УДК 004.912

О. Л. Жижимов [1], **Ю. В. Титова** [2], **А. М. Федотов** [1]

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

[2] *Новосибирский государственный университет*
ул. Пирогова, 2, Новосибирск, 630090, Россия

zhizhim@mail.ru, yuliya.titova14@gmail.com, fedotov@nsu.ru

РЕАЛИЗАЦИЯ МЕТОДОВ АБСТРАКТНОГО ДОСТУПА К ТЕЗАУРУСУ

Статья посвящена описанию поиска терминов в тезаурусе и алгоритма построения дерева терминов на основе абстрактного доступа к тезаурусу, организованного в соответствии с профилем Zthes протокола Z39.50 (ISO-23950).

Ключевые слова: информационная система, тезаурус, Z39.50, поисковые атрибуты, RPN-запрос, XSLT-преобразование, иерархия терминов.

Введение

В настоящее время в мире существует большое количество тезаурусов, оформленных в виде машиночитаемых баз данных. Каждый из них, как правило имеет собственную схему данных и поддерживается собственным программным обеспечением. Как следствие, затруднена интеграция и совместное использование таких баз данных в распределенных информационных системах [1]. В связи с этим возникает проблема организации единого доступа к тезаурусу, а также унификации представления полученных данных.

В мире существует большое количество информационных ресурсов, каждый из которых имеет свою форму представления и хранения информации. Это вызывает глобальную проблему единого доступа к информации. Каждая база данных имеет уникальную структуру хранения информации, где каждое поле имеет свое название и назначение. Также для составления поисковых запросов к разным информационным ресурсам используются различные правила и языки программирования. Помимо этого, отличается способ представления найденной информации. Данные особенности вызывают определенные затруднения во время поиска информации и ее выдачи. Поэтому каждый информационный ресурс имеет свой индивидуальный интерфейс, который подходит для хранения используемых форматов данных. Пользователь вынужден осваивать каждый раз новый интерфейс и использовать предоставляемый ресурсом формат информации.

Для организации унифицированного доступа к базе данных тезауруса разработаны подходы на основе протоколов Z39.50 [1] и SRU/SRW [2]. Назначение протокола Z39.50 заключается в предоставлении компьютеру-клиенту возможности поиска информации на компьютере-сервере с помощью единой процедуры запроса. Протокол позволяет организовывать уни-

[1] ANSI/NISO Z39.50-2003. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification. NISO Press, Bethesda, Maryland, U.S.A. ISBN 1-880124-55-6. 267 p.

[2] SRU (Search/Retrieval via URL – The Library of Congress). URL: http://www.loc.gov/standards/sru

Жижимов О. Л., *Титова Ю. В.*, *Федотов А. М.* Реализация методов абстрактного доступа к тезаурусу // Вестн. НГУ. Серия: Информационные технологии. 2017. Т. 15, № 1. С. 15–35.

ISSN 1818-7900. Вестник НГУ. Серия: Информационные технологии. 2017. Том 15, № 1

версальный доступ к базам данных с разной структурой, а также определять формат представления полученной информации.

Протокол Z39.50

Протокол Z39.50 обеспечивает сетевой доступ к базам данных и является международным стандартом (ISO/FDIS 23950). Он относится к протоколам прикладного уровня эталонной модели взаимодействия открытых систем (OSI RM) [3]. Протокол Z39.50 был создан Библиотекой Конгресса США в 1980-е гг. В 1989 г. было создано Агентство поддержки Z39.50 (Z39.50 Maitenance Agency [4]), занимающееся в настоящее время развитием направлений использования протокола, а также созданием его новых версий [2].

Изначально протокол Z39.50 разрабатывался для доступа к библиографическим ресурсам, но в текущий момент он получил более широкую сферу применения. Сегодня с помощью технологии Z39.50 можно получить доступ к научно-техническим, биологическим, музейным данным, справочной информации и т. п.

Ранее, до возникновения протокола Z39.50, в качестве стандарта доступа к распределенным базам данных использовался протокол HTTP. Но он не позволяет унифицировать доступ к разнородной информации. Эту проблему можно решить только с помощью вспомогательных средств – языков программирования, например.

Поэтому информационные ресурсы, которые не поддерживают протокол Z39.50, являются обособленными и разнородными, что усложняет работу с ними. Пользователь, работая лишь с одним специальным приложением, построенным по протоколу Z39.50, может выполнять поиск по разным удаленным базам данных.

Абстрактная база данных, представленная протоколом, отображает конкретную модель существующей базы данных. Разработчику предстоит корректно сформировать структуру реальной базы данных, чтобы на ее основе можно было создать абстрактную модель.

Назовем некоторые особенности протокола Z39.50.

1. Модель представления информации, заложенная в протоколе, никак не зависит от источников информации, использующих этот протокол. Другими словами, протокол предоставляет некую абстрактную модель представления информации на каждом этапе взаимодействия клиента и сервера.

2. Протокол Z39.50 полностью обеспечивает сессионное взаимодействие клиента с сервером. Эта особенность заложена в самом протоколе и реализуется во всех его приложениях, будь то серверная система или программа-клиент.

Основанная идея представления информации при работе с протоколом Z39.50 лежит в абстрагировании от конкретной структуры какой бы то ни было базы данных. Для этого в стандарте описана некая абстрактная модель БД. Эта модель включает в себя полный набор элементов, необходимых для доступа и обработки информации, хранимой в БД. Абстрактная модель описывает в виде отдельных элементов не только, например, возможные поисковые поля или форматы выдачи информации, но и все выполняемые сервером операции.

Каждый элемент этой абстрактной модели подробно описывается до однозначного толкования и стандартизуется с присвоением уникального идентификатора – OID. Работа с каждой конкретной СУБД должна быть организована только через эту абстрактную модель путем обмена пакетами данных (APDU), содержащими последовательности идентифицируемых по меткам (tag) объектов.

Атрибутивный поиск

Согласно протоколу Z39.50 поисковые запросы формулируются не к реальной базе данных, а к абстрактной. Другими словами, данные извлекаются не из базы данных напрямую, а из промежуточных наборов, созданных сервером в момент осуществления запроса. Проме-

[3] ГОСТ Р ИСО/МЭК 7498-1-99 Информационная технология (ИТ). Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая модель. М.: ИПК, Издательство стандартов, 1999. 62 с.

[4] https://www.loc.gov/z3950/agency/

жуточные наборы данных характеризуется поисковыми атрибутами, которые используются для составления поискового запроса.

Протокол Z39.50 поддерживает обязательный тип запросов в обратной польской записи, который называет RPN-запросом (Reverse Polish Notation – Обратная польская нотация). Этот запрос может иметь сложную структуру, которая содержит комбинацию атрибутов и поисковых терминов. Атрибутами характеризуется набор параметров, которые определяют правила поиска каждого термина. Запрос может быть изображен для наглядности в виде строки. Для того чтобы указать поисковый атрибут, необходимо записать комбинацию из двух чисел (первое – тип, второе – значение). Таким образом, можно каждое поле таблицы базы данных представить в виде абстрактной записи.

Поиск в базе данных тезауруса осуществляется с помощью фиксированного набора атрибутов (Bib-1, XD-1, util и Zthes-1), которые включены в группу атрибутов Use (тип 1). Также для построения запроса используются пять типов дополнительных атрибутов (Relation (тип 2), Position (тип 3), Structure (тип 4), Truncation (тип 5), Completeness (тип 6)), уточняющих запрос. Наиболее распространен набор атрибутов Bib-1, включающий в тип Use поисковые атрибуты, такие как Author, Title, DatePublication и т. п.

Примером развернутого в строку RPN-запроса (запрос в синтаксисе PQF – Prefix Query Format [5]) может быть запрос на поиск записей, в которых автор начинается на «Иван» и встречается в любой позиции поля:

```
@attrset Bib-1 @attr 1=1003 @attr 2=3 @attr 3=3 @attr 5=1 {Иван},
```

где

`@attrset Bib-1` – задание набора атрибутов;
`@attr 1=1003` – соответствует полю `Author`;
`@attr 2=3` – равно;
`@attr 3=3` – любая позиция в поле;
`@attr 5=1` – усечение справа;
`Иван` – поисковый термин.

Запрос RPN можно представить в виде дерева, в узлах которого находятся связывающие операторы (AND, OR, AND-NOT). Листьями этого дерева являются блоки «атрибуты + терм» (АРТ). На рис. 1 схематично изображен запрос RPN. Дополнительно указан набор атрибутов, который используется по умолчанию (Attribute Set).

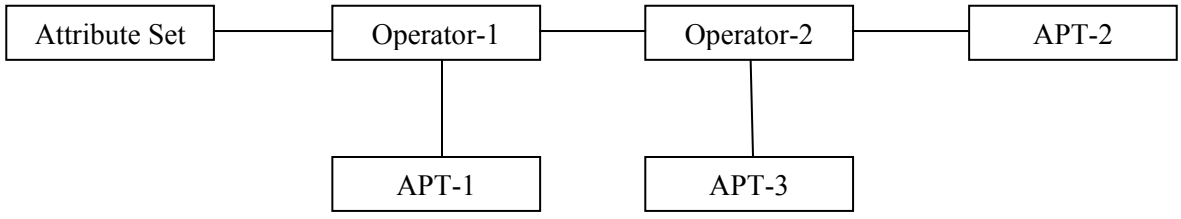


Рис. 1. Структура запроса RPN

Серверу передается древовидная RPN-структура, где каждая пара «атрибут=значение» представлена отдельным элементом. Такая организация системы запросов позволяет, с одной стороны, однозначно отобразить логику запроса, абстрагируясь от синтаксиса запроса конкретной СУБД, а с другой – абстрагироваться от поисковых полей конкретной базы данных, так как запрос формулируется всегда в терминах абстрактного набора атрибутов, например Bib-1 (набор атрибутов, ориентированный на работу с библиографическими базами данных).

[5] См.: http://www.indexdata.com/yaz/doc/tools.html#PQF

При таком подходе к процедуре поиска все базы данных становятся для пользователя одинаковыми, если поддерживают один и тот же набор поисковых атрибутов. Наборы поисковых атрибутов составляют класс объектов Z39.50 {1.2.840.10003.3}, подлежащих стандартизации. В классе {1.2.840.10003.3} для работы с тезаурусами используеся следующие наборы стандартных атрибутов (табл. 1).

Таблица 1

OID	Набор	Комментарий
1.2.840.10003.3.1	bib-1	Библиографическая информация
1.2.840.10003.3.11	util	Утилиты
1.2.840.10003.3.12	xd-1	Междоменный набор
1.2.840.10003.3.13	Zthes-1	Навигация по тезаурусам

Поисковые атрибуты

В соответствии с Z39.50 для работы с тезаурусами используются поисковые атрибуты с архитектурой Class 1.
Тип 1 – ***Access Point*** определяет смысловую информацию поискового термина. Является обязательным атрибутом в поисковом запросе. В табл. 2 приведены значения набора атрибутов **Zthes-1** для Типа 1.

Таблица 2

Набор атрибутов	Тип	Значение	Поиск по	Описание
Zthes-1	1	1	termQualifier	Поиск по значению termQualifier терминов верхнего уровня
Zthes-1	1	2	termType	Поиск по значению termType терминов верхнего уровня
Zthes-1	1	3	thesAdmin	Используется для различных поисковых запросов, связанных с административными деталями конструкции тезауруса. thesAdmin может принимать значения: “Start” – поиск всех записей, пригодными в качестве исходных точек для просмотра. “Whole” – поиск специальной записи, описывающей тезаурус в целом.
Zthes-1	1	4	relatedTermID	Используется в сочетании с семантическим квалификатором (атрибут типа 2) со значением равным: NT, BT, USE, UF, RT, LE Ведет поиск по всем записям в указанном отношении к записи, чей termID равен искомому термину. Например, поиск abc123 с точкой доступа relatedTermID и семантическим квалификатором “NT” находит более узкие термины в записи, чей termID равен abc123.

Для архитектуры Class 1 определены и другие типы атрибутов. В частности, для набора util (OID: 1.2.840.10003.3.11) отличные от Типа 1 атрибуты приведены в табл. 3.

Таблица 3

Тип	Значение	Название	Описание
Тип 2. ***Semantic Qualifier***			
2	1	Null	
2	2	Person	Персона
2	3	Institution	Организация
2	4	Process	Процесс
Тип 3. ***Language***			
3	Нет числовых значений атрибутов		
Тип 4. ***Content Authority***			
4	Нет числовых значений атрибутов		
Тип 5. ***Expansion/Interpretation***			
5	1	Left Truncation on Word boundary	Усечение слева на границе слова
5	2	Right Truncation on Word boundary	Усечение справа на границе слова
5	3	Left Truncation on Character boundary	Усечение слева на границе символа
5	4	Right Truncation on Character boundary	Усечение справа на границе символа
5	5	Regular Expression	Регулярное выражение
5	6	Masking	Задание маски. Символ '?' (вопросительный знак) используется для маскировки переменного количества символов. Символ '#' используется для маскировки одного символа.
5	7	Case Insensitive	Нечувствительность к регистру
5	8	Punctuation Insensitive	Нечувствительность к пунктуации
5	9	Whitespace Insensitive	Нечувствительность к пробелам. Сопоставление выполняется так, как если бы сервер нормализовал все экземпляры пробелов
5	10	Phonetic	Согласование основано на звуковом сходстве
5	11	Stem	Сопоставление основано на лексическом или лингвистическом сходстве; Совпадение успешно, когда термин и точка доступа имеют одну и ту же основу
5	12	Plural Matching	Множественное соответствие
5	13	No Stopwords	Нет стоп-слов
5	14	Search Words Stopped	Поиск слов остановлен. Имеет значение только в ответном запросе. Используется сервером, для индикации того, что одно или несколько слов в термине были им использованы в качестве стоп-слов.
5	15	No Other Expansion	Нет других расширений
5	16	RightTruncateEachWord	Усечение справа каждого слова
5	17	LeftTruncateEachWord	Усечение слева каждого слова
5	18	LeftAnchored	Закреплен слева
Тип 6. ***Normalized Weight***			
6	Нет числовых значений атрибутов		
Тип 7. ***Hit Count***			
2	Нет числовых значений атрибутов		

Окончание табл. 3

Тип	Значение	Название	Описание
Тип 8. ***Comparison***			
8	1	Always Matches	Всегда соответствует
8	2	Never Matches	Никогда не соответствует
8	3	Equal	Равно
8	4	Less Than	Меньше чем
8	5	Less Than Or Equal	Меньше чем или равно
8	6	Greater Than	Больше чем
8	7	Greater Than Or Equal	Больше чем или равно
8	8	Not Equal	Не равно
8	9	Contained Within	Содержится внутри
8	10	Relevance Feedback	Обратная связь релевантности
Тип 9. ***Format/structure***			
9	1	AdjacentWords	Смежные слова
9	2	AllTheseWords	Все слова
9	3	AnyOfTheseWords	Любые слова
Тип 10. ***Occurrence***			
10	Нет числовых значений атрибутов		
Тип 11. ***Indirection***			
11	1	URI	Термин является указателем (URI) к настоящему термину. Например, поддерживаемый термин должен быть URL, в этом случае, серверу следует пройти по ссылке и поставить её как термин. Это можно использовать, например, для обратной связи по релевантности, когда документ представляется термином, и поддерживается URL на него.
11	2	Scan Display Term	Сервер должен подменить действительный термин на отображаемый термин
Тип 12. ***Functional Qualifier***			
12	1	Null	
12	2	Creation	Создание
12	3	Modification	Модификация
12	4	Review	Просмотр
12	5	Deletion	Удаление

Для набора Zthes-1 (OID: 1.2.840.10003.3.13) отличные от Типа 1 атрибуты приведены в табл. 4.

Таблица 4

Тип	Значение
Тип 2. ***Semantic Qualifier***	
2	NT
2	BT
2	USE
2	UF
2	RT
2	LE

Приведенные в табл. 3–4 атрибуты с архитектурой Class 1 отличаются от архитектуры атрибутов Bib-1, которые чаще всего используются при поиске в Z39.50 и являются атрибутами «по умолчанию». Ниже приведены атрибуты Bib-1.

Тип 2 – ***Relation*** указывает на то, как поисковый терм соотносится с выбираемыми данными из полей, определенных атрибутом Тип 1 (***Use***) (табл. 5).

Таблица 5

Тип	Значение	Название	Описание
2	1	Less than	Меньше чем
2	2	Less than or equal	Меньше чем или равно
2	3	Equal	Равно
2	4	Greater or equal	Больше или равно
2	5	Greater than	Больше чем
2	6	Not equal	Не равно

Тип 3 – ***Position*** определяет место нахождения поискового терма (табл. 6).

Таблица 6

Тип	Значение	Название	Описание
3	1	First in field	Первый в поле
3	2	First in subfield	Первый в подполе
3	3	Any position in field	Любая позиция в поле

Тип 4 – ***Structure*** указывает, какую структуру имеет поисковый терм (табл. 7).

Таблица 7

Тип	Значение	Название	Описание
4	1	Phrase	Набор слов, разделенных пробелом
4	2	Word	Слово
4	3	Key	Последовательность символов, содержащихся в слове
4	4	Year	Год как четыре цифры
4	5	Date	День, месяц, год
4	6	Word list	Список слов, состоящий из одного или нескольких слов, разделенных пробелом

Тип 5 – ***Truncation*** указывает на то, что представляет собой поисковый терм (табл. 8).

Таблица 8

Тип	Значение	Название	Описание
5	1	Right truncation	Усечение справа
5	2	Left truncation	Усечение слева
5	3	Left and right	Справа и слева
5	100	Do not truncate	Без усечения

Тип 6 – ***Completeness*** указывает на обязательную область совпадения при поиске.

Таблица 9

Тип	Значение	Название	Описание
6	1	Incomplete subfield	Набор слов, разделенных пробелом
6	2	Complete subfield	Слово
6	3	Complete field	Последовательность символов, содержащихся в слове

Полный список атрибутов всех типов приведен в описании Z39.50 и дополнительных документах [6].

Для работы с тезаурусами применяются также наборы атрибутов XD-1 и util, что обеспечивает поиск по всем элементам записи схемы Zthes, которая используется для внешнего представления тезауруса [7] (табл. 10).

Таблица 10

Набор атрибутов	Тип	Значение	Поиск по	Описание
XD-1	1	1	termName	Имя терма
XD-1	1	4	termNote	Описание терма
Util	1	1	termCreatedDate	Используется вместе с функциональным квалификатором «creation» (12=2)
Util	1	1	termModifiedDate	Используется вместе с функциональным квалификатором «modification» (12=3)
Util	1	2	termCreatedBy	Используется вместе с функциональным квалификатором «creation» (12=2)
Util	1	2	termModifiedBy	Используется вместе с функциональным квалификатором «modification» (12=3)
Util	1	2	termCreatedBy или termModifiedBy	Используется без функционального квалификатора
Util	1	3	termLanguage	Язык записи
Util	1	4	termID	Строка идентификатор (целочисленное или строка символов) назначается сервером, который однозначно идентифицирует запись в БД

[6] http://www.loc.gov/z3950/agency/defns/oids.html

[7] ANSI/NISO. Z39.19:2005 Guidelines for the construction, format and management of monolingual controlled vocabularies. NISO Press: Bethesda, MD, 2005. ISBN:1-880124-65-3.

Построение RPN запросов

Для построения RPN (PQF)-запросов нужно использовать следующую модель:

```
query ::= top-set query-struct
top-set ::= [ '@attrset' attrsetname ]
attrsetname :: 'Bib-1' | 'XD-1' | 'util' | 'Zthes'
query-struct ::= simple | complex
complex ::= operator query-struct query-struct.
operator ::= '@and' | '@or' | '@not'
simple ::= attr-spec term.
attr-spec ::= '@attr' [ attrsetname ] typ '=' value [ attr-spec ]
typ :: '1' | . . . |'12'
value :: numeric or string
term ::= string.
```

Примером простого RPN-запроса может быть запрос на поиск термина *Абак*, встречающегося в заголовке (termName):

```
@attr XD-1 1=1 Абак
```

где

`@attr XD-1 1=1` соответствует полю «Имя термина»;

`Абак` – поисковый термин.

Пример более сложного запроса для поиска терминов *Абак* или *Калькулятор*, встречающихся в заголовках:

```
@or @attr XD-1 1=1 Абак @attr XD-1 1=1 Калькулятор
```

Пример запроса, который находит записи, содержащие *Абак* в поле termName и *ru* в поле termLanguage или *Калькулятор* в поле termName и *ru* в поле termLanguage:

```
@or @and @attr XD-1 1=1 Абак @attr util 1=3 ru
    @and @attr XD-1 1=1 Калькулятор @attr util 1=3 ru
```

Пример запроса с использованием дополнительного атрибута, который позволяет задать в качестве поискового термина последовательность символов, содержащихся в слове:

```
@attr XD-1 1=1 @attr util 5=4 @attr util 9=2 {поиск инф}
```

Результатом такого запроса могут быть термины *Поиск информации* и *Информационный поиск.*

Пример запроса, где в качестве поискового термина задается фраза. Фраза представляет собой набор слов, разделенных пробелом. Результат запроса зависит от последовательности слов в запросе. В данном случае ответ будет содержать фразу *Поиск информации.*

```
@attr XD-1 1=1 @attr util 5=4 {поиск инф}
```

Пример запроса, который находит все записи за 2017-ый год, поиск осуществляется по полю termCreatedDate:

```
@attr util 1=1 @attr util 12=2 {2017}
```

Протокол SRU/SRW

Протокол SRU/SRW представляет собой XML-ориентированный клиент-серверный протокол. Протокол был разработан Библиотекой Конгресса США в среде библиотечных систем и предназначался для интеграции библиотечных каталогов с репозитариями открытого доступа. Первая спецификация протокола опубликована в 2003 г.

Структура протокола создана в результате 20-летнего опыта разработки протокола Z39.50. Протокол SRU/SRW позволяет создавать универсальные клиентские приложения для доступа к различным информационным ресурсам. Разработка интерфейсов к SRU/SRW хранилищам данных значительно проще, чем для Z39.50. Протокол SRU/SRW является одновременно надежным и простым для понимания, сохраняя при этом все сильные стороны своего предшественника. Одним из применений протокола SRU/SRW является решение задачи по созданию шлюзов доступа к базам данных.

Конструкции протокола могут передаваться двумя способами: как SOAP (SRW – Search/Retrieve Web Service)-сообщения или как параметры URL (SRU – Search/Retrieval by URL). SRW (Search/Retrieve Web Service)-сообщения передаются от клиента к серверу с помощью XML через HTTP с использованием рекомендации W3C SOAP. Протокол SRW поддерживает рекомендации Web Services Interoperability.

SRU (Search/Retrieve URL Service) – стандартный протокол для поисковых запросов на основе XML. Протокол SRU использует стандартный синтаксис CQL (Contextual Query Language), основан на URL. При использовании протокола SRU (Search Retrieval by URL) мы можем построить запрос к различным базам данных, основываясь только на методе GET протокола HTTP.

Язык запросов, используемый в SRW/SRU – **CQL (Context Query Language)**, который представляет собой формальный язык запросов к информационно-поисковым системам, определенный в терминах абстрактных индексов. Язык запросов CQL основан на концепции семантического или контекстного поиска, а не поиска по синтаксису. Один и тот же поисковый запрос может выполняться в самых различных структурах баз данных на разных серверах, но важно то, что оба сервера понимают смысл запроса. Для обеспечения контекстной совместимости CQL использует контекстный набор (Context Sets). В табл. 11 перечислен список контекстных наборов (Context Sets), которые используются для доступа к тезаурусам [8].

Таблица 11

Контекстный набор	Краткое обозначение	Идентификатор
CQL context set Version 1.2	cql	info:srw/cql-context-set/1/cql-v1.2
Dublin Core Context Set Version 1.1	dc	info:srw/cql-context-set/1/dc-v1.1
ZThes thesaurus context set v1.0.1	zthes	http://zthes.z3950.org/cql/1.0.1/
Record Metadata Context Set Version 1.0	rec	http://srw.cheshire3.org/contextSets/rec/1.0/

[8] http://zthes.z3950.org/cql/index.html

Соответствие абстрактных индексов для различных контекстных наборов для Zthes приведено в табл. 12.

Таблица 12

Наименование	Эквивалент RPN	Описание
zthes.qual	Zthes-1 1=1	Поиск по termQualifier
zthes.type	Zthes-1 1=2	Поиск по termType
zthes.admin	Zthes-1 1=3	Используется для поисковых запросов, связанных с административными деталями структуры тезауруса
zthes.nt	Zthes-1 1=4 2=nt	Поиск дочерних терминов, т. е. терминов более узкого смысла
zthes.bt	Zthes-1 1=4 2=bt	Поиск родительских терминов, т. е. терминов более широкого смысла
zthes.use	Zthes-1 1=4 2=use	Поиск предпочтительного термина, т. е. термина, используемого вместо этого
zthes.uf	Zthes-1 1=4 2=uf	Поиск предпочтительного термина, чей termID равен поисковому запросу
zthes.rt	Zthes-1 1=4 2=rt	Поиск всех «связанных терминов», чьи termID равны поисковому запросу
zthes.le	Zthes-1 1=4 2=le	Поиск всех лингвистических эквивалентов записи, чьи идентификаторы termID равны поисковому запросу
zthes.vocab	Zthes-1 1=5	Поиск по termVocabulary
zthes.cat	Zthes-1 1=6	Поиск по termCategory
zthes.status	Zthes-1 1=7	Поиск по termStatus
zthes.approval	Zthes-1 1=8	Поиск по termApproval
zthes.vocab	Zthes-1 1=5	Поиск по termVocabulary
zthes.cat	Zthes-1 1=6	Поиск по termCategory
rec.lang	util 1=3	Поиск по termLanguage
rec.id	util 1=4	Поиск по termID
rec.creationDate	util 1=1 12=2	Поиск по termCreatedDate
rec.creationAgentName	util 1=2 12=2	Поиск по termCreatedBy
rec.lastModificationDate	util 1=1 12=3	Поиск по termModifiedDate
rec.lastModificationAgentName	util 1=2 12=3	Поиск по termModifiedBy
rec.modificationAgentName	util 1=2	Поиск по termCreatedBy или termModifiedBy
dc.title	XD-1 1=1	Поиск по termName
dc. description	XD-1 1=4	Поиск по termNote
cql.anywhere	util 1=11	Поиск по всем элементам

Запрос CQL может состоять либо из одного поискового предложения, либо из нескольких поисковых предложений, связанных логическими операторами. В качестве простого примера приведем запрос на поиск термина *Абак* в поле termName:

```
dc.title = Абак
```

Далее расположен пример более сложного запроса, состоящего из двух поисковых предложений. Данный запрос находит записи, которые содержат *Абак* в поле termName или *BT* в поле termType:

```
dc.title = Абак or zthes.type = BT
```

Имя индекса (index name) всегда включает базовое и может также включать префикс, который определяет контекстный набор (Context Sets). Базовое имя и префикс разделяются точкой:

```
zthes.qual = BFC88BB8
```

Для поиска фраз используется соотношение *all*, *any*. В найденных записях должны быть представлены все слова (all) или любое из указанных слов (any) в любом порядке. Следующий запрос найдет все записи, которые содержат слова *Информационный* или *поиск*:

```
dc.title any "Информационный поиск"
```

В этом случае будут найдены все записи, которые содержат слова *Информационный* и *поиск* :

```
dc.title all "Информационный поиск"
```

Приложение для трансформации абстрактных запросов PQF

Для реализации приложения, реализующего абстрактный доступ к тезаурусу, был выбран «Тезаурус по информатике ИВТ СО РАН» [3]. Данный тезаурус представлен в соответствии с профилем Zthes, который соответствует стандарту ISO 25964-2:2013. Тезаурус имеет внешнее представление терминов через WEB-интерфейс (рис. 2) и представление в XML-формате, которое выглядит следующим образом:

```
<?xml version="1.0" encoding="UTF-8"?>
<Zthes xmlns:dc="http://purl.org/dc/elements/1.1/">
 <thes>
  <dc:title>Тезаурус по информатике</dc:title>
  <dc:creator>Федотов А.М.</dc:creator>
  <dc:rights>ИВТ СО РАН</dc:rights>
  <dc:language>ru</dc:language>

<dc:identifier>http://db4.sbras.ru:210/th_compsci</dc:identifier>
  <thesNote>Тезаурус по информатике создан в рамках работ по
...</thesNote>
 </thes>
 <term>
    <termID>979</termID>
    <termQualifier>BFC88BB8</termQualifier>
    <termName>Абак</termName>
    <termLanguage>ru</termLanguage>
    <termNote>(доска) — счётная доска, применявшаяся для арифметических вычислений приблизительно с V века до н. э. в Древней Греции, Древнем Риме. Доска абака была разделена линиями на полосы, счёт осуществлялся с помощью размещённых на полосах камней или других подобных предметов. Камешек для греческого абака назывался псифос; от этого слова было произведено название для счёта — псифофория, «раскладывание камешков».</termNote>
    <termCreatedDate>20141118</termCreatedDate>
    <termModifiedDate>20161026</termModifiedDate>
    <termModifiedBy>71 (Самбетбаева Мадина )</termModifiedBy>
    <termApproval>approved</termApproval>
  <relation>
```

```
    <termID>8492</termID>
    <relationType>BT</relationType>
    <termQualifier>C2C5F549</termQualifier>
    <termName>Вычислитель</termName>
    <termLanguage>ru</termLanguage>
  </relation>
  <relation>
    <termID>943</termID>
    <relationType>BT</relationType>
    <termQualifier>0F079126</termQualifier>
    <termName>Вычислительное устройство</termName>
    <termLanguage>ru</termLanguage>
  </relation>
  <relation>
    <termID>2198</termID>
    <relationType>BT</relationType>
    <termQualifier>1EEE296C</termQualifier>
    <termName>Цифровое вычислительное устройство</termName>
    <termLanguage>ru</termLanguage>
  </relation>
  <relation>
    <termID>1503</termID>
    <relationType>NT</relationType>
    <termQualifier>765A4B88</termQualifier>
    <termName>Рабдологический абак</termName>
    <termLanguage>ru</termLanguage>
  </relation>
  <relation>
    <termID>936</termID>
    <relationType>RT</relationType>
    <termQualifier>54F38E0C</termQualifier>
    <termName>Калькулятор</termName>
    <termLanguage>ru</termLanguage>
  </relation>
  <relation>
    <termID>949</termID>
    <relationType>RT</relationType>
    <termQualifier>1734E407</termQualifier>
    <termName>Арифмометр</termName>
    <termLanguage>ru</termLanguage>
  </relation>
  <relation>
    <termID>946</termID>
    <relationType>RT</relationType>
    <termQualifier>931BE724</termQualifier>
    <termName>Суммирующая машина</termName>
    <termLanguage>ru</termLanguage>
</relation>
</term>
</Zthes>
```

Рис. 2. Представление терминов через Web-интерфейс

Внутреннее представление тезауруса реляционное, данные хранятся в постреляционной СУБД PosgreSQL в виде таблицы со следующей структурой (табл. 13).

Таблица 13

Имя элемента	Название элемента
title	Название термина
link_id	UID Основной link_id (termID)
term_qualifier	term_qualifier
field	Нормальная форма термина
term_vocabulary	term_vocabulary
description	Описание термина (SN Scope Note)
document_language	Язык термина
term_category	term_category – принадлежность к множеству (MT)
resource_type	Тип термина: main(TT), add(NT), info(ND), фиктивный(NL)
relation_bt	relation_bt – связь с родительским термином
relation_nt	relation_nt – связь с дочерним термином
relation_use	relation_use – термин, который используется вместо данного
relation_uf	relation_uf – аскрипторы
relation_synonim	relation_syn – полные синонимы
relation_rt	relation_rt (список ассоциативных связей)
relation_le	relation_le (термин на другом языке)
subject_udc	Коды УДК (UDC)
subject_grnti	Коды классификатора ГРНТИ
document_correct	Правильность заполнения
date	Дата записи
last_mod	Дата модификации
document_owner	Владелец записи
document_modifier	Кто исправил запись

Для выполнения доступа к базе данных тезауруса, пользователю следует построить абстрактный запрос. Для этого было разработано пользовательское WEB-приложение, которое генерирует абстрактные запросы к базе данных тезауруса. Чтобы получить абстрактный за-

прос, пользователю необходимо заполнить поля ввода формы следующими поисковыми параметрами: имя поискового термина, название набора атрибутов, поисковый атрибут типа Use (Access Point). По желанию пользователь может построить запросы для поиска более точной информации, используя дополнительные поисковые атрибуты и логические операторы (рис. 3).

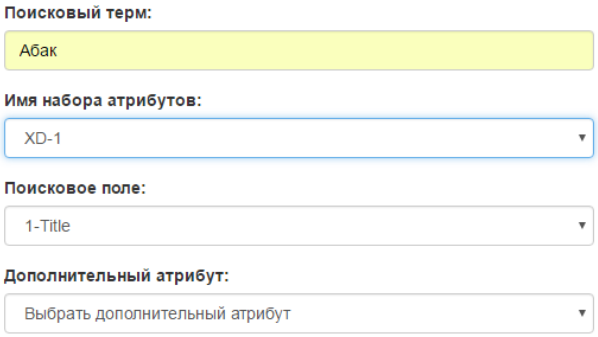


Рис. 3. Выбор параметров абстрактного запроса

После заполнения данных необходимо нажать на кнопку «Добавить». В расположенном ниже поле ввода будут отображены выбранные поисковые параметры (рис. 4).

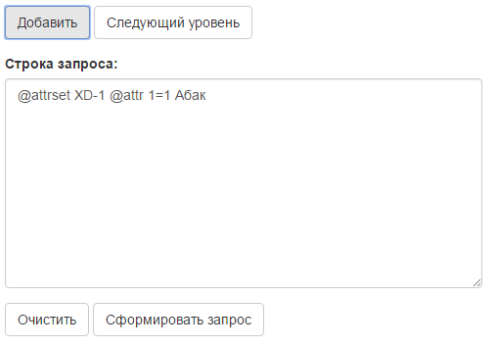


Рис. 4. Отображение поисковых параметров

Для создания более сложных поисковых конструкций пользователь должен использовать кнопку «Следующий уровень». Таким образом, выбранные заново поисковые параметры будут добавлены к предыдущему фрагменту абстрактного запроса (рис. 5).

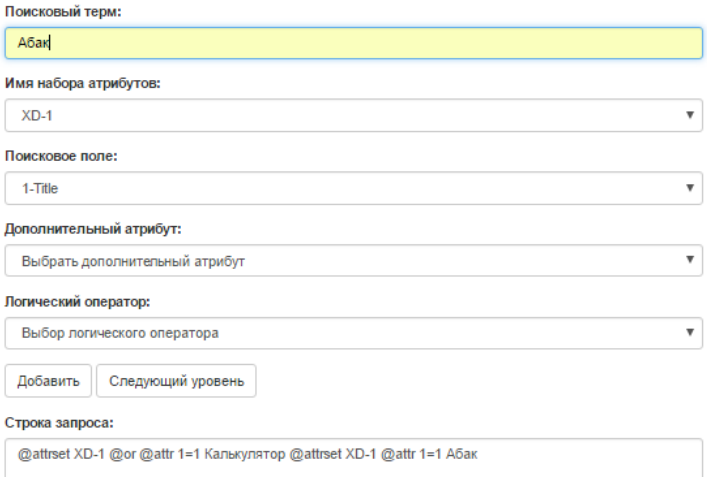


Рис. 5. Пример создания сложных запросов

Принцип работы приложения:

Для преобразования абстрактного запроса в реальный SQL-запрос к базе данных тезауруса была разработана функция, встраиваемая в СУБД, которая работает по следующему алгоритму.

Для начала пользователь должен построить абстрактный запрос, например с помощью приложения, и проверить его корректность с помощью приложения:

```
@or @attr XD-1 1=1 Абак @attr XD-1 1=1 @attr 5=1 Калькулятор
```

В функцию в качестве входного параметра передается данный запрос в форме строки. Далее происходит разбор запроса:

`@or` – логический оператор *ИЛИ*
`@attr XD-1 1=1` – поиск по «Имени термина»
`Абак` – поисковый термин
`@attr XD-1 1=1` – поиск по «Имени термина»
`@attr 5=1` – усечение справа
`Калькулятор` – поисковый термин

Полученный результат заменяется на фрагменты SQL-запроса:

```
@or -> or
    @attr XD-1 1=1 -> title
Абак -> 'Абак'
@attr XD-1 1=1  -> поиск по title
@attr 5=1 -> LIKE
Калькулятор -> 'Калькулятор%'
```

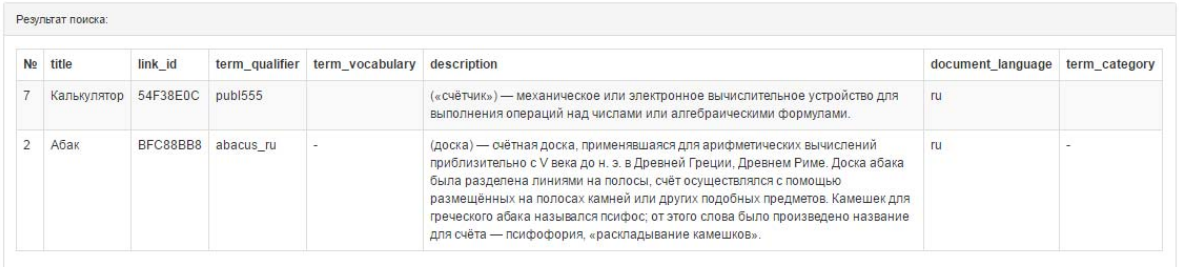
Далее происходит расстановка логических операторов и скобок:

```
((title = 'Абак') or (title LIKE 'Калькулятор%'))
```

После этого к базе данных тезауруса выполняется SQL-запрос:

```
SELECT * FROM zthes_cat WHERE ((title = 'Абак')
       or (title LIKE 'Калькулятор%'))
```

Полученный результат выводится в таблице в пользовательском приложении (рис. 6).



Результат поиска:

№	title	link_id	term_qualifier	term_vocabulary	description	document_language	term_category
7	Калькулятор	54F38E0C	publ555		(«счётчик») — механическое или электронное вычислительное устройство для выполнения операций над числами или алгебраическими формулами.	ru	
2	Абак	BFC88BB8	abacus_ru	-	(доска) — счётная доска, применявшаяся для арифметических вычислений приблизительно с V века до н. э. в Древней Греции, Древнем Риме. Доска абака была разделена линиями на полосы, счёт осуществлялся с помощью размещённых на полосах камней или других подобных предметов. Камешек для греческого абака назывался псифос; от этого слова было произведено название для счёта — псифофория, «раскладывание камешков».	ru	-

Рис. 6. Результат поиска

Построение дерева терминов

Одной из важных задач является извлечение из базы данных тезауруса дерева терминов (рис. 7). Для удобства пользователю необходима вся иерархия терминов. Для начала пользователь должен задать следующие поисковые параметры: головной термин (`termName`) и глубину вложения дочерних терминов (`depth`).

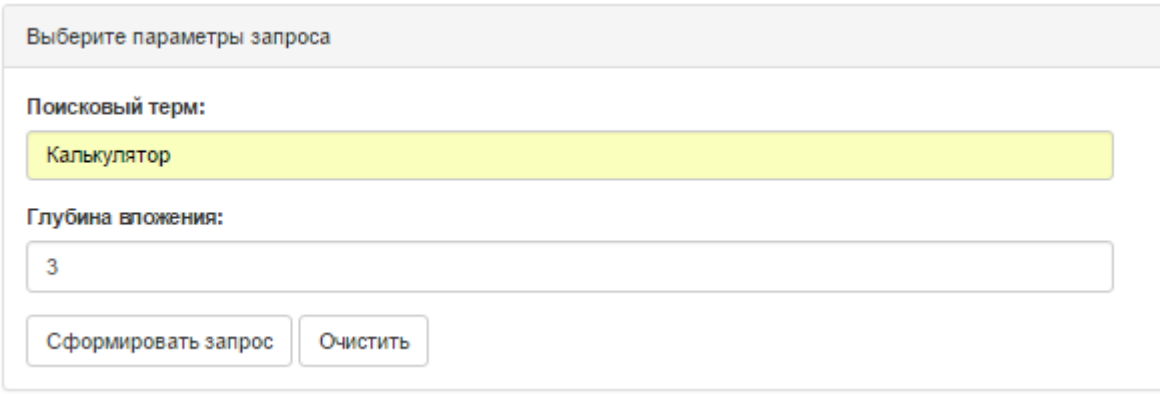


Рис. 7. Построение дерева терминов

К базе данных тезауруса строится рекурсивный SQL-запрос, который выдает все дочерние термины заданного головного термина. Алгоритм извлечения дерева терминов выглядит следующим образом:

- происходит выборка дочерних элементов первого уровня заданного головного термина;
- затем дочерние элементы первого уровня рассматриваются как головные элементы, снова происходит выборка всех дочерних терминов заданного головного термина.

Абстрактный запрос для выдачи иерархии терминов можно представить в следующей форме. Сначала из базы данных тезауруса извлекается головной термин «Калькулятор»:

```
@attr XD-1 1=1 Калькулятор
```

Из полученной записи выбираются все дочерние термины головного термина «Калькулятор». Ниже приведен пример поиска дочернего термина «ENIAC» с termID = 1503:

```
@attr Zthes-1 1=4 @attr Zthes-1 2=NT 1503
```

Описанные выше абстрактные запросы можно представить в виде одного рекурсивного SQL-запроса:

```
WITH RECURSIVE temp1 ( "title","relation_bt",PATH, LEVEL ) AS (
SELECT T1."title",T1."relation_bt", CAST (T1."title" AS TEXT) as
                                PATH, 1
 FROM zthes_cat T1 WHERE T1."relation_bt" LIKE '%Калькулятор%'
                                union
select T2."title", T2."relation_bt", CAST ( temp1.PATH ||'->'||
                T2."title" AS TEXT) ,LEVEL + 1
FROM zthes_cat T2 INNER JOIN temp1 ON(T2."relation_bt" LIKE '%'
                    || temp1."title" || '%')
                                )
          select * from temp1 ORDER BY PATH LIMIT 100
```

В результате работы программы после обработки запроса пользователь получает ответ, который отображается в виде XML. К полученному XML-документу подключается таблица стилей XSLT, которая содержится в файле с расширением .xsl (рис. 8).

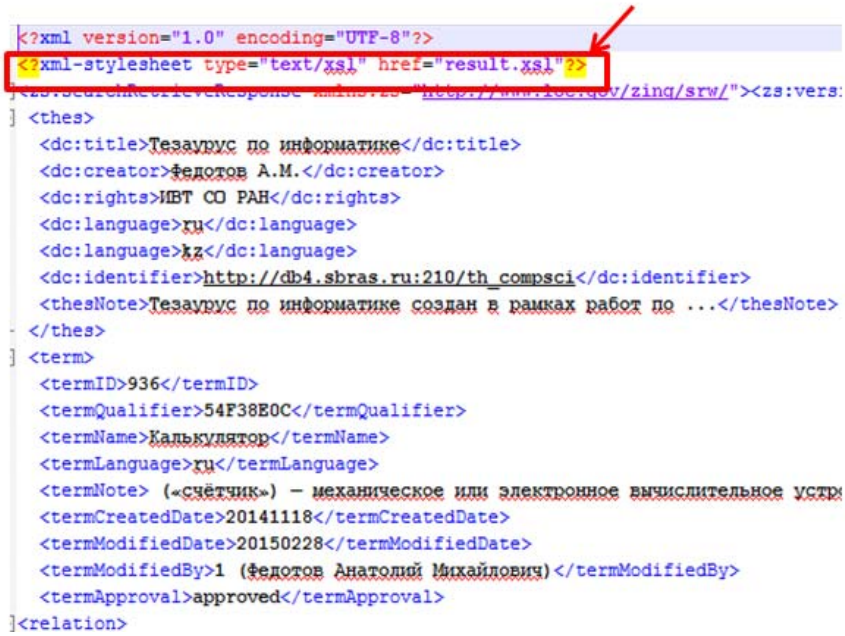


Рис. 8. Подключение таблицы стилей XSLT

XSLT (eXtensible Stylesheet Language Transformations) – язык преобразований, который предназначен для трансформации XML-документов. С помощью преобразования XSLT возможно не только выполнять трансформацию отображения XML-документа, но и создавать несложные программы. Работа XSLT-преобразования осуществляется на стороне клиента.

В результате работы поискового запроса пользователь получает дерево терминов, представленное в виде XML-файла, которое визуализируется при помощи XSLT-преобразования (рис. 9).

- Калькулятор
  - ENIAC
    - EDVAC
    - Принципы фон Неймана
  - Арифмометр
    - Арифметическая машина
    - Арифметический прибор (машина Лейбница)
    - Ая
    - Зубчатое колесо
    - Машина
    - Машина
    - Модель вычислителя
    - Ступенчатый вычислитель
    - Счет
    - Счеты (счеты русские)
    - Шина
  - Жиынтық машина
  - Модель вычислителя
    - Модель коллектива вычислителей
  - Паскалина
    - Паскалина
  - Суммирующая машина
    - Калькулятор
    - Машина Якобсона
    - Паскалина
    - Рабдологический абак

Рис. 9. Результат XSLT-преобразования

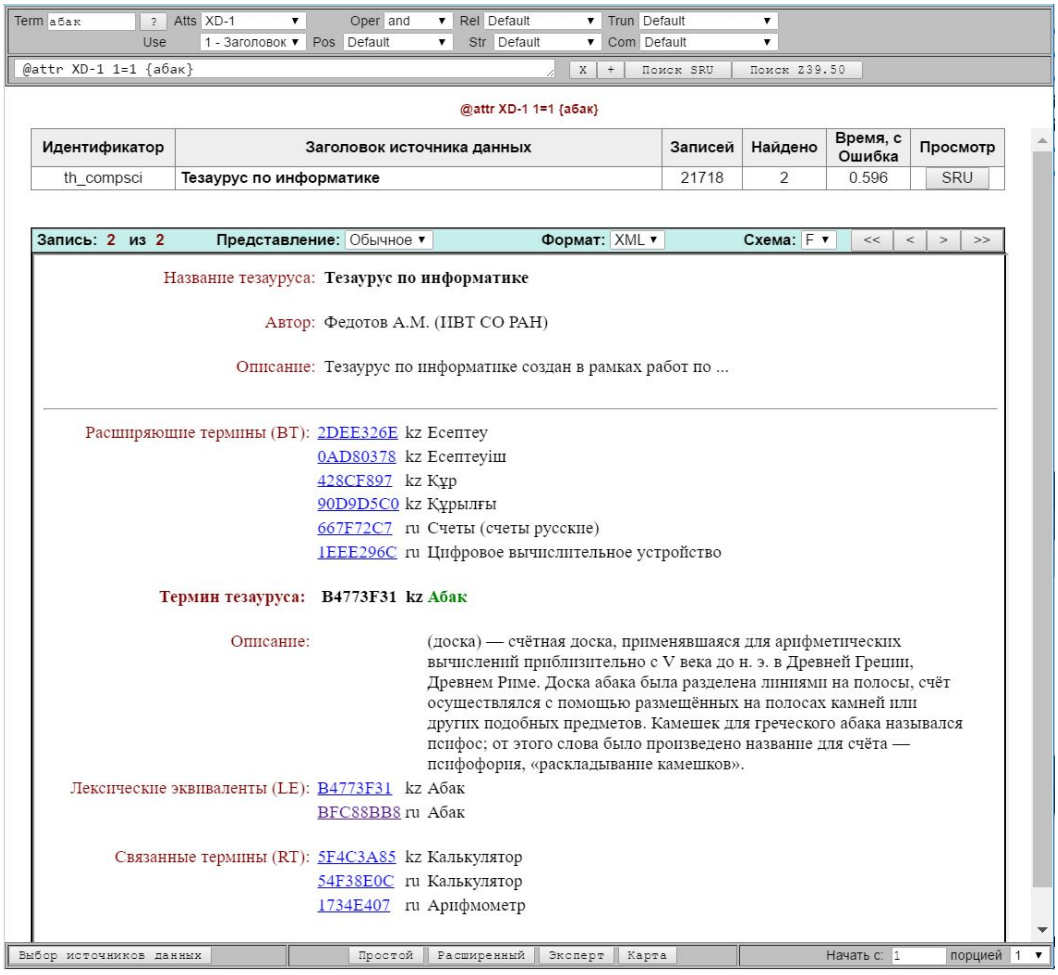


Рис. 10. Интерфейсы ZooSPACE для доступа к тезаурусам

ZooSPACE как платформа реализация методов абстрактного доступа

Описанные выше методы абстрактного доступа к тезаурусам для поиска, извлечения и визуализации информации реализованы также на платформе интеграции данных ZooSPACE [4] для сервера ZooPARK-ZS и подсистемы ZooSPACE-W. При этом поддерживается доступ к тезаурусам по протокалам Z39.50 и SRW/SRU, графические интерфейсы для конструирования запросов PQF, просмотр записей в различных форматах, в том числе XML с использованием преобразований XSLT [5]. Пример графических интерфейсов ZooSPACE для доступа к тезаурусам приведен на рис. 10.

Заключение

В статье представлен полный цикл работ с абстрактными RPN (PQF) запросами протокола Z39.50 (от генерации запроса до его выполнения), а также рассмотрен альтернативный подход на основе языка запросов CQL. Запросы могут быть простыми и сложными. Написано WEB-приложение, которое в интерактивном режиме создает RPN-запрос, проверяет его корректность и выполняет. Причем первые две задачи решаются на машине клиента. Реализован весь спектр запросов из архива Z39.50, включая поиск «фраз» и «наборов символов», а также построение дерева терминов, по некоторому искомому термину.

Все параметры приложения (в том числе, наборы атрибутов) являются внешними для приложения, поэтому оно может работать с любой базой данных тезауруса (нужно только заменить множество параметров).

Список литературы

1. *Мазов Н. А., Жижимов О. Л.* Применение протокола Z39.50 для работы с тезаурусами и классификационными схемами // Библиотеки и ассоциации в меняющемся мире: новые технологии и новые формы сотрудничества: Тр. VII Междунар. конф. М.: Изд-во ГПНТБ России, 2000. Т. 1. С. 402–405.

2. *Жижимов О. Л., Мазов Н. А.* Принципы построения распределенных информационных систем на основе протокола Z39.50 / ОИГГМ СО РАН. Новосибирск: ИВТ СО РАН, 2004. ISBN 5-9554-0017-6. 361 с.

3. *Федотов А. М., Идрисова И. А., Самбетбаева М. А., Федотова О. А.* Использование тезауруса в научно-образовательной информационной системе // Вестн. НГУ. Серия: Информационные технологии. 2015. Т. 13, № 2. С. 86–102. ISSN 1818-7900. EISSN 2410-0420.

4. *Жижимов О. Л., Федотов А. М., Шокин Ю. И.* Платформа ZooSPACE – организация доступа к разнородным распределенным ресурсам // Электронные библиотеки. 2014. Т. 17, № 2. ISSN 1562-5419. URL: http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2014/part2/ZFS

5. *Santeyeva S. A., Zhizhimov O. L.* User interfaces for working with thesauri and rubricators in distributed heterogeneous information systems on the example platform ZooSPACE // Совместный выпуск по материалам международной научной конференции «Вычислительные и информационные технологии в науке, технике и образовании» (CITech-2015) (24–27 сентября 2015 г.): Вычислительные технологии, т. 20; Вестник КазНУ им. Аль-Фараби. Серия математика, механика и информатика 2015. № 3 (86). Part I. Алматы; Новосибирск, 2005. P. 224–230.

Материал поступил в редколлегию 25.01.2017

O. L. Zhizhimov [1], **Yu. V. Titova** [2], **A. M. Fedotov** [1]

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

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

zhizhim@mail.ru, yuliya.titova14@gmail.com, fedotov@nsu.ru

IMPLEMENTATION OF THE METHODS OF ABSTRACT ACCESS TO THE THESAURUS

The article is about the algorithm of the terms tree construction using Z39.50 protocol. The object of research is the thesaurus of Information Technology which is based on Zthes standard data scheme. The alternative approach based on the CQL query language is also contemplated in the article.

There are many thesauruses in the world. And each thesaurus has a different structure. Consequently there is a problem of universal access to the thesaurus and data representation. The aim of research is to develop an application for abstract access to the thesaurus.

The research approach is based on analysis of standards, data scheme for thesaurus and search attributes representation. PHP programming language and Z39.50 protocol were used for application development. Protocol Z39.50 is a standard of network access to databases. The thesaurus on Information Technology is based on Zthes data scheme and supports Z39.50 protocol requests processing.

As a result of the research the application for abstract access to thesaurus was developed. The application allows users searching terms in the thesaurus. The search result is presented in an XML-file. The XML-file is visualized in browser using XSLT-transformation.

Keywords: thesauruses, Z39.50 protocol, data scheme, XSLT-transformation, the terms tree, RPN-query.

References

1. Zhizhimov O. L., Mazov N. A. Thesaurus and classification schemes access from the Internet via Z39.50 protocol. *Computing technologies*, 2000, T. 5. Special issue, related to questions of the development geoinformatical technologies and distant probes in SB RAS, p. 23–28. ISSN 1560-7534. EISSN 2313-691X. (in Russ.)

2. Zhizhimov O. L., Mazov N. A. Principles of building distributed information systems based on the Z39.50 protocol. UIGGM SB RAS. Novosibirsk, ICT SB RAS, 2004, 361 p. (in Russ.)

3. Fedotov A. M., Idrisova I. A., Sambetbayeva M. A., Fedotova O. A. Using the thesaurus in the scientific and educational information system. *Vestnik of Novosibirsk State University. Series: Information Technologies*, 2015, vol. 13, № 2, p. 86–102. ISSN 1818-7900. EISSN 2410-0420. (in Russ.)

4. Zhizhimov O. L, Fedotov A. M., Shokin Y. I. The ZooSPACE platform – hybrid distributed resource access organization. *Digital libraries*, 2014, vol. 17, № 2. ISSN 1562-5419. URL: http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2014/part2/ZFS/ (in Russ.)

5. Santeyeva S. A., Zhizhimov O. L. User interfaces for working with thesauri and rubricators in distributed heterogeneous information systems on the example platform ZooSPACE. *United issue on "Computational and information technologies in science" international scientific conference materials* (CITech-2015) (September 24–27, 2015): Computing and information technologies in science, technics and education, Vestnik of Al-Pharabi KazNU, "Mathematics, Mechanics and informatics" series, № 3 (68) / Al-Pharabi KazNU. 2015. Part I. Almaty, Novosibirsk, 2005, p. 224–230.

For citation:

Zhizhimov O. L., Titova Yu. V., Fedotov A. M. Implementation of the Methods of Abstract Access to the Thesaurus. *Vestnik NSU. Series*: *Information Technologies*, 2017, vol. 15, no. 1, p. 15–35. (in Russ.)