важно учесть направление линий, так как несколько контуров могут содержать различающиеся направления, но при этом иметь общие линии. Такая задача решается добавлением отрицательного направления линии в конкретном контуре.

После получения контуров модели необходимо построить плоскости, содержащие контуры в основе, при этом учесть, что контуры могут совпадать в точности до линий, но иметь различные направления. Решение проблемы повторяющихся контуров приводит к избавлению от идентичных плоскостей и сохранению связности генерируемой модели.

Из полученных плоскостей строятся уникальные контуры объемов, содержащие набор плоскостей, затем из контуров объемов формируются логические объемы. На основе логических объемов формируются физические, необходимые для расчета синтетических сигналов в них.

Полученные трехмерные физические объемы являются финальной точкой генерации трехмерной модели (рис. 3). Время, необходимое для ручного описания геометрии трехмерной модели, содержащей 4 пикета, и подбора параметров расчетной сети, занимает от 20 часов. В свою очередь, автоматическое построение этой же модели занимает до 10 минут, что существенно экономит время и исключает ошибки при описании геометрии.

На следующем шаге полученная модель импортируется в ПО GMSH, где генерируется трехмерная расчетная сеть. Генерация производится встроенным в GMSH алгоритмом триангуляции Делоне, который позволяет создавать расчетную сеть, состоящую из тетраэдров. Плотность сети максимально высокая в центре координат исследуемого пикета и уменьшает-

[1] CGAL. Software project that provides easy access to efficient and reliable geometric algorithms in the form of a C++ library. URL: https://www.cgal.org/ (дата обращения 10.12.2017).

[2] GMSH. A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. URL: http://gmsh.info/ (дата обращения 21.11.2017).

ся в зависимости от увеличения радиуса области исследования. Рассчитывается при помощи полинома второй степени. Параметры полинома интерпретатор может редактировать на шаге создания сети. Для получения оптимальной расчетной сети рекомендуется использовать алгоритм оптимизации Netgen [3], но иногда он дает сбой, так как является экспериментальным. На выходе имеется модель, состоящая из тетраэдров, которая передается на вход в программное средство расчета синтетических сигналов Modem3D.

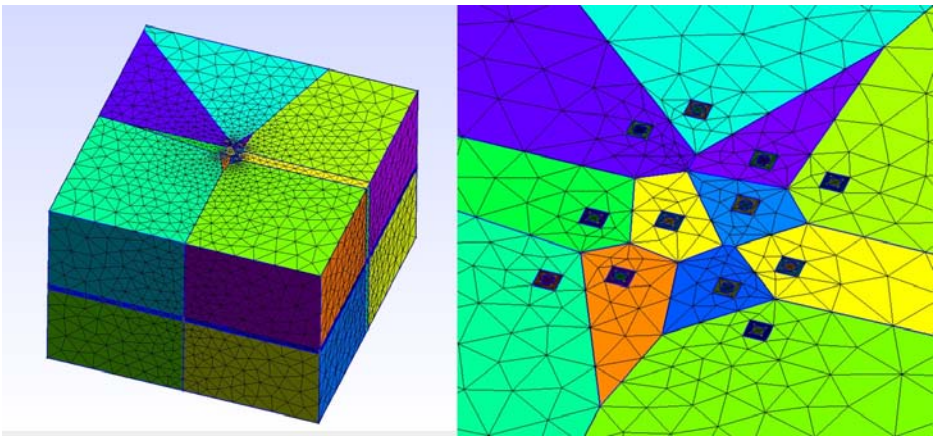


Рис. 3. Автоматически сгенерированная модель на 12 пикетов

Важной особенностью алгоритма является то, что интерпретатор на каждом шаге может изменять параметры как начальной модели, так и расчетной сети для интересуемой модели. Данная особенность алгоритма обусловлена тем, что он предлагает гибкий инструментарий для получения интересуемого результата.

Экспериментальные данные на 4 пикета

Генерация трехмерных моделей производилась на основе данных электромагнитного мониторинга последствий землетрясения на юго-востоке села Мухор-Тархата (рис. 4). Полигоном является участок 20 000 × 20 000 м, который содержит 4 пикета исследования. Каждый пикет представляют собой генераторную петлю размером 200 × 200 м и соосную измерительную петлю 100 × 100 м. Все петли одновитковые.

Пикеты 10, 31, 32 расположены на равнине, пикет 1 – в пойме притока реки Кокозек. Расстояние между пикетами варьируется от 500 до 1 000 м. На генераторные петли подавался ток в 1 А. В табл. 1 приведены данные одномерной инверсии для каждого исследуемого пикета.

Экспериментальные данные хорошо согласуются с синтетическим сигналом одномерных моделей в пределах метрологической погрешности измерительной аппаратуры, поэтому далее результаты трехмерного моделирования (трехмерная модель) сравниваются с синтетическими сигналами одномерного моделирования (рис. 5). Это позволяет сравнивать данные на более широком временном диапазоне и для трехмерного моделирования оценивать значения времени, при которых начинают влиять граничные условия.

[3] Netgen Mesh Generator. URL: https://sourceforge.net/p/netgen-mesher/git/ci/master/tree/ (дата обращения 27.11.2017).

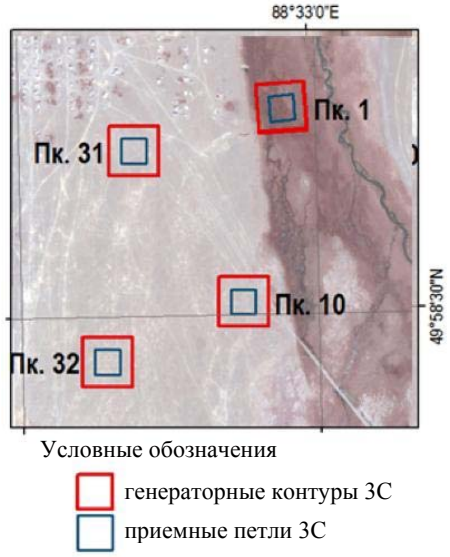


Рис. 4. Полигон Мухор-Тархата

Таблица 1

Результаты одномерной инверсии для 4 пикетов

Пикет 1		Пикет 10		Пикет 31		Пикет 32	
ρ	*h*	ρ	*h*	ρ	*h*	ρ	*h*
90	130	190	155	140	125	220	120
42	340	37	230	32	225	33	300
800		250		250		280	

Примечание: здесь и далее ρ – сопротивление (Ом), *h* – глубина (м).

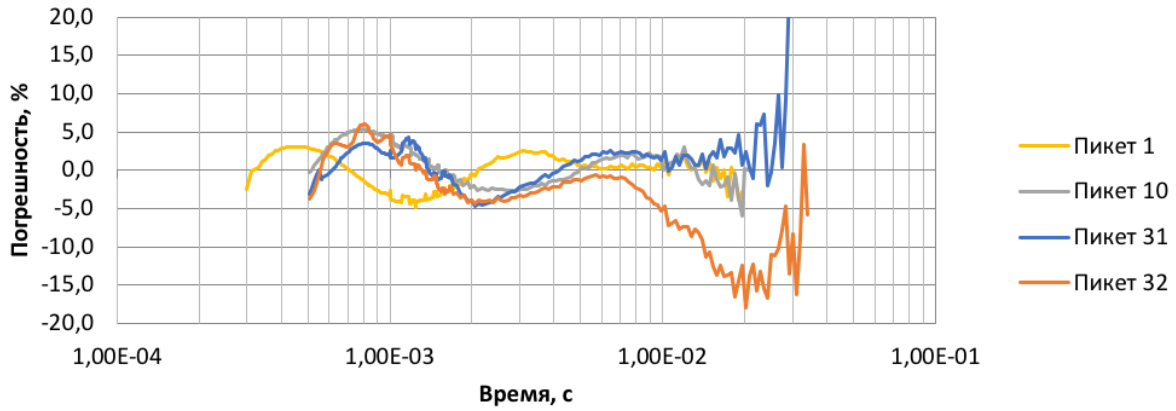


Рис. 5. Сравнение экспериментальных данных с синтетическим сигналом, вычисленным для одномерных моделей

Для генерации синтетического сигнала используется программное обеспечение Modem3D, использующее метод конечных элементов [1]. Для проверки достоверности результатов используется ПО EMS [4], которое позволяет производить моделирование синтетических сигналов ЗСБ для одномерных моделей. Расчетная сеть состояла из ~ 1,5 млн тетраэдров. Расчеты производились при помощи разработанного в ИНГГ СО РАН решателя для Modem3D – «MG PCG ||». Задача состояла из 10 линейных и 120 логарифмических итераций

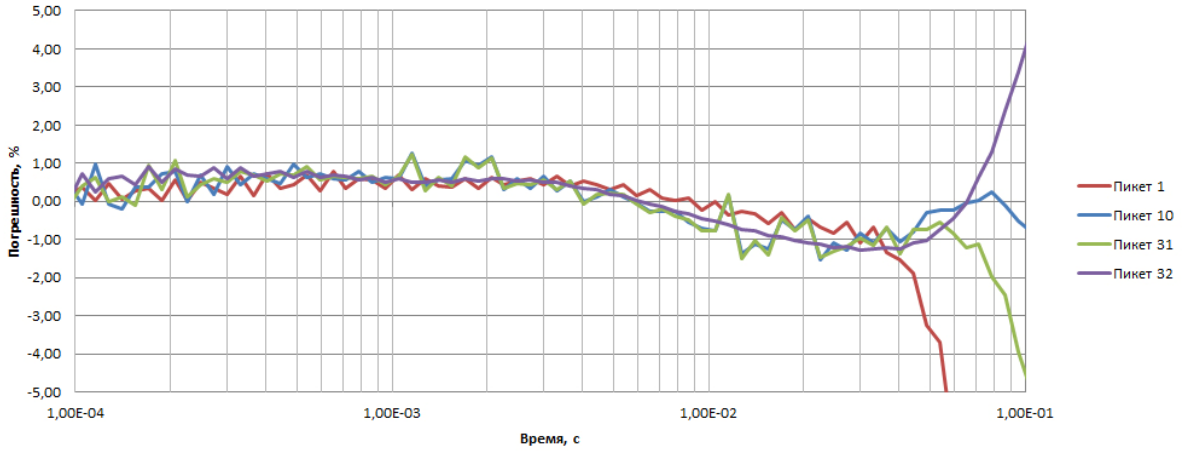


Рис. 6. Погрешности трехмерного моделирования для 4 пикетов

и решалась на ЭВМ с Intel Core i5 2-го поколения. Расчеты сигналов для каждого пикета в среднем занимали около 20 часов в двухпоточном режиме и расходовали до 2,4 GB оперативной памяти.

В результате сравнения результатов трехмерного и одномерного моделирования погрешность составила 1 % в интервале времени от 0,3 до 35 мс для одномерного моделирования в каждом пикете (рис. 6), что укладывается в метрологические характеристики измерительной аппаратуры – 5 %.

Экспериментальные данные на 12 пикетов

Для построения модели на 12 пикетов была расширена предыдущая модель на 4 пикета. В табл. 2 приведены данные одномерной инверсии для каждого исследуемого пикета.

Таблица 2

Результаты одномерной инверсии для 12 пикетов

Пикет 2		Пикет 4		Пикет 6		Пикет 11	
ρ	*h*	ρ	*h*	ρ	*h*	ρ	*h*
100	140	150	145	180	120	115	130
35	300	28	210	32	280	45	30
1500		2000		2000		2000	

Пикет 20		Пикет 21		Пикет 22		Пикет 30	
ρ	*h*	ρ	*h*	ρ	*h*	ρ	*h*
100	130	130	150	90	130	110	120
38	230	300	290	25	210	19	270
1500		2000		2000		2000	

Параметры установок и размер области исследования соответствовали модели на 4 пикета, так как имеют небольшой координатный разброс установок относительно друг друга. Модель состояла из ~2,2 млн тетраэдров. Задача решалась на той же системе 180 часов в двухпоточном режиме при расходе памяти до 6 GB. Для параллельного решения нескольких задач использовалась Grid-система HTCondor [4], позволяющей использовать ресурсы ЭВМ, подключенных в единую локальную сеть.

[4] HTCondor™. Specialized workload management system for compute-intensive jobs. URL: https://research.cs.wisc.edu/htcondor/ (дата обращения 10.12.2017).

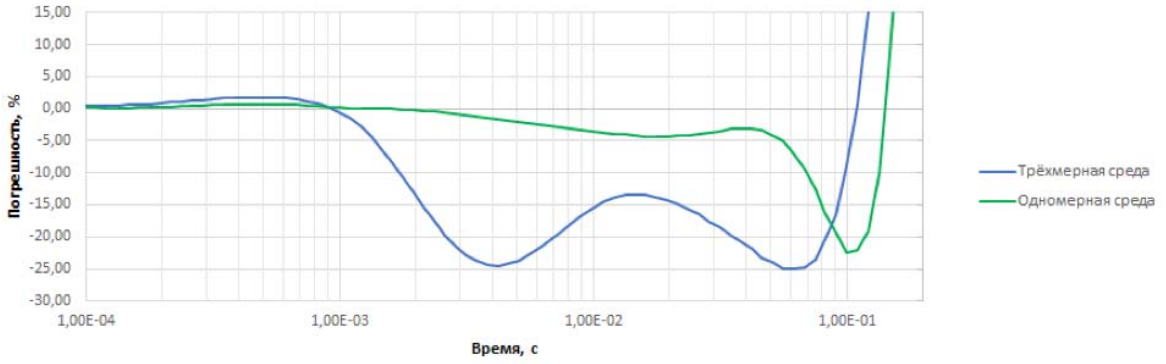


Рис. 7. Моделирование сигнала в модели на 12 пикетов

Полученные результаты (рис. 7) показывают отклонение сигнала со временем, что нельзя наблюдать при одномерном моделировании, так как плотности сеток приблизительно одинаковые как в одномерных, так и в трехмерных моделях. Расхождение сигналов объясняется влиянием параметров модели соседних пикетов. Этот факт свидетельствует о том, что неправильно подобраны одномерные модели, и необходимо их скорректировать, либо о том, что невозможно истолковать зарегистрированные сигналы в рамках одномерных моделей, – например, присутствуют наклонные границы пластов, или в пластах находятся посторонние объекты. Для корректного истолкования данных необходимо моделировать сигнал в более сложных двухмерных и трехмерных моделях.

Выводы

Разработанный алгоритм для метода зондирования становлением поля в ближней зоне позволяет в автоматическом режиме строить трехмерные геоэлектрические модели на основе результатов одномерной инверсии, тем самым ускоряя процесс получения оценки качества истолкования данных. Созданное программное средство Geo3DBuilder позволяет генерировать геометрию трехмерных моделей, а также подбор плотности расчетной сети. Интерпретатор использует Geo3DBuilder для настройки входных параметров модели, а уже в построенных моделях корректирует параметры расчетной сети. Применение алгоритма сводит к минимуму участие человека, что исключает появление ошибок интерпретатора, а также уменьшает время, необходимое на построения этих моделей.

Интерпретатор оценивает качество моделирования и подбирает параметры расчетной сети, изменяя коэффициенты полинома и размеры моделей. Кроме того, интерпретатор сам определяет, какой погрешности ему необходимо добиться и сколько он может уделить времени на расчеты.

Следующим шагом развития направления исследований будет получение моделей, более приближенных к реальным условиям. Эта цель будет достигаться путем создания моделей с наклонными границами, что позволит сделать более «плавные» переходы внутри одного пласта между пикетами.

Список литературы

1. *Иванов М. И., Катешов В. А., Кремер И. А., Эпов М. И.* Программное обеспечение модем 3D для интерпретации данных нестационарных зондирований с учетом эффектов вызванной поляризации // Записки Горного института. 2009. Т. 183. С. 242–245.

2. *Сафиуллина А. А., Власов А. А.* Автоматическое построение трехмерных геоэлектрических моделей по результатам одномерной интерпретации с помощью диаграмм Вороного // Интерэкспо ГЕО-Сибирь: Материалы Междунар. науч. конгресса. Новосибирск, 2016. С. 18–22.

3. *Карабцев С. Н., Стуколов С. В.* Построение диаграммы Вороного и определение границ области в методе естественных соседей // Вычислительные технологии. 2008. Т. 13. С. 65–80.

4. *Эпов М. И., Ельцов И. Н.* Прямые и обратные задачи индуктивной геоэлектрики в одномерных средах. Новосибирск: Изд-во ОГГиМ СО РАН, 1992. 31 с.

Материал поступил в редколлегию 14.03.2018

M. V. Chubarov [1], **A. A. Vlasov** [2]

[1] *Novosibirsk State University*
1 Pirogov Str., Novosibirsk, 630090, Russian Federation

[2] *Trofimuk Institute of Petroleum Geology and Geophysics SB RAS*
1 Acad. Koptyug Ave., Novosibirsk, 630090, Russian Federation

m.chubarov@g.nsu.ru, vlasovaa@ipgg.sbras.ru

AUTOMATION OF CONSTRUCTION OF THREE-DIMENSIONAL GEOELECTRIC MODELS FOR THE METHOD OF SOUNDING THE FORMATION OF THE FIELD IN THE NEAR ZONE BASED ON THE RESULTS OF ONE-DIMENSIONAL INVERSION

The article proposes an algorithm for automating the construction of three-dimensional geoelectric models for the method of sounding the formation of the field in the near zone based on the results of one-dimensional inversion in order to calculate synthetic signals for three-dimensional models, as well as to accelerate the production of qualitative evaluation of field materials, and to minimize interpretation errors. An important part of the algorithm is the automatic generation of three-dimensional computational networks necessary for the calculation of synthetic signals in models. The results of the algorithm are prepared three-dimensional models of the studied medium with a calculated synthetic electromagnetic signal. The algorithm is tested on the data of electromagnetic monitoring of the consequences of the earthquake that occurred in 2003 in the Altai Republic.

Keywords: VES, TEM, GMSH, Modem3D, Geo3dBuilder, building three-dimensional models, three-dimensional geoelectrical model of sounding in the near zone, Voronoi diagrams, CGAL, EMS, HTCondor, Condor.

References

1. Ivanov M. I., Kartashov, V. A., Kremer I. A., Epov M. I. Software modem 3D for data interpretation of transient soundings account for the effects of induced polarization. *Journal of the Proceedings of the Mining Institute*, 2009, vol. 183, p. 242–245. (in Russ.)

2. Safiullina A. A., Vlasov A. A. Automatic construction of three-dimensional geoelectric models based on the results of one-dimensional interpretation using Voronoi diagrams. *International Scientific Congress «Interexpo GEO – Siberia»*. Novosibirsk, 2016, p. 18–22. (in Russ.)

3. Karabtsev S. N., Stukalov S. V. Construction of Voronoi diagrams and defining the boundaries of the region in the natural neighbours method. *Computational Technologies*, 2008, vol. 13, p. 65–80. (in Russ.)

4. Epov M. I., El'tsov I. N. Direct and inverse problems of geoelectrics inductive in one-dimensional environments. Novosibirsk, OIGGM SB RAS Publ., 1992, 31 p. (in Russ.)

For citation:

Chubarov M. V., Vlasov A. A. Automation of Construction of Three-Dimensional Geoelectric Models for the Method of Sounding the Formation of the Field in the Near Zone Based on the Results of One-Dimensional Inversion. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 104–112. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-104-112

УДК 004.89
DOI 10.25205/1818-7900-2018-16-2-113-121

Г. Э. Яхъяева, **А. Р. Абсайдульева**

Новосибирский государственный университет
ул. Пирогова, 1, Новосибирск, 630090, Россия

gulnara@math.nsc.ru, aliuysha-abs@mail.ru

СЕМАНТИЧЕСКИЙ ПОДХОД К МОДЕЛИРОВАНИЮ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств является составной частью нормативно-методического обеспечения системы оценки качества освоения студентом учебного материала. Онтологический подход к моделированию фонда оценочных средств позволяет формировать актуальные оценочные документы, которые учитывают не только всевозможные пожелания экзаменатора, но и степень усвоения различных компетенций и степень овладения различными профессиональными функциями.

В статье приведено описание онтологии, семантической и нечеткой моделей фонда оценочных средств. Описан алгоритм порождения шаблона оценочного документа, состоящий из трех этапов: сужение, ограничение и определение. На каждом этапе генерируется соответствующая нечеткая модель, в рамках которой проверяется непротиворечивость заданного шаблона и выполнимость данного шаблона на имеющейся базе оценочных средств. Для формирования комплекта оценочных документов используется алгоритм CLOPE, позволяющий кластеризовать категорийные данные.

Ключевые слова: оценочный документ, комплект оценочных документов, фонд оценочных документов, онтология, семантическая модель, нечеткая модель.

Введение

Каждое учебное заведение планирует, какими способами и средствами будут оцениваться результаты обучения и как будут достигнуты цели образовательной программы. Высшее учебное заведение обязано обеспечить разработку объективных процедур оценки уровня знаний и умений обучающихся.

Первым этапом формирования фонда оценочных документов (далее ФОС) факультета является разработка ФОС по отдельным дисциплинам [1]. ФОС по дисциплине представляет собой совокупность контролирующих материалов, предназначенных для измерения уровня достижения студентом установленных результатов обучения данной учебной дисциплины. К таким материалам традиционно относятся экзаменационные билеты, варианты контрольных работ, тесты и т. п.

Особенностью образовательных стандартов нового поколения является применение компетентностного подхода при формировании учебного процесса. Понятие «компетенция» сложное и интегрированное, оно характеризует способность обучающегося применить все приобретенные навыки, умения и знания для решения задач в профессиональной и социальной областях [2]. В связи с этим возникает необходимость формировать оценочные документы, позволяющие оценивать уровень освоения обучаемым различных компетенций. К таким материалам можно отнести рефераты, эссе, курсовые работы и т. п. Эти оценочные средства могут разрабатываться в рамках отдельных дисциплин и быть междисциплинарными.

Яхъяева Г. Э., *Абсайдульева А. Р.* Семантический подход к моделированию фонда оценочных средств // Вестн. НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 113–121.

При планировании учебного процесса, необходимо учесть актуальные потребности рынка студентов с целью подготовки выпускника, способного успешно работать в профессиональной сфере. Исходя из этого в некоторых учебных заведениях темы выпускных квалификационных работ и дипломных проектов согласовываются с работодателями [3], а некоторые университеты заключают соглашения с различными компаниями для поддержания обратной связи [4].

Таким образом, возникает необходимость в формировании оценочных документов, отвечающих требованиям различных профессиональных стандартов, а также требованиям потенциальных работодателей (см, например, [5]). Такой оценочный документ может формироваться из оценочных материалов отдельных модулей учебной дисциплины или нескольких дисциплин.

В связи с постоянным развитием науки и техники структура и содержание учебных дисциплин, а также требования работодателей постоянно меняются. Следовательно, существует потребность в создании программной системы, работающей с постоянно пополняемой базой оценочных материалов. Данная программная система должна обладать возможностью автоматизированного порождения актуальных документов оценки, которые будут отвечать различным запросам экзаменаторов. Семантический подход к моделированию ФОС позволяет успешно справляться с этой задачей.

Семантический подход используется в различных сферах и системах обучения, например для моделирования и управления учебной программой [6]. Моделирование облегчает доступ к учебной программе и позволяет ее составителям просматривать общий учебный план и обеспечивать соответствие основным целям учебного учреждения. Учебная программа представляет собой структуру, где учебные единицы связаны с результатами и задачами обучения. Кроме того, семантическое моделирование учебной программы облегчает периодическую оценку и анализ соответствия стандартам и потребностям рынка, его можно использовать для обзора, оценки и улучшения программы путем определения ее основных элементов, связывания учебных единиц с задачами и результатами, а также между собой (для определения последовательностей и предпосылок).

Онтологии предметных областей (история, география, программирование и др.) и онтологии различных элементов обучения (урок, способы оценивания, упражнения) [7] позволяют более полно описать сферы учебной деятельности и извлечь необходимую информацию. Описание данных учащихся полезно для оценивания и персонализации. Персонализация в соответствии с профилем учащегося может включать в себя упорядочивание учебного материала и отслеживание его эффективности (данные об оценках и пройденный материал) [8].

Семантическое моделирование также активно используется в оценивании качества освоения учебного материала учащимся. Система [9] представляет собой концептуальную карту, основанную на оценках, полученных студентами в процессе обучения. Эта карта используется в качестве инструмента для определения знаний студентов по пройденным темам. Онлайн-система оценивания OeLe [10] основана на онтологии, которая автоматически маркирует текстовые ответы учащихся на вопросы концептуального характера. Это делается путем сопоставления ответа ученика в форме карты понятий и онтологии предметной области. Помимо оценивания работы студентов, система OeLe также предоставляет информацию об эффективности освоения знаний учащимся, а также содержит отзывы преподавателей о студентах.

Онтология и семантическая модель

Формализация предметной области Δ = «Фонд оценочных средств факультета» производилась на языке логики описаний [11] с применением методов семантического моделирования [12] и теории нечетких моделей [13; 14]. Центральным понятием в данной предметной области является понятие «задание». Под *Заданием* мы понимаем минимальную составляющую единицу оценочного документа. Все задания делятся на *Теоретические Задания* и *Практические Задания*. Теоретические задания направлены на оценивание знаний студента. Практические задания направлены на оценивание умений и навыков студента.

По своей конструкции *Задание* делится на следующие виды: *Вопрос*, *Задача*, *Тестовое Задание*, *Списочное Задание* и *Шаблонное Задание*. Вопрос формализуется в виде предложения (или нескольких предложений) естественного языка, которые хранятся в системе в виде строковой величины и являются единым неделимым объектом. *Задача* хранится в системе в виде двух строковых величин: текст вопроса и текст решения / ответа. Текст решения / ответа не доступен студентам и может быть использован только экзаменатором. Тестовое задание может содержать от 3 до 10 строковых величин: вопрос, правильный ответ, неправильные ответы. *Списочное Задание* по своей структуре похоже на *Вопрос*, однако, в отличие от него, здесь оценка сложности (см. ниже) не является обязательной. Пример *Списочного Задания*: тема реферата, тема эссе и т. п. *Шаблонное Задание* подразумевает набор инструкций, следуя которым нужно выполнить задание (например, отчет о производственной практике).

Задания также можно разделить по уровню сложности. На сегодняшний день в системе реализовано три уровня сложности: *Легкое Задание*, *Задание Средней Сложности*, *Сложное Задание*. В дальнейшем возможна настройка системы на более мелкую градацию сложности заданий.

Задание должно быть привязано к учебному плану факультета: к отдельному модулю учебного плана, модулю дисциплин, отдельной дисциплине или к отдельной теме дисциплины. Также задание может быть привязано к профессиональному стандарту или к отдельным его профессиональным функциям. Эта привязка не является обязательной (например, если задание относится к общеобразовательной дисциплине). Задание должно быть привязано к некоторой компетенции или группе компетенций. Эту привязку можно производить либо непосредственно, либо автоматически через привязку к учебному плану.

Таким образом, строится множество CON_a атомарных понятий предметной области Δ, которое делится на следующие шесть классов:

- $\mathbb{P}_1 = \{$*Теоретическое*, *Практическое*$\}$;
- $\mathbb{P}_2 = \{$*Вопрос*, *Задача*, *Тестовое*, *Списочное*, *Шаблонное*$\}$;
- $\mathbb{P}_3 = \{$*Легкое*, *Среднее*, *Сложное*$\}$;
- $\mathbb{P}_4$ – множество понятий, отражающих взаимосвязь с учебным планом;
- $\mathbb{P}_5$ – множество понятий, отражающих взаимосвязь с множеством компетенций;
- $\mathbb{P}_6$ – множество понятий, отражающих взаимосвязь с трудовыми стандартами.

Каждое атомарное понятие мы воспринимаем как одноместный предикат, т. е. $P(x) \in CON_a$. Множество всех понятий CON данной предметной области строится согласно стандартному синтаксису, т. е. каждое понятие $\varphi \in CON$ является булевой комбинацией атомарных понятий.

В частности, нас будут интересовать формулы вида

$$Q(x,y) = \left(\neg P_1(x) \vee \neg P_1(y)\right) \& \ldots \& \left(\neg P_n(x) \vee \neg P_n(y)\right),$$

где $P, \ldots, P_n \in \left(\mathbb{P}_4 \cup \mathbb{P}_5 \cup \mathbb{P}_6\right)$. Семантически эти формулы означают, что два задания относятся, например, к разным темам в рамках одной дисциплины или к разным дисциплинам в рамках одного модуля, или направлены на проверку разных компетенций и т. д.

Для полного описания онтологии предметной области Δ задается конечное множество аксиом $Ax_\Delta \subseteq CON$. Мы рассматриваем аксиомы трех видов: аксиомы общего-частного, аксиомы исключения и аксиомы полноты. Поясним более подробно смысл этих аксиом.

Аксиомы общего-частного. Классы понятий $\mathbb{P}_4$, $\mathbb{P}_5$ и $\mathbb{P}_6$ иерархически упорядочены, например: если Задание относится к теме *«Логика высказываний»*, то оно относится к дисциплине *«Математическая логика»*, а значит, относится к модулю *«Базовых дисциплин»*.

Аксиомы исключения. Каждый из классов понятий $\mathbb{P}_1$, $\mathbb{P}_2$ и $\mathbb{P}_3$ является взаимоисключающим, например: если задание является легким, то оно не может быть сложным.

Аксиомы полноты. Мы предполагаем, что работаем в полностью определенной предметной области. Следовательно, мы считаем, что каждое рассматриваемое Задание должно об-

ладать хотя бы одним признаком из классов $\mathbb{P}_1$, $\mathbb{P}_2$, $\mathbb{P}_4$ и $\mathbb{P}_5$. Заметим, что задание может не обладать уровнем сложности (класс $\mathbb{P}_3$), а также может не быть привязано к трудовому стандарту (класс $\mathbb{P}_6$).

Заметим, что множество $CON_a \subseteq Ax_\Delta$ образует онтологию предметной области Δ (которую, согласно традициям Логики Описаний, будем называть $TBox$) и является первой компонентой Базы Знаний предметной области Δ. Второй компонентой Базы Знаний (будем обозначать через $ABox$) является описанием конкретных Заданий. Рассмотрим конечное множество Заданий $\mathbb{T} = \{t_1, t_2, ..., t_m\}$. Каждое задание t_i характеризуется наличием / отсутствием тех или иных понятий из CON_a. Тогда

$$ABox = \left\{ P(t) \middle| \text{ понятие } P(x) \in CON_a \text{ истинно на задании } t \in T \right\}.$$

Средствами Логики Описаний мы проверяем непротиворечивость $ABox$ и $TBox$. Далее пару $\langle TBox, ABox \rangle$ и будем называть Базой Знаний предметной области Δ. По мере появления в данной предметной области новых понятий (например, новых профессиональных стандартов) или новых заданий База Знаний будет расширяться. Однако общая структура Базы Знаний будет оставаться неизменной.

Для проверки непротиворечивости запросов экзаменатора и для определения мощности множества всех оценочных документов, соответствующих запросу экзаменатора, мы используем нечеткую модель [15] предметной области Δ. Эта модель строится на основе *семантической модели предметной области.*

Определение 1. Алгебраическую систему $\mathfrak{A}_\mathbb{T} = \langle \mathbb{T}, CON_a \rangle$ будем называть *семантической моделью* предметной области Δ, если $\mathfrak{A}_\mathbb{T} \models (Ax_\Delta \cup ABox)$.

Определение 2. Упорядоченную тройку $Fuz(\mathfrak{A}_\mathbb{T}) \rightleftharpoons \langle \{t\}, CON_a, \mu \rangle$ назовем *нечеткой моделью* предметной области Δ, порожденной моделью $\mathfrak{A}_\mathbb{T}$, если для любого понятия $\varphi(x) \in CON$ имеем

$$\mu_\mathbb{T}(\varphi(t)) = \frac{\left\| \left\{ t \in \mathbb{T} \middle| \mathfrak{A}_\mathbb{T} \models \varphi(t) \right\} \right\|}{\left\| \mathbb{T} \right\|}.$$

Значениями истинности предложений (понятий) в нечеткой модели являются числа из интервала [0,1], которые отражают статистику Базы Знаний. Более подробное описание свойств нечетких моделей можно найти в работах [16; 17].

Шаблоны оценочных документов

Оценочный документ – это набор заданий, который предоставляется студенту с целью проверки его знаний, умений или навыков. Традиционными оценочными документами являются экзаменационный билет, вариант контрольной работы, тест и т. п. В разрабатываемой системе реализована возможность формирования различных оценочных документов. Оценочный документ может быть сформирован для оценки внутри одной дисциплины или быть междисциплинарным, может быть направлен на проверку овладения той или иной компетентностью или трудовой функцией. Вид оценочного документа формирует *экзаменатор*, составляя *шаблон оценочного документа*.

Формирование шаблона оценочного документа происходит в три этапа: *сужение*, *ограничение* и *определение.*

Этап сужения направлен на формализацию цели проверки. На этом этапе экзаменатор определяет набор дисциплин или набор компетенций, или набор трудовых функций, на проверку которых направлен оценочный документ. В системе данный запрос формализуется в виде формулы:

$$F_1(x) \vee ... \vee P_k(x),$$

где $P_1, ..., P_k \in (\mathbb{P}_4 \cup \mathbb{P}_5 \cup \mathbb{P}_6)$.

Далее производится проверка формулы $F_1(x)$ на выполнимость на модели $Fuz(\mathfrak{A}_{\mathbb{T}})$. Описание алгоритма нахождения значения истинности бесквантворного приложения на нечеткой модели можно найти в работе [18].

Если оказывается, что $\mu_{\mathbb{T}}(F_1(x)) \le k_1 \ \ (k_1 \in [0,1))$, то делается вывод о неполноте базы знаний. В этом случае экзаменатору нужно либо пополнить базу знаний новыми заданиями, либо пересмотреть цель проверки.

На втором этапе экзаменатор задает количество заданий в оценочном документе и накладывает ограничения на совместимость заданий. По умолчанию ставится следующее ограничение совместимости заданий:

$$F_2(x_1,...,x_n) = \left(\bigwedge_{i \ne j} (x_i \ne x_j) \right),$$

где n – количество заданий в одном документе.

Это ограничение позволяет отслеживать, чтобы в одном оценочном документе не было повторяющихся заданий. По желанию экзаменатор может наложить более сильные ограничения. Например: *никакие два задания в оценочном документе не должны принадлежать одной дисциплине*, *и никакие два задания в оценочном документе не должны быть направлены на проверку одной и той же компетенции*. Общий вид формулы, задающей ограничения следующий:

$$F_2(x_1,...,x_n) = \left(\bigwedge_{i \ne j} (Q_1(x_i,x_j) \,\&\, ... \,\&\, Q_l(x_i,x_j)) \right).$$

Для проверки этого ограничения строится модель $\mathfrak{A}_{\mathbb{T}^n} = \langle \mathbb{T}^n, \sigma(F_1) \rangle$ и соответствующая ей нечеткая модель $Fuz(\mathfrak{A}_{\mathbb{T}})$. Так же, как на предыдущем этапе, требуется, чтобы значение истинности формулы $F_2(x_1,...,x_n)$ на модели $Fuz(\mathfrak{A}_{\mathbb{T}^n})$ превышало некоторый заданный порог, т. е. $\mu_{\mathbb{T}^n}(F_2(\langle x_1,...,x_n \rangle)) > k_2$, где $k_2 \in [0,1)$.

На третьем, последнем этапе формирования шаблона оценочного документа определяется структура этого документа. На этом этапе экзаменатор имеет возможность задать тип, сложность заданий в оценочном документе и т. п. Например: *Первое и второе задания являются Теоретическими*, *третье задание является Практическим. Хотя бы одно задание должно быть Сложным. Если задание Направлено на освоение общеобразовательной компетенции*, *то оно должно быть Средней сложности.*

Такое определение задается некоторой бесквантворной формулой $F_3(x_1,...,x_n)$ в сигнатуре $\sigma^* = \mathbb{P}_1 \cup \mathbb{P}_2 \cup \mathbb{P}_3 \cup \sigma(F_1)$. Для проверки корректности определения структуры оценочного документа строится расширение модели $\mathfrak{A}_{\mathbb{T}^n}$ на сигнатуру σ^*, т. е. модель $\mathfrak{A}^*_{\mathbb{T}^n} = \langle \mathbb{T}^n, \sigma^* \rangle$. Значение истинности формулы $F_2(x_1,...,x_n) \,\&\, F_3(x_1,...,x_n)$ на нечеткой модели $Fuz(\mathfrak{A}^*_{\mathbb{T}^n})$ показывает, сколько различных оценочных документов, соответствующих построенному шаблону, можно сгенерировать в рамках данной базы заданий.

Заметим, что если $\mu^*_{\mathbb{T}^n}(F_2(x_1,...,x_n) \,\&\, F_3(x_1,...,x_n)) = 0$, то необходимо проверить данную формулу на логическую выполнимость. Если формула $F_2(x_1,...,x_n) \,\&\, F_3(x_1,...,x_n)$ невыполнима, то система делает вывод, что запрос экзаменатора некорректен и просит пересмотреть запрос.

Комплекты оценочных документов

Часто нам приходится экзаменовать не одного студента, а группу студентов, поэтому есть необходимость формирования комплекта оценочных документов, соответствующих данному запросу. При этом необходимо, чтобы оценочные документы, входящие в один комплект, были уникальны, т. е. имели как можно меньше одинаковых заданий.

Пусть $\mathbb{D}$ – множество всех оценочных документов, соответствующих заданному шаблону. Очевидно, что $\mathbb{D} \subseteq \mathbb{T}^n$.

Для формирования комплекта оценочных документов был выбран алгоритм кластеризации CLOPE [19], работающий на множестве документов $\mathbb{D}$. Его основным преимуществом является относительно высокая скорость для категорийных данных. Суть алгоритма заключается в разбиении на кластеры, при котором максимизируется специальная функция Profit, высчитываемая для каждого кластера C. Эта функция зависит от количества уникальных заданий кластера и некоторого коэффициента r – чем больше этот коэффициент, тем ниже уровень сходства и тем больше кластеров будет сгенерировано.

В результате работы алгоритма необходимо приблизиться к числу, равному количеству документов в комплекте. Для этого выполняется итеративная процедура.

1. На первой фазе происходит инициализация, формирующая начальное разбиение. Затем осуществляется повторный обход по документам (от одного, до трех раз), и если изменений не произошло, то алгоритм прекращает свою работу.

Оценочные документы для удобства хранятся в таблице. Происходит процедура считывания оценочных документов из таблицы, которые кладутся в соответствующий или новый кластер с максимальным значением $\mathrm{Profit}(C, r)$. В результате в таблицу записывается пара <ОД, номер кластера>.

2. Вторая фаза алгоритма аналогична первой. Для каждого оценочного документа происходит поиск подходящего кластера путем максимизации опорной функции $\mathrm{Profit}(C, r)$. При этом если номер нового кластера не совпадает с записью в таблице, то старое значение перезаписывается на новое.

Эта фаза повторяется, пока есть какие-то изменения.

В конце все пустые кластеры удаляются. Если полученное число кластеров равно или совсем немного отличается от требуемого (не более чем на два), то считается, что требуемый результат был достигнут. Тем самым формируется итоговый комплект документов путем случайного выбора любого представителя из каждого класса (одного или двух). В случае если отличие более чем на два, необходимо увеличить или уменьшить коэффициент r, в зависимости от разницы и повторить процедуру снова.

Заключение

Данная работа посвящена семантическому моделированию фонда оценочных средств факультета, который является одним из компонентов образовательного процесса университета. В статье дается описание онтологии ФОС, основных (атомных) понятий предметной области, а также набора терминологических аксиом. Семантический подход позволяет создавать различные оценочные документы: в рамках одной дисциплины, междисциплинарные, документы для проверки соответствия требованиям работодателя и т. д.

Средства логики описаний и теории нечетких моделей используется для проверки согласованности запросов экзаменатора и базы знаний. Они также применяются в вычисления мощности набора всех оценочных документов, соответствующих запросу преподавателя. Реализованный алгоритм кластеризации CLOPE позволяет генерировать уникальные наборы, отвечающие одному запросу.

Список литературы

1. *Zindinova N. S.* Creation of the funds of assessment means in the framework of the discipline with consideration for introduction of the federal educational state standards of higher vocational education // Вестн. Омского университета. 2014. № 2 (72). С. 182–184.

2. Профессиональные стандарты в области информационных технологий. М.: АП КИТ, 2008. 616 с.

3. *Титаренко С. А.* Контрольно-оценочные средства как мера форсированности профессиональных и общих компетенций // Проблемы и перспективы развития образования (IV): Материалы Междунар. науч. конф. Пермь: Меркурий, 2013. С. 133.

4. *Perez-Jimenez A., Reyes-Zurit F.* (Eds.). Feedback between universities and companies // 7th International Technology, Education and Development Conference (INTED). Valencia, Spain, 2013. P. 2916–2923.

5. *Sofjina V. N., Gribanova D. Y., Melenevskaja O. Y.* Monitoring of Students' Professional Merits at the University // International Scientific Conference on Society, Integration, Education. Rezekne, Latvia, 2015. P. 215–223.

6. *Бахвалов С. В., Берестнева О. Г., Марухина О. В.* Применение онтологического моделирования в задачах организации учебного процесса ВУЗа // Онтология проектирования. 2015. Т. 5, № 4 (18). С. 387–398.

7. *Смирнов С. В.* Онтологический анализ предметных областей моделирования // Изв. Самар. НЦ РАН. 2001. Т. 3, № 1. С. 62–70.

8. *Пронина В. А., Шипилина Л. Б.* Использование отношений между атрибутами для построения онтологии предметной области // Проблемы управления. 2009. № 1. С. 27–32.

9. *Mouromtsev D., Kozlov F., Kovriguina L., Parkhimovich O.* ECOLE: Student Knowledge Assessment in the Education Process // WWW 2015 Companion – Proceedings of the 24th International Conference on World Wide Web. 2015. P. 695–700.

10. *Litherland K., Carmichael P., Martínez-García A.* Ontology-based e-assessment for accounting: Outcomes of a pilot study and future prospects // J. Account. Educ. 2013. Vol. 31, no. 2. P. 162–176.

11. *Baader F., McGuinness D., Nardi D., Patel-Schneider P.* The description logic handbook: Theory, implementation, and applications. Cambridge: Cambridge University Press, 2007.

12. *Пальчунов Д. Е.* Моделирование мышления и формализация рефлексии. II. Онтологии и формализация понятий // Философия науки. 2008. № 2 (37). С. 62–99.

13. *Пальчунов Д. Е., Яхъяева Г. Э.* Нечеткие алгебраические системы // Вестн. НГУ. Серия: Математика, механика, информатика. 2010. Т. 10, вып. 3. С. 75–92.

14. *Пальчунов Д. Е., Яхъяева Г. Э.* Нечеткие логики и теория нечетких моделей // Алгебра и логика. 2015. Т. 54, № 1. С. 109–118.

15. *Yakhyaeva G.* Fuzzy model truth values // APLIMAT. 2007. № 6. С. 423–431.

16. *Пальчунов Д. Е., Яхъяева Г. Э., Ясинская О. В.* Применение методологии онтологического моделирования для задач диагностирования заболеваний позвоночника // Вестн. НГУ. Серия: Информационные технологии. 2015. Т. 13, № 3. С. 42–51.

17. *Яхъяева Г. Э., Карманова А. А., Ершов А. А., Савин Н. П.* Вопросно-ответная система для управления информационными рисками на основе теоретико-модельной формализации предметных областей // Информационные технологии. 2017. Т. 23, № 2. С. 97–106.

18. *Яхъяева Г. Э., Ясинская О. В.* Методы согласования знаний по компьютерной безопасности, извлеченных из различных документов // Вестн. НГУ. Серия: Информационные технологии. 2013. Т. 11, вып. 3. С. 63–73.

19. *Yang Y., Guan H., You J.* CLOPE: A fast and Effective Clustering Algorithm for Transactional Data // Proc. of SIGKDD'02. Edmonton, Alberta, Canada, 2002.

Материал поступил в редколлегию 04.04.2018

G. E. Yakhyaeva, **A. R. Absayduleva**

Novosibirsk State University
1 Pirogov Str., Novosibirsk, 630090, Russian Federation

gulnara@math.nsc.ru, aliuysha-abs@mail.ru

SEMANTIC APPROACH TO MODELING OF THE FUND OF ASSESSMENT MEANS

The fund of assessment means (FAM) is an integral part of the normative and methodological support of the system for assessing the quality of the student's learning. The ontological approach to

FAM modeling makes it possible to form current evaluation documents that take into account all the wishes of the examiner.

The article describes the ontology, semantic and fuzzy models of the FAM. An algorithm for generating a template for an assessment document is consisting of three steps: narrowing, restriction, and definition. At each stage, a corresponding fuzzy model is generated, within which the consistency of the given template is checked and the feasibility of this template on the available basis of valuation tools is checked. The CLOPE algorithm is used for generating a set of evaluation documents, which allows clustering the category data.

Keywords: assessment document, set of assessment documents, fund of assessment means, ontology, semantic model, fuzzy model.

References

1. Zindinova N. S. Creation of the funds of assessment means in the framework of the discipline with consideration for introduction of the federal educational state standards of higher vocational education. *Vestnik Omskogo universiteta*, 2014, no. 2 (72), p. 182–184. (in Russ.)

2. Professional standards in the field of information technology. Moscow, AP KIT, 2008, 616 p. (in Russ.)

3. Titarenko S. A. Control and evaluation tools as a measure of the forcing of professional and general competences. Problemy i perspektivy razvitiya obrazovania (IV): materialy medgdunarodnoi konferentsii. Perm, Merkuriy, 2013, p. 133. (in Russ.)

4. Perez-Jimenez A., Reyes-Zurit F. (Eds.). Feedback between universities and companies. *7th International Technology, Education and Development Conference (INTED)*. Valencia, Spain, 2013, p. 2916–2923.

5. Sofjina V. N., Gribanova D. Y., Melenevskaja O. Y. Monitoring of Students' Professional Merits at the University. *International Scientific Conference on Society, Integration, Education*. Rezekne, Latvia, 2015, p. 215–223.

6. Bahvalov S. V., Berestneva O. G., Maruhina O. V. Application of ontological modeling in the problems of organization of the educational process of the university. *Ontologiya proiektirovaniya*, 2015, vol. 5, no. 4 (18), p. 387–398. (in Russ.)

7. Smirnov S. V. Ontological analysis of modeling object areas. *Izvestia Samarskogo nsuchnogo centra RAN*, 2001, vol. 3, no. 1, p. 62–70. (in Russ.)

8. Pronina V. A., Shipilina L. B. Using Attribute Relationships to Construct a Domain Ontology. *Problemi upravleniya*, 2009, № 1, p. 27–32. (in Russ.)

9. Mouromtsev D., Kozlov F., Kovriguina L., Parkhimovich O. ECOLE: Student Knowledge Assessment in the Education Process. *WWW 2015 Companion – Proceedings of the 24th International Conference on World Wide Web*, 2015, p. 695–700.

10. Litherland K., Carmichael P., Martínez-García A. Ontology-based e-assessment for accounting: Outcomes of a pilot study and future prospects. J. Account. Educ., 2013, vol. 31, no. 2, p. 162–176.

11. Baader F., McGuinness D., Nardi D., Patel-Schneider P. The description logic handbook: Theory, implementation, and applications. Cambridge, Cambridge University Press, 2007.

12. Palchunov D. E. Modeling of intellection and formalization of reflection. II. Ontologies and formalization of concepts. *Filosofiya nauki*, 2008, no. 2 (37), p. 62–99. (in Russ.)

13. Palchunov D. E., Yakhyaeva G. E. Fuzzy Algebraic Systems. *Vestnik NSU. Series: Mathematics, mechanics, informatics*, 2010, vol. 10, no. 3, p. 75–92. (in Russ.)

14. Palchunov D. E., Yakhyaeva G. E. Fuzzy logics and theory of the fuzzy models. *Algebra i logika*, 2015, vol. 54, № 1, p. 109–118. (in Russ.)

15. Yakhyaeva G. Fuzzy model truth values. *APLIMAT*, 2007, no. 6, p. 423–431.

16. Palchunov D. E., Yakhyaeva G. E., Yasinskaya O. V. Application of model-theoretic methods and ontological modeling to automate the diagnosis of diseases. *Vestnik NSU. Series: Information Technologies*, 2015, vol. 13, no. 3, p. 42–51. (in Russ.)

17. Yakhyaeva G. E., Karmanova A. A., Ershov A. A., Savin N. P. Question-answering system for managing of the information risks based on model-theoretic formalization of the object domains. Informatsionnie tekhnologii, 2017, vol. 23, no. 2, p. 97–106. (in Russ.)

18. Yakhyaeva G. E., Yasinskaya O. V. Metody soglasovaniya znaniy po kompjuternoi bezopasnosti, izvlechennykh iz razlichnykh dokumentov. *Vestnik NSU. Series: Information Technologies*, 2013, vol. 11, no. 3, p. 63–73. (in Russ.)

19. Yang Y., Guan H., You J. CLOPE: A fast and Effective Clustering Algorithm for Transactional Data // Proc. of SIGKDD'02. Edmonton, Alberta, Canada, 2002.

For citation:

Yakhyaeva G. E., Absayduleva A. R. Semantic Approach to Modeling of the Fund of Assessment Means. *Vestnik NSU. Series*: *Information Technologies*, 2018, vol. 16, no. 2, p. 113–121. (in Russ.)

DOI 10.25205/1818-7900-2018-16-2-113-121

СВЕДЕНИЯ ОБ АВТОРАХ

Абсайдульева Алия Рашидовна – магистрант факультета информационных технологий Новосибирского государственного университета

Апанович Зинаида Владимировна – кандидат физико-математических наук, старший научный сотрудник Института систем информатики им. А. П. Ершова СО РАН (Новосибирск)

Батура Татьяна Викторовна – кандидат физико-математических наук, доцент, старший научный сотрудник Института систем информатики им. А. П. Ершова СО РАН (Новосибирск)

Бельченко Илья Владимирович – аспирант Кубанского государственного технологического университета (Краснодар)

Брак Иван Викторович – кандидат биологических наук, старший научный сотрудник Лаборатории аффективной, когнитивной и трансляционной нейронауки Научно-исследовательского института физиологии и фундаментальной медицины (Новосибирск)

Букшев Иван Евгеньевич – магистрант факультета информационных технологий Новосибирского государственного университета, iOS-разработчик Центра финансовых технологий (Новосибирск)

Власов Александр Александрович – кандидат технических наук, научный сотрудник Института нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН (Новосибирск)

Дьяченко Роман Александрович – доктор технических наук, доцент, директор Института компьютерных систем и информационной безопасности Кубанского государственного технологического университета (Краснодар)

Исаченко Владимир Викторович – магистрант факультета информационных технологий Новосибирского государственного университета

Князева Анна Анатольевна – кандидат технических наук, младший научный сотрудник Института вычислительных технологий СО РАН (Новосибирск)

Козодоев Алексей Викторович – младший научный сотрудник Института оптики атмосферы им. В. Е. Зуева СО РАН (Томск)

Козодоева Елена Михайловна – младший научный сотрудник Института оптики атмосферы им. В. Е. Зуева СО РАН (Томск)

Колобов Олег Сергеевич – кандидат технических наук, ведущий программист Института сильноточной электроники СО РАН (Томск)

Малых Александр Евгеньевич – магистрант факультета информационных технологий Новосибирского государственного университета, инженер НЦИТ «УНИПРО» (Новосибирск)

Мурзинцев Степан Витальевич – аспирант кафедры теоретической кибернетики и прикладной математики Алтайского государственного университета (Барнаул)

Сазонова Юлия Игоревна – магистрант факультета информационных технологий Новосибирского государственного университета, инженер-программист Лаборатории технологий анализа и обработки биомедицинских данных Института вычислительных технологий СО РАН (Новосибирск)

Стрекалова Светлана Евгеньевна – магистрант факультета информационных технологий Новосибирского государственного университета

Трошков Сергей Николаевич – магистрант механико-математического факультета Новосибирского государственного университета

Турчановский Игорь Юрьевич – кандидат физико-математических наук, директор филиала Института вычислительных технологий СО РАН (Томск)

Федотов Анатолий Михайлович – член-корреспондент РАН, доктор физико-математических наук, профессор, главный научный сотрудник Института вычислительных технологий СО РАН (Новосибирск)

Цхай Александр Андреевич – доктор технических наук, профессор, главный научный сотрудник Института водных и экологических проблем СО РАН; профессор Алтайского государственного технического университета им. И. И. Ползунова (Барнаул)

Чубаров Максим Витальевич – магистрант факультета информационных технологий Новосибирского государственного университета

Яхъяева Гульнара Эркиновна – кандидат физико-математических наук, доцент, доцент кафедры общей информатики факультета информационных технологий Новосибирского государственного университета

Авторы представляют статьи на русском или английском языке объемом от 0,5 авторского листа (20 тыс. знаков) до 1 авторского листа (40 тыс. знаков), включая иллюстрации (1 иллюстрация форматом 190 × 270 мм = 1/6 авторского листа, или 6,7 тыс. знаков).

Публикации, превышающие указанный объем, допускаются к рассмотрению только после индивидуального согласования с редакцией журнала.

Требования к оформлению основного текста и иллюстративных материалов

Текст рукописи должен быть представлен в редколлегию в виде файла MS Word. Гарнитура Times New Roman, размер шрифта 14, межстрочный интервал 1,5, размеры полей – стандартные значения текстового процессора. Форматирование – выравнивание по ширине страницы, переносы слов отключены, красной строки нет. Не допускается ручное форматирование абзацев (пробелами, лишними переводами строк, разрывами страниц).

Подзаголовки набираются полужирным шрифтом. Для выделения частей текста и отдельных слов и словосочетаний допускается использование курсива или полужирнго шрифта. Подчеркивание, разрядка, изменение основного кегля и выделение цветом и т. п. не используются.

Все иллюстрации к рукописи статьи должны быть приложены в отдельных файлах. При этом в тексте может содержаться как включенное изображение с указанием имени файла, так и только указание. Все иллюстрации, содержащие схемы, графики, алгоритмы, скриншоты и другие изображения должны быть представлены в максимально высоком качестве, без каких-либо потерь и искажений (.jpg, .tif).

Все иллюстрации должны иметь подрисуночную подпись.

Нумерация последовательная и неразрывная от начала статьи. Не допускается использование других наименований, кроме «рис.» и усложнение нумерации (например, «рис. 3.2.»). Ссылка на иллюстрацию в тексте должна быть приведена в круглых скобках: (рис. 1).

Формулы должны быть набраны с использованием редактора **MathType**. Кегль основных символов – 11, греческие символы набираются прямым шрифтом, латинские – курсивом. Нумеруются только те формулы, на которые автор ссылается в тексте.

Ссылки на литературу указываются цифрами в квадратных скобках. Список литературы нумеруется в порядке цитирования и оформляется в соответствии с ГОСТом на библиографическое описание. Ссылки на неопубликованные работы, а также на интернет-ресурсы (кроме электронных изданий, поддающихся библиографическому описанию) оформляются в виде сноски.

Дополнительные материалы

К статье обязательно прилагаются следующие материалы.

- Информация об авторе / авторах на русском и английском языках:
  - ФИО полностью,
  - сведения об ученой степени и ученом звании,
  - должность и место работы, почтовый адрес,
  - электронный адрес,
  - контактный телефон.
- Индекс УДК (Универсальной десятичной классификации).
- Название статьи на русском языке и его авторский перевод на английский язык.
- Аннотация статьи на русском и английском языках.
- Ключевые слова (не более 10–15), на русском и английском (Keywords) языках.
- Список литературы на русском и английском (References) языках.

Доставка материалов

Материалы предоставляются в редакцию только по электронной почте:

inftech@vestnik.nsu.ru