Подключение баз молекулярно-генетических данных к посреднику среды создания интегрированных электронных библиотек

Котляров Ю. В.
ВМиК МГУ
kot@ipi.ac.ru

Подколодный Н. Л.
ИЦиГ СО РАН
pnl@bionet.nsc.ru

Аннотация

Рассматриваются проблемы интеграции баз молекулярно-генетических данных. Предлагается способ описания структуры SRS баз данных[1]. Описывается преобразование языка запросов предметного посредника в язык запросов SRS. Определены возможности SRS-QL, которые нельзя представить на языке запросов посредника и предложен способ их реализации. Описывается архитектура адаптера, реализующего преобразование моделей данных и языков запросов.

1 ВВЕДЕНИЕ

В настоящее время в мире существует большое число разнородных слабоструктурированных молекулярно-генетических баз данных, содержащих результаты многочисленных, взаимодополняющих, пересекающихся и возможно противоречивых экспериментальных исследований [1, 2].

Одной из наиболее распространенных систем унифицированного доступа к базам молекулярно-генетических данных является система SRS (Sequence Retrieval System) [3]. На сегодняшний день в мире установлено более 50 SRS серверов, которые объединяют более 250 молекулярно-генетических баз данных, в частности, такие базы данных, как TRRD, MEDLINE, GeneBank, EMBL, Swiss-Prot, PIR, HSSP, PDB, NDB и др.

На Web интерфейсе SRS (SRSWWW) имеется возможность задавать запросы к базе данных на языке SRS Query Language (SRS-QL) [4].

С каждым годом объем данных в области молекулярной биологии экспоненциально возрастает, увеличивается число баз данных, разрабатываемых в различных регионах, усложняются структуры данных. Вследствие этого, крайне актуальной является проблема интеграции информационных ресурсов в этой области. В рамках проекта РФФИ[2] “Создание интегрированных библиотек на основе неоднородных распределенных электронных коллекций научной информации” предлагается универсальный подход интеграции информации путем создания *предметных посредников* над коллекциями данных [5, 6, 7], предоставляющих пользователям и приложениям виртуальное унифицированное представление информации в канонической модели данных языка СИНТЕЗ [8].

Язык СИНТЕЗ используется для унифицированного представления и манипулирования неоднородными информационными ресурсами. Важно отметить, что СИНТЕЗ предоставляет модели метаданных фактически любого вида и является расширяемым для отражения дальнейшего развития моделей данных и технологий. Он создает богатые возможности для описания гетерогенных информационных ресурсов: структурированных (информационные системы, базы данных), слабструктурированных (гипертекстовые документы) и неструктурированных (текстовые коллекции) данных, элементов баз знаний, онтологических спецификаций, деятельностей, потоков работ.

Взаимодействие посредника с конкретными коллекциями осуществляется посредством *адаптеров*, разрабатываемых специально для каждой модели данных.

В качестве языка запросов к адаптерам посредник использует подмножество языка запросов SOQL (SYNTHESIS Object Query Language). SOQL представляет собой вариант OQL ODMG [9] для использования в составе языка СИНТЕЗ. SOQL синтаксически близок к SQL-92. В нем поддерживается модель данных СИНТЕЗ и используется СИНТЕЗ-ориентированная нотация. Язык SOQL имеет следующие возможности: создания объектов, задания редуктов над типами, вызова операций, полиморфизма, типизации переменных. Кроме того, в языке можно использовать стандартные (для SQL-92 и OQL) операторы такие как — скалярные, логические, операторы сравнения. В подмножестве исключены возможности агрегирования данных: `count`, `sum`, `min`, `max`, `avg`, `GROUP`

[1] Базы данных доступ, к которым осуществляется через SRS интерфейс.

©Вторая Всероссийская научная конференция
ЭЛЕКТРОННЫЕ БИБЛИОТЕКИ:
ПЕРСПЕКТИВНЫЕ МЕТОДЫ И ТЕХНОЛОГИИ,
ЭЛЕКТРОННЫЕ КОЛЛЕКЦИИ
26-28 сентября 2000г., Протвино

[2] гранты 98-07-91061, 99-07-90203, 98-07-91078.

BY, HAVING; и упорядочения: ORDER BY.

Целью настоящей работы является рассмотрение вопросов подключения баз молекулярно-генетических данных к предметному посреднику, разработка адаптера, осуществляющего преобразование (1) унифицированного языка запросов посредника в язык запросов коллекции и (2) слабоструктурированных SRS баз данных в каноническую объектную модель данных посредника.

Базы данных SRS слабоструктурированы потому, что их структура явно не задана, т.е. они внутренне структурированы, но эта структура недоступна пользователям и приложениям. Данные могут быть неполными, избыточными, нерегулярными и даже неправильными. Поэтому проблема отображения баз данных в объектное представление является основной проблемой в данной работе.

Дальнейшее изложение статьи построено следующим образом. В первом разделе описывается опыт других проектов, направленных на преобразование молекулярно-генетических баз данных в объектную модель.

Во втором разделе рассматривается модель данных SRS, предлагается подход отображения SRS баз данных в каноническое представление.

Во третьем разделе кратко описывается преобразование языка запросов посредника в язык запросов SRS.

В четвертом разделе приводится описание архитектуры адаптера и рассматриваются ключевые моменты, касающиеся его реализации.

2 ПОДОБНЫЕ ПРОЕКТЫ

На сегодняшний день в мире ведется несколько проектов по представлению молекулярно-генетических данных в объектном виде. Одним из них является проект Европейского института биоинформатики (European Bioinformatics Institute) [10]. Для получения объектного представления базы данных описываются спецификации отображения на внутреннем языке SRS — Icarus [11]. В спецификации к именам и типам атрибутов объектов приписываются лексемы, получаемые при грамматическом разборе базы данных. Из описанных спецификаций генерируется код, выполняющий преобразование текстовой информации в объекты. Основное ограничение этого подхода в том, что для получения объектного представления коллекций необходимо создавать дополнительные модули на SRS сервере. Кроме того, существует проблема, связанная со сложностью представления полей в базах данных, когда некоторое поле может быть представлено несколькими атрибутами в объекте. В этом случае необходимо изменять правила разбора полей на подполя и только после этого описывать спецификации для отображения полей в объектный вид.

Проект Bioperl [12] также ставит перед собой задачу получения объектного представления записей слабоструктурированных молекулярно-генетических баз данных. Для каждого типа поля базы данных описывается модуль на языке Perl, который разбирает поля и подполя записей базы данных и формирует объекты из результатов разбора. Каждый модуль документируется и свободно распространяется для использования. Недостатком является то, что при изменении формата полей или добавлении нового поля приходится вносить изменения в код модуля. Кроме того, при столь огромном количестве баз данных написание и сопровождение модулей для каждого типа поля является просто невыполнимой задачей.

3 ПРЕОБРАЗОВАНИЕ SRS БАЗ ДАННЫХ

Базы данных SRS хранятся в виде текстовых ASCII файлов. Выдавая результаты на запрос, SRS модифицирует результаты поиска, добавляя к исходному содержанию файла гипертекстовые ссылки[3] на другие базы данных и прочую HTML разметку. Все данные, которые можно получить от SRS сервера, представляются только *строковым типом*.

Характеризуя структуру данных SRS, следует отметить, что она достаточно сложна. Каждый файл содержит последовательность *записей*. Записи в свою очередь состоят из *полей*. Причем количество и типы полей в разных записях могут быть неодинаковыми. Каждое поле представлено парами — **<имя_поля> <значение_поля>**.

Поля можно условно разбить на два вида:

Атомарные поля — поля, значения которых не делятся на подполя. Примером атомарных полей могут служить поля AN (SiteAccessionNumber), SQ (CoreSequence) в приведенной ниже записи[4].

Сложные поля — поля, значения которых состоят из значений двух и более подполей. Примером, сложных полей могут служить поля ID (GeneID), PQ (CoreSequencePosition), AG (ExperimentCodes).

```
...
AN S421
ID <A HREF=wgetz?[trrdgenes4-id:HBV:HBVE]>
   Gene: HBV:HBVE</A>
NM NF-1bs;
DR SAMPLES; <A HREF=wgetz?-e+[SAMPLES-id:NF-1]>
   NF-1</A>;
AT increase
SQ cggcaacggcctggtctgtgccaagtgtttg
PQ 1160 to 1190
PF 1165 to 1190
AG 1.1.1, 7.1 [<A HREF=wgetz?[TRRDBIB4-Authors:
   Ben]>Ben</a>-Levy R. et al., 1989]
//
...
```

Следует отметить, что структура некоторых сложных полей нерегулярна, т.е. может изменяться от записи к записи в базе данных. Ниже представлены возможные варианты поля AG (ExperimentCodes) в записях базы TRRDSITES.

```
...
AG 7.1, 7.2 [Delvin B.H. et al., 1989]
...
AG rat hepatoma H4IIEC3 cells: 6.1.1, 6.3.1, 6.5
```

[3] В терминах SRS — перекрестные ссылки.

[4] Здесь и далее примеры записей взяты из базы данных TRRDSITES [13].

```
   [Varanasi U. et al., 1996]
...
AG Afrikan green monkey kidney CV1 cells: 6.2,
   6.5 (ciprofibrate, 9-cisRA),
   6.6 (rRXRalpha, rPPARalpha)
   [Varanasi U. et al., 1996]
...
```

В первом из представленных полей отсутствует текстовый комментарий, отделяемый от последовательности кодов символом двоеточия (:), а в двух других — присутствует. В поле может встречаться один и более цифровых кодов, перечисленных через запятую, для которых могут быть определены некоторые комментарии в круглых скобках. Далее следует библиографическая ссылка в квадратных скобках.

Необходимо разработать способ, который позволит удобно и эффективно описывать структуру баз данных SRS. Реализация такого способа будет являться одной из основных частей адаптера, который осуществляет преобразование данных SRS в каноническую модель посредника.

Сделаем некоторые утверждения относительно соответствия сущностей SRS баз данных сущностям посредника.

Каждая запись SRS базы данных представляется экземпляром абстрактного типа данных посредника. Соответственно каждое поле представляется значением атрибута этого типа. Простые поля отображаются в атрибуты скалярного типа данных языка СИНТЕЗ. Сложные поля — в атрибуты абстрактного типа данных.

Выше мы определили, что все данные, получаемые от SRS, представляются только строковым типом. Это обстоятельство кажется очень удобным при преобразовании данных в каноническое представление. В самом деле, достаточно знать, к какому типу необходимо привести значение строкового типа данных SRS. Гораздо более сложная задача возникает при определении, того как будут соответствовать значения полей атрибутам типов канонического представления данных. Рассмотрим два случая: для простых полей SRS баз данных и для сложных.

В случае простых полей, их значения полностью соответствуют значениям атрибутов типов посредника.

Например[5]:

```
AN S421                 ANField = "S421"
NI 5                    NIField = 5
AT increase             ATField = "icrease"
```

Для преобразования сложных полей в значения атрибутов типов посредника необходим предварительный разбор на подполя.

Например, поле `PQ` будет представлено в типе посредника как атрибут абстрактного типа данных с двумя атрибутами целочисленного типа.

```
PQ -238 to -223         {PQField;
                          PQFrom = -238;
                          PQTo = -223;
                        }
```

[5]Здесь используется неформальный синтаксис языка СИНТЕЗ.

Гораздо более сложный разбор необходим для поля `AG` (см. выше). Для описания структуры полей SRS баз данных нами предлагается использовать контекстно-свободные грамматики [14]. Таким образом, грамматики соответствуют типам посредника, а деревья разбора над грамматикой — экземплярам типов.

Для синтаксического описания используется расширенная форма Бэкуса-Наура (EBNF) в следующей нотации:

`<symbol>` — нетерминальный символ `symbol`
`"terminal"` — терминальный символ `terminal`
`<x> <y>` — последовательность `xy`
`(<x> | <y>)` — представляет `x` или `y`
`[<x>]` — представляет `x` или пустую строку
`{<x>}` — представляет последовательность `x` или пустую строку

Подробнее рассмотрим грамматику, описывающую поле `AG`.

```
"AG" [<txt_cmt> ":"] <Codes> "[" <Ref> "]"

    <Codes>::= <code> {"," <code>}
    <Ref>::= <Lnk> <rest> "," <year>
    <Lnk>::=
      "<A HREF=" <url> ">" <anchor> "</A>"

      <txt_cmt>::= STRING
      <code>::= STRING
      <rest>::= STRING
      <year>::= INTEGER
      <url>::= STRING
      <anchor>::= STRING
```

Таким образом, использование контекстно-свободных грамматик для описания структуры баз данных позволит:

- унифицированно и наглядно описывать структуру SRS баз данных;
- генерировать код, осуществляющий разбор баз данных по их грамматическому описанию.

Дерево разбора результатов запросов SRS является структурированным представлением данных SRS и легко может быть преобразовано в экземпляры типов посредника.

4 ПРЕОБРАЗОВАНИЕ ЯЗЫКА ЗАПРОСОВ

Занимаясь отображением канонического языка запросов в язык запросов коллекции, необходимо ответить на два вопроса:

1. Все ли средства канонического языка представимы на языке запросов коллекции? Если нет, то необходимо выделить те возможности, которые представить в языке запросов коллекции нельзя и, по возможности возложить ответственность за их выполнение на адаптер.

2. Все ли возможности языка запросов коллекции присутствуют в каноническом языке запросов? Если нет, необходимо разработать способы реализации недостающих возможностей.

Здесь будут даны ответы на поставленные вопросы.

Отвечая на первый из них, можно сказать, что на SRS-QL невозможно задание вложенных запросов.

Что касается второго вопроса, то в подмножестве SOQL отсутствует возможность реализовать *операции связи*[6] языка SRS-QL стандартным средствами. Эти операции предлагается реализовать при помощи методов типов посредника. Адаптер должен уметь выполнять трансформацию вызова методов в синтаксические конструкции SRS-QL.

Вся работа с типами, классами, редуктами, типизированными переменными возлагается на адаптер.

5 АРХИТЕКТУРА АДАПТЕРА

Адаптер является компонентом архитектуры посредника и предоставляет интерфейс для доступа к информационному источнику.

Адаптер (1) принимает запрос на каноническом языке (подмножество SOQL) преобразует его в синтаксис языка запросов коллекции (SRS-QL), отсылает запрос SRS серверу; (2) получая результаты от SRS сервера, преобразует их в каноническое представление и передает посреднику.

В архитектуре адаптера можно выделить следующие компоненты (рис. 1):

- JDBC интерфейс. Интерфейсная часть адаптера.
- Синтаксический анализатор запросов посредника.
- Преобразователь запросов в термины SRS баз данных (использует трансформационную таблицу).
- Трансформационная таблица, задающая соответствие сущностей посредника сущностям баз данных SRS.
- Клиент SRS, ответственный за отсылку преобразованного запроса SRS серверам и получение результатов от них. Работает по протоколу HTTP.
- Синтаксический анализатор SRS данных.
- Пребразователь дерева разбора SRS данных в экземпляры типов посредника.

Посредством JDBC интерфейса строка запроса передается синтаксическому анализатору запросов. Синтаксический анализатор осуществляет разбор запроса, возвращает дерево разбора и передает преобразователю запроса, который, используя трансформационную таблицу, преобразует запрос к терминам и синтаксису SRS-QL. Далее преобразованный запрос передается SRS клиенту,

[6] операции для работы с перекрестными ссылками [4].

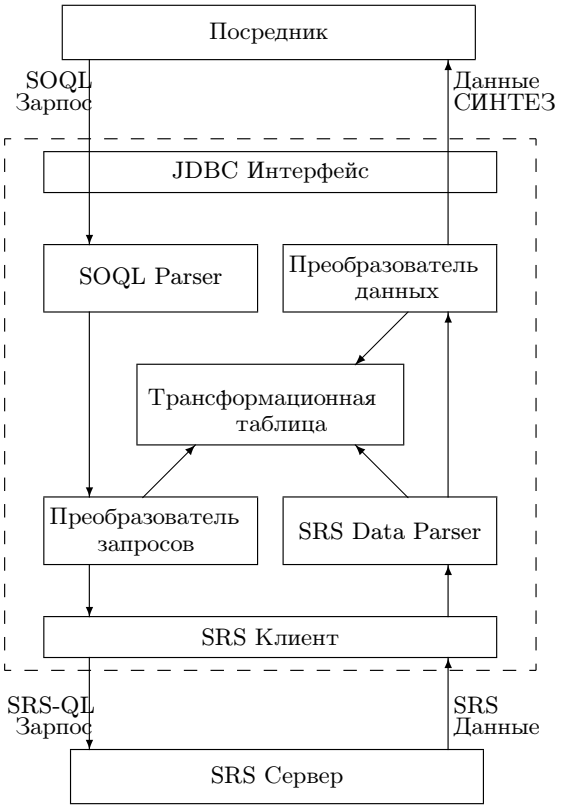

Рис. 1: Архитектура адаптера

который устанавливает соединение с сервером по протоколу HTTP, отправляет строку запроса, получает результаты в виде HTML файла и возвращает их синтаксическому анализатору SRS данных. Синтаксический анализатор строит дерево разбора данных SRS и передает его преобразователю данных. Преобразователь, из полученного дерева, конструирует экземпляры типов посредника. После этого, посредник имеет возможность получать данные через JDBC интерфейс.

Все компоненты адаптера реализованы на языке программирования Java с использованием пакета JDK 1.2.2. Синтаксические и лексические анализаторы SRS данных и языка запросов SOQL сгенерированы с использованием ANTLR [15].

6 ЗАКЛЮЧЕНИЕ

Итогом представленной работы являются следующие результаты.

Предложен способ описания структуры баз молекулярно-генетических данных SRS.

Описано преобразование языка запросов посредника в язык запросов SRS. Выявлены возможности SRS-QL, которые нельзя представить на языке запросов посредника и предложен способ их реализации.

Разработана архитектура адаптера, реализующего преобразование моделей данных и языков запросов. Реализованы все компоненты адаптера на языке программирования Java с использованием пакета JDK 1.2.2.

Лексический и синтаксические анализаторы SRS данных автоматически генерируются по грамматическим описаниям SRS баз данных. В качестве генератора анализаторов используется ANTLR (ранее PCCTS).

Однако, в процессе разработки возникла идея создания дополнительного инструментария адаптера, который позволит существенно упростить процесс подключения новой базы данных SRS к посреднику. При регистрации новой базы данных необходимо предоставить (для посредника) спецификацию типов и классов на языке СИНТЕЗ; (для адаптера) трансформационную таблицу, описывающую соответствие имен и типов посредника именам и полям коллекции; (для генератора синтаксического анализатора SRS данных) грамматическое описание базы данных. Кроме того, требуется реализовать код, который, получая от синтаксического анализатора дерево разбора данных, будет конструировать экземпляры типов посредника. Для этого требуется создание инструментария, который, получая на вход грамматическое описание базы данных, дополненное именами и типами, будет генерировать:

- спецификацию типов и классов предметного посредника на языке СИНТЕЗ;
- трансформационную таблицу;
- синтаксический анализатор данных SRS;
- код, создающий экземпляры типов посредника.

Список литературы

[1] Brucks C., *Molecular Biology Database List*, NAR, 1999. 27(1). P. 1-9.

[2] Rayl K. D., Gaasterland T., *Overview of selected molecular biological databases*, ANL/MCS-TM-200, November, 1994.

[3] Etzold T., Ulyanov A., Argos P., *SRS: Information Retrieval System for Molecular Biology Data Banks*, Methods in Enzymolog, 226, 1996.

[4] SRS Users Manual. `http://srs.ebi.ac.uk/`.

[5] Kalinichenko L.A., Briukhov D.O., Skvortsov N.A., Zakharov V.N., *Infrastructure of the subject mediating environment aiming at semantic interoperability of heterogeneous digital library collections.* Institute for Problems of Informatics RAS. Proceedings of the Second All Russian Conference on Digital Libraries, Protvino, September, 2000.

[6] Kalinichenko L. A., *Compositional Specification Calculus for Information Systems Development*, In Proceedings of the East-West Symposium on Advances in Databases and Information Systems (ADBIS'99), Maribor, Slovenia, September 1999, Springer Verlag, LNCS, 1999.

[7] Briukhov D., Kalinichenko L., *Component-based information systems development tool supporting the SYNTHESIS design method.* Proceedings of the East European Symposium on "Advances in Databases and Information Systems", September 1998, Poland, Springer, LNCS N 1475, 1998

[8] Калиниченко Л. А., *СИНТЕЗ: язык определения, проектирования и программирования интероперабельных сред неоднородных информационных ресурсов*, ИПИ РАН, Москва, 1995.

[9] Cattell R.G.G. et al., *The Object Data Standard: ODMG 3.0*, Morgan Kaufmann Publishers, 2000.

[10] Coupaye T., *Wrapping SRS with CORBA: from Textual Data to Distributed Objects*, Bioinformatics Journal, Chris Sander, Gary Stormo, Eds., Oxford University Press, 15(4), 1999.

[11] SRS Developers Manual. `http://srs.ebi.ac.uk/`.

[12] The Bioperl Project. `http://bio.perl.org/` .

[13] Transcription Regulatory Regions Database. `http://www.bionet.nsc.ru/trrd/`.

[14] Ginsburg S., *The Mathematical Theory of Context-Free Languages*, McGraw-Hill, 1966.

[15] ANTLR (ANother Tool for Language Recognition). `http://www.antlr.org/`