УДК 004.031.4

# Технология реализации ИРННД

И. В. Каменева, А. В. Ковалев, Т. Б. Шатовская
*Харьковский национальный университет радиоэлектроники, Украина*

В статье представлена исходная концептуальная схема интеллектуального репозитория научных наборов данных, которая базируется на агентных технологиях и онтологиях. Система предназначена для использования исследователями в области интеллектуального анализа данных и машинного обучения. Акцент делаем на хранение данных с использование онтологических моделей.

***Ключевые слова:*** *Репозитрий, агентные технологии, онтологии, Semantic web, RDF, Oracle Database 10g*

У статті представлена вихідна концептуальна схема інтелектуального репозиторія наукових наборів даних, яка базується на агентних технологіях і онтологіях. Система призначена для використання дослідниками у галузях інтелектуального аналізу даних (ІАД) та машинного навчання. Акцент робимо на зберігання даних з використанням онтологічних моделей.

***Ключові слова:*** *Репозиторій, агентні технології, онтології, Semantic web, RDF, Oracle Database 10g*

In the paper we present the initial conceptual scheme of intellectual repository of scientific data sets. It is based on agent technology and ontologies. System is intended for usage by researchers in the fields of data mining and machine learning. The emphasis is on data storage using ontological models.

***Key words:*** *Repository, agent technology, ontology, semantic web, RDF, oracle Database 10g*

## 1. Введение

Все чаще для проведения собственных исследований специалисты в области интеллектуального анализа данных и машинного обучения создают различные наборы исследовательских данных для хранения в различных статистических репозиториях. Примерами таких систем могут служить: The UCI Machine Learning Repository [1], который является наиболее популярным среди исследователей благодаря своей классификации наборов данных, а также содержанием множества наиболее часто встречающихся выборок, XMLData Repository [2], Frequent Itemset Mining Dataset Repository [3]. При этом поиск необходимого набора данных под конкретную проблемную область достаточно затруднительный процесс. Ещё одной проблемой является трансформация данных в соответствии с требованиями решаемой задачи, а в статистических репозиториях эта проблема особенно актуальна. Также актуальным остается процесс обмена, поиска и предобработки статистических наборов данных, при этом такие возможности в существующих системах отсутствуют или не реализованы в достаточной мере.

## 2. Парадигма ИРННД (Интеллектуального Репозитория Научных Наборов Данных)

В настоящее время множество научных разработок ведется в области Semantic web [4], и данная парадигма и легла в основу «Интеллектуального репозитория научных наборов данных» (ИРННД). Одним из основных подходов

хранения данных является их онтологическое описание [5], что позволяет наделить огромные массивы данных, опубликованных в сети, большей осмысленностью, повысить удобство работы с информацией. Самыми распространенными форматами хранения онтологий является Resource Description Framework (RDF) и Ontology Web Language (OWL) [6]. RDF – язык описания ресурсов способом, "понятным" компьютеру на семантическом уровне. Ресурс – любая (физическая или абстрактная) сущность, имеющая уникальный идентификатор URI. Существуют различные подходы хранения онтологий: файловая система и базы данных. В представленной системе используется СУБД Oracle Database 10g и ее технология Oracle Spatial [7]. Oracle Spatial – это технология СУБД Oracle Database 10g, включающая дополнительные возможности по обработке пространственных данных для поддержки пространственных сервисов, различного рода приложений, предназначенных для обработки или предоставления информации о местонахождении объектов и других информационных систем. СУБД Oracle 10g включает поддержку RDF/RDFS, давая возможность разработчикам приложений использовать преимущества платформы семантической организации данных. Эта функциональность включает в себя SQL-уровень данных для хранения пространственных данных, поддержка SRARQL [6], новые операторы и функции для выполнения пространственных запросов и анализа данных.

Благодаря тому, что семантическое описание данных привносит структурированность и упрощает понимание данных агентами, агентные технологии и получили столь широкое развитие. Мультиагент – это аппаратная или программная сущность, способная действовать в интересах достижения целей, поставленных перед ним владельцем и/или пользователем. Классифицируются агенты на четыре основных типа: простые, умные (smart), интеллектуальные (intelligent) и действительно интеллектуальные (truly intelligent). Интерес для построения мультиагентных систем в задачах хранения и поиска данных представляют в большей степени интеллектуальные и действительно интеллектуальные агенты, которые отличаются тем, что поддерживают помимо автономного выполнения, взаимодействия с другими агентами и слежения за окружением – адаптивность поведения.

### 3. Представление системы

ИРННД предназначен для использования исследователями в области интеллектуального анализа данных и машинного обучения. Основной целью системы является хранение наборов данных, их классификация и поиск. Система представляет собой веб-приложение, использующее агентную платформу Jadex.

Варианты использования данной системы изображены на рис. 1.


*Рис.1. Варианты использования системы*

В свою очередь каждый их представленных вариантов использования может быть уточнен, используя несколько других вариантов использования.

Вариант использования «Регистрация» включает в себя следующие варианты использования: регистрация начинающего пользователя, регистрация продвинутого пользователя.

Вариант использования «Просмотр наборов данных» включает в себя следующие варианты использования: просмотр всех наборов данных, находящихся в системе; просмотр конкретного набора данных в расширенном виде: все метаданные, комментарии, оценки.

Вариант использования «Фильтрация наборов данных» включает в себя следующие варианты использования: фильтрация наборов данных по методу исследования, фильтрация наборов данных по параметрам метаданных.

Вариант использования «Менеджмент комментариев» включает в себя следующие варианты использования: редактирование комментария, удаление комментария.

Вариант использования «Менеджмент пользователей» включает в себя следующие варианты использования: редактирование данных о пользователе, удаление пользователей, изменение статуса пользователя, назначение пользователя администратором.

Вариант использования «Менеджмент наборов данных» включает в себя следующие варианты использования: редактирование метаданных о наборе данных, редактирование местоположения файла набора данных, удаление набора данных, добавление методов исследований для использования набором данных.

Вариант использования «Менеджмент методов исследований» включает в себя следующие варианты использования: добавление метода исследования, редактирование данных о методе исследования, удаление метода исследования, изменение в иерархической структуре методов исследования.

Все варианты использования системы распределяются между агентами. В данной системе разработаны агент пользователя, менеджер агент, агент ресурса и агент поиска, которые позволяют быстро, эффективно и оптимально осуществлять запрос пользователя.

ИРННД имеет следующие функциональные возможности: регистрация пользователей, аутентификация пользователей, менеджмент научных наборов данных: добавление новых выборок в систему, редактирование выборок, удаление выборок; менеджмент методов исследования: добавление новых методов, редактирование методов, удаление методов; поиск по онтологии наборов данных по критериям: тип задачи, предметная область, имя набора данных, ключевые слова, вид (шкала) входной/выходной характеристики, количество элементов (объем), дата (помещения в репозиторий, модификации, создания), автор задачи/набора данных/экспериментов, методы, какими решалась задача; эффективность работы набора данных. Добавление результатов работы с выборками, определение степени валидности набора данных.

Общая схема работы системы изображена на рис. 2.

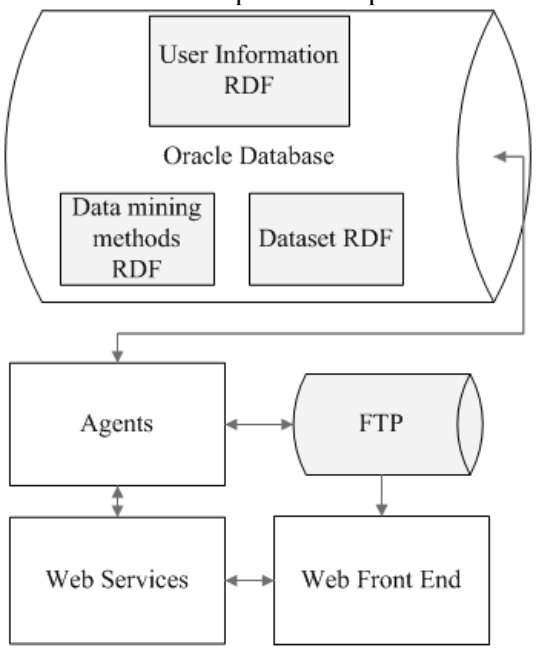


*Рис.2. Общая схема работы системы.*

Система состоит из следующих блоков: сервер баз данных, FTP сервер, веб-сервисы, агентная платформа Jadex, веб-интерфейс.

Сервер баз данных Oracle используется для хранения информации о пользователях системы, о методах исследований и мета данные для наборов данных.

FTP сервер используется для хранения файлов наборов данных.

Агентна платформа Jadex используется для проведения бизнес-логики системы.

Веб-сервисы используются для взаимодействия пользователя с веб-интерфейса с агентной платформой.

**4. Формальная модель онтологии**

Классическое понятие «онтология» для искусственного интеллекта, ввел Груббер в 1993 году, и которое подразумевает под собой явное описание и представление некоторой части концептуализации [8]. Формально онтология состоит из терминов, организованных в таксономию, их определений и атрибутов, а также связанных с ними аксиом и правил вывода.

Формальная модель онтологии (1) представляет собой упорядоченную тройку:

$$O = \langle \aleph, \Re, \Phi \rangle, \tag{1}$$

где $\aleph$ – конечное множество концептов (классов, понятий, терминов) предметной области, которую представляет онтология O;

$\Re$ – конечное множество отношений между концептами (понятий, терминов) заданной предметной области;

$\Phi$ – конечное множество функций интерпретации (аксиоматизации), заданных на концептах и/или отношениях онтологии O [9].

**5. Модель описания ресурсов**

RDF [6] – модель описания ресурсов (Resource Description Framework), это стандартная модель для обмена данными в сети. RDF содержит средства, которые облегчают объединение данных даже в тех случаях, когда схемы данных отличаются. RDF специально поддерживает изменения схем с течением времени, не требуя от потребителей данных внесения каких-либо изменений.

RDF расширяет структуру ссылок в Сети, позволяя использовать URI для обозначения именованных связей между сущностями аналогично тому, как обозначаются с помощью URI сами связываемые сущности (такую расширенную ссылку называют триплетом). Использование этой простой модели позволяет смешивать структурированные и полу-структурированные данные, публиковать их и разделять между разными приложениями.

Такая ссылочная структура образует направленный и помеченный граф, в котором ребра представляют собой именованные ссылки между ресурсами, образующими узлы графа. Представление в виде графа является наиболее простой для восприятия моделью RDF-данных и часто используется в ориентированных на простоту восприятия визуальных иллюстрациях.

Рабочая Группа W3C по Доступу к Данным разработала язык запросов SPARQL [6]. SPARQL определяет запросы в терминах шаблонов графа, которые сравниваются с направленным графом, представляющим собой RDF-данные. SPARQL включает в себя возможности по запрашиванию соответствия как необходимым, так и необязательным шаблонам, а также возможности объединения таких шаблонов с помощью логических операций конъюнкции и дизъюнкции ("и" и "или"). Результат сравнения также может быть использован для конструирования новых графов RDF с использованием отдельных шаблонов.

SPARQL может быть использован как часть программной среды общего назначения, такой как Jena, но запросы также могут быть отправлены в виде сообщений на удаленную точку доступа SPARQL при использовании вспомогательных технологий SPARQL-протокол и результаты запросов SPARQL в XML [6]. Используя такие точки доступа SPARQL, приложения могут запрашивать удаленные RDF данные и, даже, формировать новые RDF графы без какой-либо локальной обработки.

SPARQL – язык запросов, разработанный для модели данных RDF. Сами запросы выглядят и ведут себя как RDF, то есть запросы не зависят от физического представления RDF-данных (структуры базы данных, их представления в файле RDF/XML и т.д.) Если, например, запрос сделан через XQuery, приложение должно знать, каким образом эти конкретные данные представлены в RDF/XML (и это при том, что RDF/XML – только один из возможных форматов сериализации RDF-данных).

Текущая стандартизированная версия SPARQL позволяет только получать данные из RDF. Нет никакого эквивалента для SQL-операторов INSERT, UPDATE и DELETE. Большинство основанных на RDF приложений обрабатывают новые, изменяющиеся и утрачивающие актуальность данные с помощью API конкретной системы хранения RDF. Кроме того, данные RDF могут существовать виртуально (то есть создаваться по SPARQL-запросу). Существуют системы, которые создают данные RDF из других видов размеченных данных, таких как данные в форме Wiki-разметки или Atom Syndication Format.

Если посмотреть на список систем хранения RDF на сайте W3C, то он будет весьма и весьма обширным. Но для возможности использования в нашем проекте они должны отвечать двум основным требованиям: предоставлять API для работы с RDF на Java; развиваться (не быть устаревшей версией недоработанного проекта).

Этим требованиям соответствуют только Sesame и Jena [10].

### 6. Oracle Spatial

Первым масштабным проектом по реализации хранения онтологий в пространственном виде стала СУБД Oracle Database 10g. Oracle Spatial [7] – это технология СУБД Oracle Database 10g, включающая дополнительные возможности по обработке пространственных данных для поддержки пространственных сервисов, различного рода приложений, предназначенных для обработки или предоставления информации о местонахождении объектов и

других информационных систем. СУБД Oracle 10g включает поддержку RDF/RDFS, давая возможность разработчикам приложений использовать преимущества платформы семантической организации данных. Прикладные разработчики могут дополнять значение к данным и метаданным, определяя новые наборы термов и отношений между ними. Эти наборы термов (”онтологии”) более приспособлены для осуществления запросов и анализа, основанного на семантическом подходе, чем обычные наборы данных. Онтологические наборы данных, часто содержащие миллионы элементов данных и отношений между ними, которые могут быть сгруппированы в триплеты, используют новую RDF модель данных. Oracle допускает расширение биллионами триплетов для удовлетворения требований большинства приложений. Принципы хранения RDF в Oracle Spatial 10g: RDF данные хранятся как направленный, логический граф; субъекты и объекты отображаются как узлы, а предикаты как связи, у которых субъект является начальным узлом, а объект конечным; связи представляют собой полный RDF триплет; Oracle Spatial RDF Модель данных;

RDF модель данных поддерживает три типа объектов базы данных: модель (RDF граф, состоящий из набора триплетов), база правил (набор правил), индекс правила (направленный RDF граф).

Для осуществления семантических запросов используется оператор SDO_RDF_MATCH.

Основными преимуществами использования Oracle Spatial 10g являются: поддержка децентрализованного управления данными; поддержка всех RDF типов данных; SQL поиск и восстановление RDF моделей; осуществление запросов к RDF моделям, с использованием схемы графа; сочетание запросов RDF (SPARQL) с другими SQL операторами; логический вывод, основанный на RDFS (RDF схемы) правилах; логический вывод, основанный на правилах, определяемых пользователем в приложении.

**7. Суктура хранения данных**

В данной системе используется следующая структура хранения данных, представленная на рис.3.

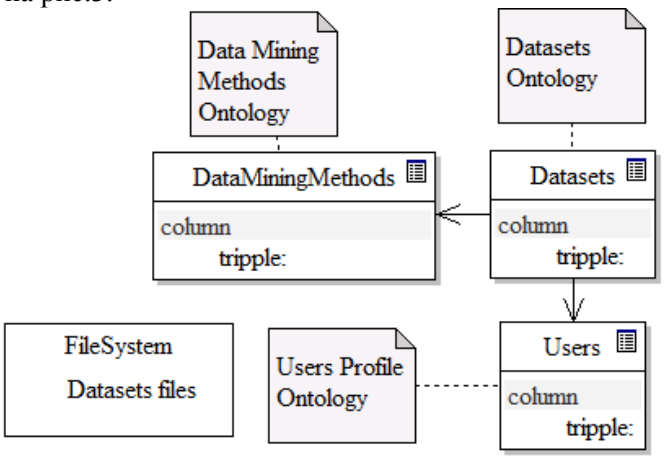


*Рис.3. Структура хранения данных.*

В базе данных существуют три таблицы для хранения основных данных системы: DataMiningMethods – для хранения данных о методах Data Mining, Datasets – для хранения данных о выборках, Users – для хранения данных о пользователях.

Рассмотрим онтологическую модель пользователя.

В табл.1 представлено описание слотов класса Address (адрес), который служит для базового описания адреса.

*Табл.1. – Слоты класса Address*

| Атрибут | Тип | Мощность | Наличие | Описание |
|---|---|---|---|---|
| country | String | 1 | Mandatory | Страна |
| city | String | 1 | Optional | Город |

В табл. 2 представлены слоты класса University (университет), который служит для базового описания университета.

*Табл.2. – Слоты класса University*

| Атрибут | Тип | Мощность | Наличие | Описание |
|---|---|---|---|---|
| name | String | 1 | Mandatory | Название университета |
| address | Address | 1 | Optional | Слот содержит класс с информацией об адресе университета |

В табл.3 представлены слоты класса Preference (предпочтения), который служит для описания региональных предпочтений пользователя, а также поисковых запросах.

*Табл.3. – Слоты класса Preference*

| Атрибут | Тип | Мощность | Наличие | Описание |
|---|---|---|---|---|
| language | String | 1 | Optional | Язык |
| format | String | 1 | Optional | Формат данных |
| search | String | * | Optional | Слот содержит множество поисковых запросов пользователя |

Слоты абстрактного класса Account представлены в табл. 4.

*Табл.4. – Слоты абстрактного класса Account*

| Атрибут | Тип | Мощность | Наличие | Описание |
|---|---|---|---|---|
| login | String | 1 | Mandatory | Логин |
| password | String | 1 | Mandatory | Пароль |
| created | String | 1 | Mandatory | Дата создания |
| email | String | 1 | Mandatory | Электронный адрес |
| preferences | Preference | 1 | Optional | Слот содержит класс с информацией о предпочтениях |
| title | String | 1 | Optional | Отображаемое имя |

Класс Account является базовым представлением пользователя.

Слоты абстрактного класса Person, для которого базовым классом является Account представлены в табл. 5.

*Табл. 5 – Слоты абстрактного класса Person*

| Атрибут | Тип | Мощность | Наличие | Описание |
|---|---|---|---|---|
| first_name | String | 1 | Optional | Имя |
| last_name | String | 1 | Optional | Фамилия |
| gender | Symbol (Male, Female) | 1 | Optional | Пол (мужской\женский) |
| university | University | 1 | Optional | Слот содержит класс с информацией об университете |

Для классов Beginner и Experienced базовым является класс Person. Класс Beginner не имеет слотов отличных от слотов класса Person. Слоты класса Experienced представлены в табл. 6.

*Табл. 6. – Слоты класса Experienced*

| Атрибут | Тип | Мощность | Наличие | Описание |
|---|---|---|---|---|
| speciality | String | * | Mandatory | Специализация |

**8. Подготовка базы данных для использования системы**

Система использует базу данных Oracle 10g версии 10.0.0.2. Для работы с возможностями RDF, использую библиотеку Jena на систему необходимо поставить последнее обновление.

Следующим шагом необходимо настроить СУБД для работы с пространственными данными. Первым шагом необходимо под пользователем, имеющим права SYSDBA выполнить следующие команды, изображенные на рис.4.

```
CREATE TABLESPACE rdf_tblspace
DATAFILE '/oradata/orcl/rdf_tblspace.dat'
SIZE 1024M REUSE AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED
SEGMENT SPACE MANAGEMENT AUTO;
EXECUTE SDO_RDF.CREATE_RDF_NETWORK ('rdf_tblspace');
```

*Рис.4. Установка поддержки пространственных данных*

Первая команда создает табличное пространство для последующего использования пространственных данных. Вторая команда настраивает систему для возможности использования пространственных данных.

Следующим шагом является создание таблиц и RDF моделей для каждой онтологии системы. Последовательность выполнения команд представлена на рис. 5.

```
CREATE TABLE users_rdf_data (id NUMBER, triple SDO_RDF_TRIPLE_S);
EXECUTE SDO_RDF.CREATE_RDF_MODEL ('users', 'users_rdf_data', 'triple');
CREATE TABLE datasets_rdf_data (id NUMBER, triple SDO_RDF_TRIPLE_S);
EXECUTE SDO_RDF.CREATE_RDF_MODEL ('datasets', 'datasets_rdf_data',
'triple');
CREATE TABLE methods_rdf_data (id NUMBER, triple SDO_RDF_TRIPLE_S);
EXECUTE SDO_RDF.CREATE_RDF_MODEL ('methods', 'methods_rdf_data',
'triple');
```

*Рис. 5. Создание таблиц*

После успешного выполнения этих команд, в каждую из моделей необходимо загрузить первоначальные данные, то есть схему описания онтологической модели. Для выполнения этого шага необходимо иметь схемы онтологий в формате N-TRIPLE. Так как для создания онтологий использовался Protege, то на выходе мы получаем онтологию в формате RDF. Для того, чтобы трансформировать онтологию в формате RDF в онтологию в формате N-TRIPLE использовался он-лайн конвертер RDF Converter [6]. Для загрузки данных в систему был написан скрипт, который на вход получал онтологию в виде файла, конфигурацией настраивалась строка подключения к серверу, имя онтологической модели, имя пользователя и пароль.

После успешной загрузки моделей СУБД готова к использованию системой.

**9. Выводы**

Система использует базу данных Oracle 10g версии 10.0.0.2. Для работы с RDF используется библиотека Jena. Затем устанавливается поддержка пространственных данных и создаются таблицы и RDF модели для каждой онтологической модели системы. После успешного выполнения этих команд, в каждую из моделей необходимо загрузить первоначальные данные, то есть схему описания онтологической модели. Для выполнения этого шага необходимо иметь схемы онтологий в формате N-TRIPLE. Так как для создания онтологий использовался Protege, то на выходе мы получаем онтологию в формате RDF. Для того, чтобы трансформировать онтологию в формате RDF в онтологию в формате N-TRIPLE использовался он-лайн конвертер RDF Converter. Для загрузки данных в систему был написан скрипт, который на вход получал онтологию в виде файла, конфигурацией настраивалась строка подключения к серверу, имя онтологической модели, имя пользователя и пароль.

В дальнейшем данную систему можно улучшить развитием входящих в нее онтологий и увеличением числа агентов.

Агенты, которые могли бы улучшить систему: агент предобработки данных, который бы конвертировал выборки в различные форматы; агент проверки набора данных, который бы проверял соответствие набора данных установленным для него методам исследования; агент поиска наборов данных в различных репозиториях наборах данных, который мог бы взаимодействовать с агентом предобработки данных перед загрузкой данных в систему.

Также можно добавить подсистему работы со статьями и результатами научных экспериментов исследователей, которые проводились с использованием наборов данных из репозитория.

ЛИТЕРАТУРА

1. The UCI Machine Learning Repository [Электронний ресурс]. – Режим доступа: www/ URL: http://archive.ics.uci.edu/ml/ – сайт, 2007.
2. XMLData Repository [Электронный ресурс]. – Режим доступа: www/ URL: http://www.cs.washington.edu/research/xmldatasets/ – сайт, 2008.

3. Frequent Itemset Mining Dataset Repository [Электронный ресурс]. – Режим доступа: www/ URL: http://fimi.cs.helsinki.fi/data/ – сайт, 2003
4. W3C Semantic Web Activity [Электронный ресурс]. – Режим доступа: www/ URL: http://www.w3.org/2001/sw/ – сайт, 2001.
5. Web Ontology Language OWL [Электронный ресурс]. – Режим доступа: www/ URL: http://www.w3.org/2004/OWL/ – сайт, 2004.
6. Beckett D., McBride B. RDF/XML Syntax Specification. RDF Primer [Электронный ресурс]. – Режим доступа: www/ URL: http://www.w3.org/TR/REC-rdf-syntax/. – Электронное издание, сайт, 2004. – HTML формат.
7. Oracle Database 10g Release 2 Spatial [Электронный ресурс]. – Режим доступа:www/URL:http://www.oracle.com/technology/products/spatial/10gr2_tech_info.html/ – сайт, 2009.
8. Овдей О.М. Обзор инструментов инженерии онтологий / О.М. Овдей, Г.Ю. Проскудина // Электронные библиотеки. – 2004. – Т. 7, вып. 4. – С. 3-19.10.
9. Гаврилова, Т. А. Базы знаний интеллектуальных систем: учеб. / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000. – 384 с.
10. Jena Semantic Web Framework [Электронный ресурс]. – Режим доступа: www/ URL: http://jena.sourceforge.net/ – сайт, 2009 .

Надійшла - 22.04.2010.