УДК 004.652

Ю.А. КУЛИК, А.С. КОВАЛЕВ

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

МЕТОДИКА СОЗДАНИЯ КОНТЕКСТНО-НЕЗАВИСИМОЙ СИСТЕМЫ ИНТЕГРАЦИИ ДАННЫХ

Предложена методика создания системы интеграции данных на основе хранения модели предметной области как данных в контекстно-независимых отношениях «объект», «атрибут объекта», «связь между объектами». При таком подходе сами данные представляются в отношениях«экземпляр объекта», «значение атрибута экземпляра», «связь между экземплярами». Такой подход позволяет изменять модель предметной области в процессе эксплуатации системы без изменения реляционных отношений, а также в силу контекстной независимости предоставляет возможность эксплуатации системы в различных областях. Для решения задачи извлечения данных в удобном для пользователя виде применяется интеграционная концепция Global as View (GAV), позволяющая переформулировать пользовательские запросы по предметной области в запросы к контекстно-независимым отношениям. Разработана база данных, реализующая предложенный подход и создано web-приложение электронного каталога для навигации по интегрированным данным.

Ключевые слова: *интеграция данных, модель предметной области, контекстно-независимая база данных, Global as View.*

Введение

Электронные библиотеки существуют уже давно и их количество все время увеличивается. Спрос на литературу в электронной форме также постоянно растет. Исторический обзор англоязычных и русскоязычных электронных библиотек и основные особенности их использования приведены в [1 – 3].

Для создания электронных каталогов и автоматизации труда библиотечных работников в традиционных библиотеках используется ряд библиографических информационных систем. Наиболее популярными среди них являются программные комплексы УФД/Библиотека и ИРБИС. Несомненно, современные библиотеки нуждаются в подобных системах. Однако общая тенденция такова, что возрастает роль полностью электронных библиотек, и это ставит ряд задач перед специалистами в области автоматизации.

Само по себе создание и поддержание электронной библиотеки не является новой задачей. Основные шаги и технические аспекты подробно описаны в [2, 4, 5]. Существует ряд готовых программных продуктов, позволяющих создавать электронную библиотеку визуально – Greenstone, IntraText, DSpace и многие другие.

Сложности начинаются, когда необходимо объединить данные из нескольких различных по своей структуре электронных библиотек, либо корректно загрузить данные из одной электронной библиотеки в другую. Разные электронные библиотеки могут иметь различную внутреннюю архитектуру и, соответственно, конвертация из одного формата в другой представляет трудоемкую задачу. Сложность также обусловлена тем, что полнота представления предметной области (библиографического описания) может сильно варьироваться в разных базах данных.

Задача интеграции данных в электронных библиотеках становится особенно актуальной в наше время, когда в электронном формате доступно множество различных документов, но навигация по ним на данный момент представляет трудоемкую задачу.

Существуют проекты, направленные на интеграцию данных в одном месте, такие как CompuLib. CompuLib представляет собой поисковую систему, которой индексированы библиографические описания из многих других электронных библиотек. Электронного каталога система не содержит и в этом заключается ее главный недостаток – навигация по библиографическому каталогу, в которой реализован только поиск, не предоставляет возможностей просмотра информации в структурированной форме.

Таким образом, на данный момент единого электронного каталога нет не только вцелом, но и по отдельным отраслям. Интеграция в основном обеспечивается наличием ссылок с одних электронных ресурсов на другие.

В контексте проектирования электронных библиотек актуальной задачей является создание такой модели данных, которую можно было бы произвольно модифицировать во время эксплуатации системы. Это нужно в первую очередь для того, чтобы

иметь возможность загрузки в систему данных произвольной структуры, полученных из внешних источников. Если модель предметной области (в данном случае предметная область – библиографическое описание электронного документа или другого ресурса) задана единожды при создании системы, то интегрировать данные, полученные из другого источника, будет проблематично. Придется либо импортировать только те данные, которые вкладываются в модель предметной области системы-приемника, либо переделывать эту модель. Естественно, при работе с различными источниками данных такие затруднения будут возникать постоянно.

Постановка задачи

Для создания универсальной интегрирующей среды необходимо разработать максимально гибкую контекстно-независимую модель предметной области. Такая система интеграции данных применима не только для создания электронной мета-библиотеки, но и для создания специализированных каталогов.

Из всего выше сказанного вытекает постановка задачи для создания информационной системы библиографического каталога.

Система должна предоставлять такие функции:

- импорт библиографических данных из внешнего источника с произвольной структурой хранения данных;
- возможность дополнять и изменять модель предметной области для обеспечения корректного импорта данных из новых источников;
- обеспечение удобной навигации по данным, имеющимся в узле.

Таким образом, в результате должна получиться масштабируемая система, назначением которой является предоставление удобного и однообразного доступа к библиографическим объектам, загруженным из некоторого количества внешних источников.

Разработка модели хранения интегрированных данных

Методику создания систем физической интеграции данных можно разделить на такие шаги:

- создание промежуточной модели предметной области как результат суперпозиции моделей интегрируемых источников;
- выбор способа физического хранения интегрированных данных;
- разработка алгоритмов извлечения данных из хранилища в удобном для пользователя виде для обеспечения навигации по данным;
- разработка алгоритмов загрузки новых данных в хранилище;
- программная реализация системы в виде базы данных и инструментальных средств для работы с данными.

В первую очередь необходимо создать некоторую промежуточную модель описания предметной области, к которой можно было бы привести все данные, содержащиеся во внешних источниках. Очевидно, что если задать такую модель единожды при проектировании системы, объединяя модели некоторого набора источников, то в дальнейшем может возникнуть потребность в ее изменении и дополнении. Поэтому необходимо предусмотреть расширяемость и гибкость промежуточной модели данных в проектируемой системе интеграции.

Внешние источники данных, как правило, представлены в виде реляционных баз данных с библиографической информацией, отдельных файлов табличной структуры (csv) или отдельных файлов иерархической структуры (xml).

Нетрудно показать, что любой табличный файл и любой иерархический файл можно преобразовать в набор отношений реляционной схемы данных. Поэтому в качестве обобщенного источника данных рассмотрим реляционную схему данных. Реляционная схема представляет собой совокупность отношений, характеризующихся набором атрибутов. Некоторые из атрибутов могут являться внешними ключами на другие отношения схемы. Таким образом, произвольный внешний источник представляется набором отношений:

$$\begin{aligned} R_A &= (A_1, A_2, \ldots, A_n, FK_{AB}), \\ R_B &= (B_1, B_2, \ldots, B_m), \\ R_C &= (C_1, C_2, \ldots, C_k, FK_{CA}, FK_{CB}), \end{aligned} \quad (1)$$

где R – отношения, некоторые объекты предметной области, например, книги, авторы, издательства;

A, B, C – атрибуты отношений, некоторые признаки, присущие этим объектам;

FK – внешние ключи, связи между отдельными экземплярами объектов.

Можно представить это описание предметной области как граф, вершинами которого являются отношения и атрибуты, а дуги определяют принадлежность атрибутов отношениям и взаимосвязи между отношениями (внешние ключи). Пример такого графа показан на рис. 1.

Для каждого источника данных можно построить такой граф, который будет отражать полноту представления предметной области источником. Естественно, графы разных источников будут пересекаться. Возможны ситуации, когда одно и то же понятие в предметной области в одной модели будет отношением, в другой – атрибутом. Или в одной схеме понятие будет декомпозировано на несколько атрибутов, а в другой будет представлено одним.

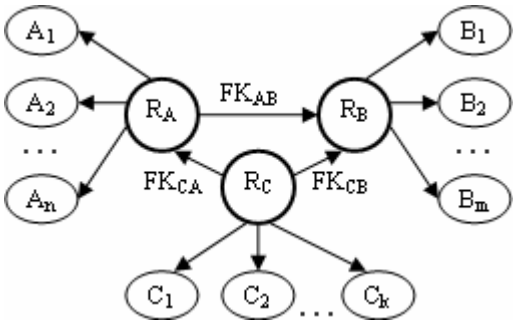

Рис. 1. Граф модели предметной области

Чтобы интегрировать данные, необходимо сначала создать модель предметной области, к которой приводить все остальные модели отдельных источников. Такая модель называется промежуточной схемой данных (mediated schema). Это трудоемкая не автоматизируемая задача, включающая анализ самой предметной области и степени полноты ее представления в различных источниках, выявление основных общих понятий. Результатом является модель предметной области, в терминах которой будут формулироваться запросы пользователей системы интеграции.

Пример схемы предметной области библиографических данных показан на рис. 2.

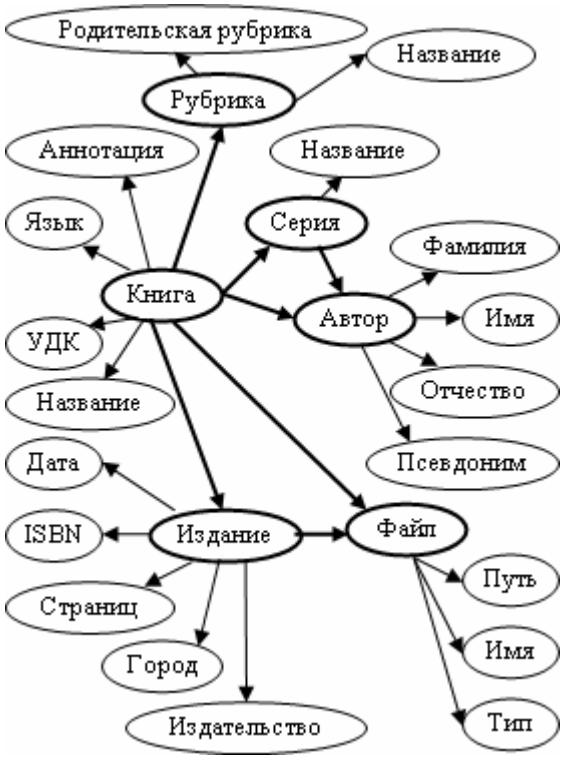

Рис. 2. Пример модели предметной области

Основной идеей разработки является универсализация хранения промежуточной модели. Так как необходимо обеспечить гибкость и модифицируемость промежуточной схемы данных, привязывать отношения к конкретным сущностям предметной области нельзя. Вместо этого предлагается хранить модель предметной области в терминах «объект», «связь», «атрибут». При этом сами данные, соответственно, представляются как «экземпляр объекта», «экземпляр атрибута», «экземпляр связи». Таким образом, схема данных для создаваемой системы интеграции состоит из двух наборов отношений. Один из наборов предназначен для описания модели предметной области, второй – для хранения самих данных.

Реляционную схему интеграционной системы можно представить в следующем виде:

$$\begin{aligned} &IS = < O, A, R, O', A', R' >, \\ &O = (O\_id, O\_Name), \\ &A = (A\_id, O\_id, A\_Name), \\ &R = (R\_id, O1\_id, O2\_id), \\ &O' = (O\_id, O'\_id), \\ &A' = (A\_id, O'\_id, A'\_Value), \\ &R' = (R\_id, O1'\_id, O2'\_id), \end{aligned} \quad (2)$$

где IS – реляционная схема разрабатываемой системы интеграции;

O – отношение для хранения объектов предметной области, таких, как «Книга», «Серия»;

A – для хранения набора атрибутов объектов;

R – возможные связи между объектами («Книга – автор», «Книга – серия»);

O' – отношение для хранения экземпляров объектов;

A' – отношение для значений атрибутов конкретного экземпляра;

R' – для хранения связи между двумя экземплярами.

Такой подход к хранению данных полностью соответствует требованиям, сформулированным в постановке задачи – для изменения модели предметной области не нужно изменять реляционную схему, достаточно вставить, удалить или изменить кортежи в существующих отношениях. Еще одним преимуществом кроме гибкости является универсальность такой системы – можно задать любую предметную область, что расширяет область возможного применения для разрабатываемой системы.

Извлечение данных

Недостатком предлагаемого подхода является алгоритмическая сложность извлечения данных и преобразования их в вид, удобный для пользователя. Пользовательские запросы должны формулироваться в терминах промежуточной модели данных, а не в терминах «объект», «атрибут», «связь».

Для решения этой проблемы применяется подход «Global as View» (GAV), разработанный в рамках теории интеграции [6,7]. Этот подход предназначен для систем виртуальной интеграции и заключается в задании правил переформулирования пользовательских запросов из промежуточной схемы в схемы интегрируемых источников данных. Правила представляют собой хорновские дизъюнкты и описывают виртуальные отношения промежуточной схемы.

В контексте разрабатываемой системы применение этого подхода означает, что данные, распределенные по набору контекстно-независимых отношений, необходимо интегрировать в отношения промежуточной схемы предметной области.

Для предметной области библиографической информации можно привести следующий пример использования подхода GAV. Предположим, в промежуточной схеме задано отношение принадлежности книги автору

$$book\_author(book.title, author.name). \qquad (3)$$

Тогда для пользовательского запроса «Получить все названия книг X, автор которых – Пушкин» Виртуальное отношение book_author будет переформулировано следующим образом:

$$\begin{aligned} &book\_author(X,"Пушкин") \subset O'(O\_id, O'\_id) \wedge \\ &\wedge O(O\_id,"book") \wedge \\ &\wedge A'(A\_id, O'\_id, X) \wedge \\ &\wedge A(A\_id, O\_id,"book.title") \wedge \\ &\wedge O'(O2\_id, O2'\_id) \wedge \\ &\wedge O(O2\_id,"author") \wedge \\ &\wedge A'(A2\_id, O2'\_id,"Пушкин") \wedge \\ &\wedge A(A2\_id, O2\_id,"author.name") \wedge \\ &\wedge R'(R\_id, O'\_id, O2'\_id) \wedge \\ &\wedge R(R\_id, O\_id, O2\_id). \end{aligned} \qquad (4)$$

Смысл этой записи состоит в следующем: должен существовать экземпляр O'_id объекта "book", которому должен соответствовать экземпляр атрибута "book.title" со значением X; также должен существовать экземпляр объекта "author" O2'_id со значением атрибута "author.name", равным "Пушкин"; также должна существовать связь между этими экземплярами O'_id и O2'_id.

Результаты

Результатом проектирования является база данных в системе управления MySQL. Ее ER-диаграмма показана на рис. 3.

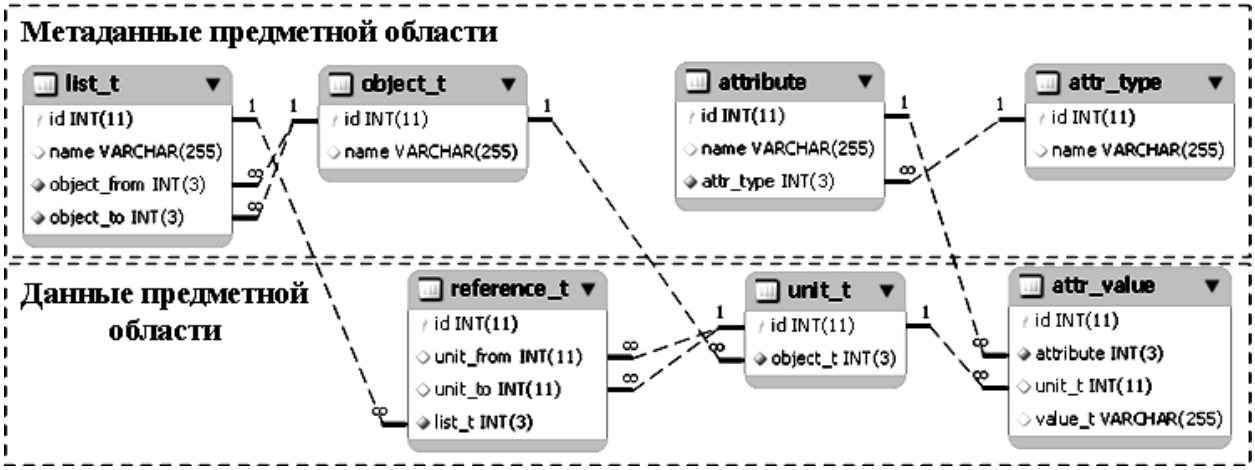

Рис. 3. ER-диаграмма базы данных системы

В таблице object_t хранится набор названий объектов, с которыми работает система. Это такие общие понятия, как книга, персона, файл и т.п.

В list_t задаются все возможные варианты связей между экземплярами объектов. Например, связь между объектом книга и объектом персона, или между пользователем и коллекцией.

В таблице attribute хранится список названий атрибутов объектов.

В attr_type хранятся возможные типы атрибутов экземпляров объектов (число, текст, и другие). Эта таблица нужна для задания различных форматов отображения для различных типов атрибутов.

В таблице unit содержатся номера конкретных экземпляров объектов – конкретные персоны, книги.

В reference хранится список всех связей между экземплярами, которые присутствуют в базе данных. Каждая строка этой таблицы указывает на объекты, которые связаны (на пример, книга и автор) и на тип связи, определенный в таблице list_t.

Значения атрибутов конкретных экземпляров объектов (из таблицы unit_t) хранятся в таблице attr_value.

Концепция GAV реализована при помощи реляционных представлений (view) в базе данных. Пример такого представления для отношения «серия» представлен ниже.

```
CREATE VIEW `vw_series` AS
select `u`.`id` AS `id`,`a51`.`value_t` AS `name`
from `unit_t` `u` join `attr_value` `a51` on(`u`.`id` = `a51`.`unit_t`)
where (`u`.`object_t` = 6) and (`a51`.`attribute` = 51);
```

Объект под номером 6 – «Серия», атрибут под номером 51 – «Название серии».

Для навигации по данным, хранящимся в базе, создано web-приложение электронного каталога на языке php. Приложение получает данные из базы через представления предметной области, реализуя тем самым концепцию GAV – пользовательские запросы, сформулированные в терминах предметной области, переводятся в запросы к контекстно-независимым таблицам базы данных при помощи правил, заданных в представлениях.

Экранная форма каталога, в котором используются построенные представления для описания предметной области, показана на рис. 4. Каталог позволяет искать книги по авторам, сериям и рубрикам. Также можно искать книги по алфавиту или пользоваться формой поиска по шаблону.

Импорт данных осуществляется при помощи специального программного модуля, на вход которого поступают данные из внешних источников, и правила их преобразования к промежуточной схеме. Для каждого источника задается свой набор правил.

Данная статья направлена на общее описание построения системы интеграции, поэтому механизм импорта здесь подробно не описывается.

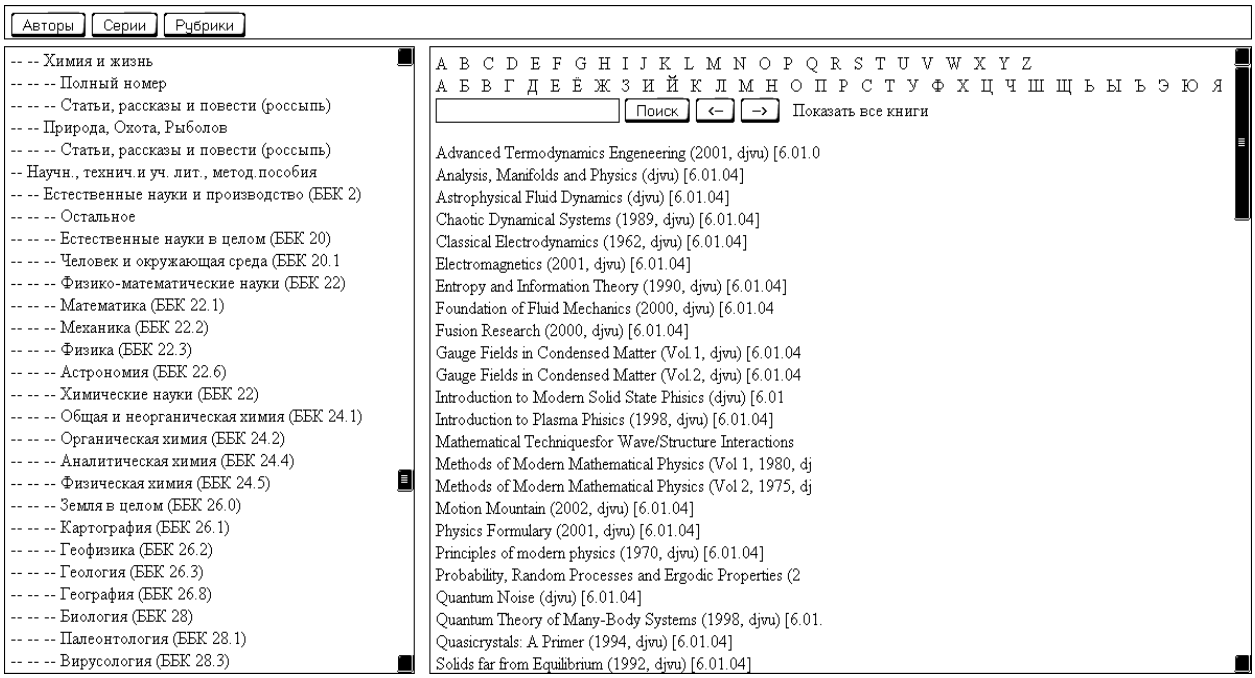

Рис. 4. Экранная форма разработанного каталога

Заключение

Предложена методика создания системы физической интеграции данных, основанная на контекстно-независимых отношениях, в которой промежуточная модель предметной области задается в качестве данных в терминах «объект», «атрибут объекта», «связь между объектами». Сами интегрируемые данные, соответственно, представляются как «экземпляр объекта», «значение атрибута экземпляра», «связь между экземплярами». Такая схема данных позволяет при необходимости изменять и дополнять модель предметной области, также отсутствует привязка к конкретной предметной области, что дает возможность использовать систему при работе в различных областях.

Для извлечения данных используется концепция виртуальной интеграции GAV, позволяющая переформулировать пользовательские запросы в терминах предметной области в запросы к контекстно-независимым отношениям в предложенной схеме.

Разработана база данных, реализующая предложенный подход и web-приложение каталога, обеспечивающее навигацию по интегрированным данным.

Дальнейшим направлением исследований является алгоритмизация обработки данных для выявления и слияния тождественных элементов данных.

Литература

1. Greenstein Daniel The Digital Library: A Biography / Daniel Greenstein, Suzanne E. Thorin. - Washington, DC: Digital Library Federation, 2002. – 76 p.

2. Митчелл Энн. Каталогизация и организация электронных ресурсов: практическое руководство для библиотекарей / Энн Митчелл, Брайан Саррэтт. – М.: Омега-Л, 2008. – 231 с.

3. Антопольский А.Б. Правовые и технологические проблемы создания и функционирования электронных библиотек / А.Б. Антопольский, Т.С. Маркарова, Е.А. Данилина. – М.: ИНИЦ «Патент», 2008. – 207 с.

4. Designing and Building Integrated Digital Library Systems – Guidelines / By Bente Dahl Rathje, Margaret McGrory, Carol Pollitt, Paivi Voutilainen; under the auspices of the IFLA Libraries for the Blind Section The Hague, IFLA Headquarters, 2005. – 67 p.

5. Witten Ian. How to Build a Digital Library / Ian Witten, David Bainbridge. – San Francisco: Morgan Kaufmann Publishers, 2003. – 34 p.

6. Levy Alon Answering Queries Using Views: A Survey / Alon Levy // The VLDB Journal. – 2001. – №10. – P. 270 – 294.

7. Logic Based Artificial Intelligence / ed. Jack Minker. – Kluwer Publishers, 2000. – 595 p.

Поступила в редакцию 4.12.2010

Рецензент: д-р техн. наук, проф., зав. каф. информатики А.Ю. Соколов, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

МЕТОДИКА СТВОРЕННЯ КОНТЕКСТНО-НЕЗАЛЕЖНОЇ СИСТЕМИ ІНТЕГРАЦІЇ ДАНИХ

Ю.О. Кулик, А.С. Ковальов

Запропоновано методику створення системи інтеграції на основі зберігання моделі предметної області як даних в контекстно-незалежних відношеннях «об'єкт», «атрибут об'єкту» «зв'язок між об'єктами». За такого підходу самі дані представляються у відношеннях «екземпляр об'єкту», «значення атрибута екземпляра», «зв'язок між екземплярами». Такий підхід дозволяє змінювати модель предметної області в процесі експлуатації системи без зміни реляційних відношень, а також через контекстну незалежність надає можливість експлуатації системи в різних областях. Для вирішення задачі видобування даних у зручному для користувача вигляді застосовується інтеграційна концепція Global as View (GAV), що дозволяє переформулювати запити користувача по предметній області в запити до контекстно-незалежних відношень. Розроблено базу даних, що реалізує запропонований підхід і створено web-програму електронного каталогу для навігації по інтегрованим даним.

Ключові слова: інтеграція даних, модель предметної області, контекстно-незалежна база даних, Global as View.

METHODOLOGY OF CREATING CONTEXT-FREE DATA INTEGRATION SYSTEM

Yu.O. Kulik, A.S. Kovalov

The methodology of creating data integration system on the basis of storing the domain model as data in context-free relations "object", "attribute", "link between objects" is offered. According to this approach the data itself is represented in relations "instance of an object", "instance attribute value", "link between instances". This approach enables to change the domain model during system exploitation without any changes to relations and also gives ability to use the system in different domains. To solve the problem of data extraction the Global as View (GAV) integration concept is used. It is used to reformulate user queries in terms of the domain into queries to context-free relations. The database in which this approach is used is created and the web-application for navigating the digital catalogue is developed.

Keywords: data integration, domain model, context-free database, Global as View.

Кулик Юрий Алексеевич – канд. техн. наук, доцент, доцент каф. информационных управляющих систем, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.

Ковалев Андрей Сергеевич – магистрант каф. информационных управляющих систем, Национальный аэрокосмический университет им. Н.Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.