УЧЕНЫЕ ЗАПИСКИ КАЗАНСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
Том 149, кн. 2 Физико-математические науки *2007*

УДК 004.6

ОНТОЛОГИЧЕСКИЙ ПОДХОД К СОЗДАНИЮ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПО КУЛЬТУРНОМУ НАСЛЕДИЮ

В.В. Иванов

Аннотация

В статье описана технология, разработанная при создании информационной системы по культуре. Ключевым аспектом является использование онтологического подхода. Технология допускает адаптацию к близким предметным областям. Освещаются вопросы создания прикладной онтологии по культуре с учетом двух задач: поддержки интеграции разнородных источников данных и осуществления интеллектуального поиска в их содержимом. Далее описываются два приложения созданной онтологии: для интеграции музейных описаний и для поиска описаний музейных предметов в объединенном хранилище. В заключение описывается прототип программной системы, демонстрирующий реализацию предлагаемой технологии на практике.

Введение

Целью описываемой в статье работы является создание информационной системы для доступа к культурному наследию России. Основное назначение системы состоит в предоставлении доступа к информации из открытых музейных источников, баз данных и т. п.

На данный момент аналогов создаваемой системы в России не существует. Основными источниками исходных данных по культуре являются музеи. Современные музейные информационные системы (ИС), как правило, ориентируются на поддержку внутренних функций: учет фондов, сверка и т. п. Значительно меньше уделяется внимания функциям, которые музей выполняет в информационном окружении. Предполагается, что создаваемая система будет использоваться для свободного доступа внешних пользователей (потенциальных посетителей, туристов, исследователей) к информации по культурному наследию, представленному в музеях.

При проектировании архитектуры единой информационной системы возникает ряд трудностей.

Многие музеи имеют собственные базы данных (БД). Эти БД относятся к одной, хотя и весьма размытой, предметной области (материальная культура), но при этом они не являются структурно однородными. Неоднородность появляется из-за различных подходов к проектированию схемы БД.

Чтобы пояснить, что подразумевается под структурной неоднородностью, рассмотрим, как описывается информация о материале, из которого сделан некоторый музейный предмет, на примере трех различных музейных БД.

1) БД Этнографического музея Казанского университета – http://www.ksu.ru/emku/ (стандарт описания предложен Российским Этнографическим музеем). Таблица *Предметы*, помимо прочих полей (номер объекта, название, дата создания и др.) содержит столбец *material*, в каждой ячейке которого через запятую могут

быть перечислены названия нескольких материалов, например, «дерево, краски, бумага».

2) БД Всероссийского реестра музеев (http://www.museum.ru). Данные по материалам и технике изготовления записываются в одно поле: «Материал и Техника». Поскольку исторически поля «материал» и «техника» указывались вместе в книгах поступлений, в БД Всероссийского реестра музеев эти два поля также «склеены» в одно.

3) БД Музея истории Казанского университета (ИС «НИКА-Музей»). В БД этой системы проведена необходимая нормализация: для сущностей «предметы» и «материалы» созданы 2 таблицы. Отношение типа «многие ко многим» между этими сущностями реализовано как отдельная таблица, которая содержит ссылки на значения первичных ключей в таблицах *Предметы* и *Материалы*.

Встречаются и более сложные случаи. Например, информация о сохранности того или иного предмета описывается в поле «Сохранность» двумя основными значениями «полная» и «неполная». Однако в это поле может быть записан и произвольный текст, точно описывающий вид повреждений, загрязнений и т. п. (при этом подразумевается неполная сохранность). В первом случае наблюдается расщепление термина между именем поля и значением в этом поле. Во втором случае текстовое значение, записанное в поле, влечет неявное упоминание другого связанного термина.

Структурная неоднородность может проявляться даже, если два музея используют одну и ту же ИС, но при этом отличаются по размерам, структуре фондов, тематическому уклону коллекций. Безусловно есть и другой вид неоднородности: различные информационные системы могут использовать для хранения данных различные СУБД. Для интеграции на этом уровне существуют стандартные технологии, например, ODBC[1], HIBERNATE[2]. Но очевидно, что структурная неоднородность схем баз данных является первичной и останется даже в случае работы с одной единственной СУБД.

Другой проблемой является отсутствие общепринятой терминологии при описании значений свойств музейных предметов. Например, вместо значения «полная сохранность» может быть указано «хорошая сохранность». Несмотря на то, что и в России и за рубежом предпринимались попытки разработать стандартные справочники и тезаурусы в области культуры, в музеях подобные стандарты практически не используются. Каждый специалист предпочитает пользоваться собственной «единственно верной» системой терминов.

Описанные проблемы приводят к двум связанным задачам, которые необходимо решать при создании информационной системы для доступа к разнородным базам данных музеев:

1) задача преодоления структурной неоднородности схем БД;

2) задача гибкой адаптации к многообразию терминологических систем, принятых и уже используемых в различных музеях.

Первая задача известна как задача интеграции разнородных источников данных. Вторая задача более сложная, и на необходимость ее решения указывается реже. Основная трудность здесь состоит в согласовании различных, возможно, противоречащих друг другу систем понятий. Представляется, что успеха при решении этой задачи можно достичь, допуская независимое существование таких систем понятий.

Очевидно, что архитектура создаваемой системы должна использовать единый механизм доступа, основанный на некоторой общей разделяемой концептуальной

[1]http://uda.openlinksw.com/odbc/
[2]http://www.hibernate.org

схеме, к которой можно привести подавляющее большинство схем музейных БД. Перспективным является применение онтологического подхода как для определения единой концептуальной схемы, так и для структурирования понятий в предметной области и формализации системы терминов естественного языка.

В статье изложены базовые аспекты технологии построения информационной системы по культуре с использованием онтологического подхода. Технология описывается на довольно высоком уровне абстракции, что допускает ее адаптацию к другим предметным областям. Первый раздел статьи посвящен процессу создания прикладной онтологии по культуре с учетом двух задач: поддержка интеграции разнородных источников данных и осуществление интеллектуального поиска в их содержимом. Второй и третий разделы статьи описывают два приложения созданной онтологии: интеграция музейных описаний и информационный поиск в объединенном хранилище. В четвертом разделе описывается прототип программной системы, демонстрирующий реализацию предлагаемой технологии на практике.

1. Создание онтологии по культуре

Поскольку отправной точкой при создании информационной системы является онтология по культуре, возникает необходимость разработать новую или выбрать уже существующую. Наиболее важным критерием при выборе онтологии является ее назначение.

1.1. Назначение онтологии. С помощью прикладной онтологии по культуре предполагается:

1) осуществлять навигацию и поиск в структурированных источниках информации (например, в базах данных музеев);

2) интегрировать разнородные информационные ресурсы в области культурного наследия.

Как показывает практика, для успешного решения первой задачи необходимо использовать онтологию размером в несколько тысяч понятий, тесно связанных с языком, на котором будет осуществляться поиск. Для решения второй задачи необходимо иметь возможность описывать в терминах онтологии разнообразные факты, извлекаемые из множества разнородных источников информации. В настоящее время неизвестно, существуют ли ресурсы с подобными параметрами. Поэтому в качестве основы новой онтологии целесообразно использовать уже имеющиеся ресурсы, поскольку разрабатывать масштабную онтологию «с нуля» неэффективно. В качестве общей концептуальной модели предметной области обычно используется некоторая онтология верхнего уровня. Для расширения системы понятий онтологии предлагается использовать тезаурусы: терминологические системы, охватывающие часть лексики естественного языка и имеющие при этом свойства формальных онтологий.

Далее кратко рассматриваются два тезауруса в области культуры и одна онтология верхнего уровня, обосновывается выбор их в качестве базы для построения прикладной онтологии по культуре.

1.2. Онтология CIDOC CRM. В качестве онтологии верхнего уровня выбрана онтология CIDOC Conceptual Reference Model (CRM) [1]. Модель CRM, разработанная Международным советом музеев, имеет статус стандарта ISO 21127. CRM содержит наиболее общие концепты и отношения, встречаемые в области музейной документации: 81 концепт и около 140 бинарных отношений. Имена понятий и свойств CRM переведены на английский, русский, немецкий, французский

и греческий языки. Онтология не зависит от языка представления и может быть выражена с помощью различных формализмов представления знаний.

Концептуальная модель CIDOC CRM является формальной онтологией верхнего уровня, которая предназначена для описания фактов исторического, географического и теоретического характера об отдельных экспонатах и музейных коллекциях в целом, а также для интеграции информации по культурному наследию.

Структурно CRM состоит из иерархии классов и широкого набора свойств, связывающих классы между собой.

Основой для расширения онтологии CRM являются следующие лингвистические ресурсы: тезаурус по вопросам культуры и музейного дела (далее – тезаурус CULTHES) и тезаурус AAT (The Art and Architecture Thesaurus, далее – тезаурус AAT), переведенный на русский язык.

1.3. Тезаурус CULTHES. Русскоязычный информационно-поисковый тезаурус CULTHES, разработанный Санкт-Петербургским университетом искусства и культуры, содержит около 8 тыс. лексических единиц (ЛЕ). Все ЛЕ тезауруса разделяются на два вида: дескрипторы и аскрипторы. Дескриптор – это лексическая единица, определяющая некоторое понятие из предметной области. Аскриптор – синонимичный вариант дескриптора. Иногда аскрипторы могут представляться комбинацией двух и более дескрипторов. Лексические единицы тезауруса могут быть связаны друг с другом следующими видами отношений:

– гипонимия (связь ВР–НВ, или ВЫШЕ-РОД–НИЖЕ-ВИД);
– меронимия (связь ВЦ–НЧ, или ВЫШЕ-ЦЕЛОЕ–НИЖЕ-ЧАСТЬ);
– синонимия (связь С–СМ, идущая от аскриптора к дескриптору);
– ассоциация (связь А);
– служебные связи для поддержания целостности и пр.

Дескрипторы тезауруса группируются по некоторому общему признаку, который отражает общий смысл всех дескрипторов группы, в непересекающиеся подмножества – фасеты. Каждый фасет содержит подгруппы – дескрипторные блоки (ДБ). Внутри ДБ дескрипторы организованы в иерархию (по отношению гипонимии).

Несмотря на то, что некоторая часть лексических единиц тезауруса устарела, существенная часть терминологии может быть использована. При создании онтологии по культуре используется около 6500 дескрипторов тезауруса CULTHES.

1.4. Тезаурус AAT. Тезаурус AAT создан и развивается фондом П. Гетти для описания предметов материальной культуры [2]. Перевод AAT и его адаптация к русскому языку и русской культуре выполнены в Научно-исследовательском вычислительном центре Московского государственного университета [3].

В настоящее время тезаурус AAT содержит около 30 тыс. дескрипторов и более 130 тыс. англоязычных терминов. Терминология тезауруса охватывает искусство, архитектуру, декоративное искусство, материальную культуру, архивные материалы с античности до наших дней.

Лексические единицы тезауруса AAT (дескрипторы и связанные с ними аскрипторы) также разделены на непересекающиеся множества – фасеты. В AAT имеется 7 фасетов: Ассоциированные понятия, Физические свойства, Стили и Периоды, Агенты (люди и организации), Деятельность, Материалы, Объекты. Внутри каждого фасета дескрипторы могут быть сгруппированы по некоторому общему признаку (например, по выполняемым функциям) в более мелкие *дескрипторные блоки.*

Тезаурус AAT вместе со списком имен деятелей культуры ULAN и списком географических названий TGN образует широчайшую информационную базу по искусству и архитектуре западной Европы. Несмотря на это, тезаурус имеет следующие недостатки.

1) Понятия AAT, ULAN и TGN привязаны к культуре Европы, в то время как создаваемая онтология ориентируется в первую очередь на российскую специфику.

2) AAT имеет сильно разветвленную иерархию понятий. Излишняя детализация запутывает неспециалистов и редко бывает востребована музейными работниками при описании музейных предметов, написании текстов или заполнении полей баз данных.

Преимущество AAT состоит в том, что он является наиболее полным среди тезаурусов по культуре и оптимально подходит для индексирования текстов.

При построении прикладной онтологии по культуре модель CRM использовалась в роли онтологии верхнего уровня, а тезаурус CULTHES – для расширения набора понятий онтологии и более точного описания их значения. Планируется также подключить к онтологии и русскоязычную версию тезауруса AAT.

1.5. Процесс наполнения онтологии верхнего уровня лексикой тезауруса. Онтология CRM имеет средства для гибкого расширения иерархии понятий при помощи встроенного класса E55_Тип. Экземплярами этого класса становятся понятия, внутренняя структура которых не представляет интереса, но которые сами по себе могут описывать элементы данных.

Понятия тезауруса формируют определенную систему типов или категорий, которая может быть подключена к онтологии. Примерами таких понятий служат ЛЕ, относящиеся к видам, жанрам искусства, материалам, способам и технике обработки, типам художественных произведений. В рамках этой системы типов может уточняться описание экземпляров некоторого класса онтологии верхнего уровня. Иерархия понятий тезауруса может быть включена в CRM с использованием подклассов и свойств класса E55_Тип. Объединение тезауруса и онтологии сводится к добавлению понятий из тезауруса в онтологию и связывании их соответствующими отношениями друг с другом и с классами CRM. Этот процесс состоит в последовательном выполнении нескольких этапов.

1) *Привязка фасетов тезауруса к подклассам класса E55_ Тип.*

На этом этапе для каждого фасета тезауруса создается подкласс класса E55_Тип, на который затем будут отображаться дескрипторные блоки фасета.

2) *Отображение дескрипторных блоков на подклассы класса E55_ Тип.*

Внутри каждого фасета дескрипторные блоки организованы в иерархию. Для каждого ДБ в онтологии создается отдельный класс. Иерархические связи между дескрипторными блоками переносятся на иерархические связи между соответствующими классами. Дескрипторные блоки верхнего уровня становятся подклассами класса-фасета, созданного на предыдущем этапе. Все дескрипторы, входящие в некоторый ДБ, становятся экземплярами нового класса в результирующей онтологии.

Класс E55_Тип имеет свойство «P2B_является_типом_чего-либо», которое может связывать экземпляры класса E55_Тип с экземплярами любого класса онтологии. Именно это свойство позволяет гибко уточнять тип экземпляра любого класса онтологии, используя, внешние источники терминов. Поскольку дескрипторные блоки создаются как подклассы класса E55_Тип, значение свойства P2B может быть ограничено так, что экземпляры некоторого класса общей онтологии CRM будут иметь связи вида P2B строго с лексическими единицами заданных ДБ.

Рис. 1. Фрагмент иерархии классов онтологии по культуре. В иерархии выделен класс-ДБ «Материалы произведений искусства» и один из его экземпляров-дескрипторов «БЕРЕЗА»

3) *Отображение связи ВЫШЕ-РОД–НИЖЕ-ВИД между дескрипторами.*

Как отмечалось выше, между дескрипторами тезауруса могут быть установлены различные виды связей. В первую очередь при расширении онтологии рассматривается гипонимия (или связь ВЫШЕ-РОД–НИЖЕ-ВИД) и синонимия. Любой дескриптор (кроме корневого дескриптора фасета) имеет связь с вышестоящим дескриптором – родовым понятием. Связь ВЫШЕ-РОД–НИЖЕ-ВИД отображается на пару взаимообратных свойств «P127F_имеет_вышестоящий_термин» и «P127B_имеет_нижестоящий_термин», которые увязывают новые экземпляры (дескрипторы) класса E55_Тип друг с другом.

4) *Отображение отношения синонимии.*

Для каждого термина, являющегося синонимом (или синонимичным вариантом) некоторого понятия, в онтологии создается новое литеральное значение, которое привязывается к экземпляру-дескриптору с указанием естественного языка, на котором выражен этот синоним.

После выполнения перечисленных этапов онтология CRM расширяется набором новых понятий, увязанных иерархическими отношениями: классы, созданные из ДБ, увязаны в иерархию отношения КЛАСС-ПОДКЛАСС, а экземпляры этих классов, полученные из дескрипторов, – в иерархию транзитивных свойств P127F–P127B. Кроме того, дескрипторные блоки служат ограничителями на значение свойства P2B, и каждый дескриптор имеет один или более текстовых вариантов (синонимов). Этого уровня достаточно для приложений, в которых планируется применять расширенную онтологию. Вид онтологии по культуре, полученной после расширения модели CIDOC CRM понятиями тезауруса CULTHES, показан на рис. 1.

Таким образом, описан способ построения прикладной онтологии по культуре, верхний уровень которой содержит общие понятия и отношения из области культурного наследия, а нижний уровень (система типов) состоит из лексических единиц и дескрипторных блоков тезауруса по искусству и музейному делу. При описании объектов, событий или процессов также выделяются два уровня. Общий уровень использует только понятия и свойства онтологии верхнего уровня. На уровне лексики предметной области описание объекта, процесса или события может быть уточнено ссылками на лексические единицы тезауруса.

Процесс наполнения онтологии лексикой тезауруса не вносит концептуальных противоречий в онтологию, поскольку иерархия понятий внутри тезауруса определена корректно. Дескрипторный блок тезауруса интерпретируется как некоторый класс объектов реального мира, что совпадает с интерпретацией класса внутри онтологии. Дескрипторы тезауруса интерпретируются как некоторые типы объектов, что соответствует их положению в прикладной онтологии (как экземпляров класса E55_Тип). Данный подход также согласуется с пониманием смысла слова в лексической семантике [4, с. 128–132]: каждый дескриптор тезауруса именует некоторое множество объектов внешнего мира (референтов). Дескриптор описывает интенсионал этого множества, а элементы множества определяются в онтологии как экземпляры классов верхнего уровня.

Среди задач, в которых использование онтологий может дать наибольший эффект, обычно называются интеграция данных и поиск информации [5, p. 460–494].

Далее рассматриваются приложения построенной онтологии для решения этих задач в выбранной предметной области.

2. Приложения онтологии по культуре. Интеграция структурированных музейных описаний

Для интеграции разнородных источников данных необходимо следующее:

1) общая онтология, включающая лингвистический ресурс – тезаурус по культуре;

2) способ отображения метаданных (или концептуальной схемы) источника на общую онтологию;

3) способ интерпретации и явного представления содержимого источника в терминах общей онтологии.

Интеграция данных не является конечной целью. Необходимость в интеграции возникает при решении других задач во многих областях информационных технологий: поиск информации, электронная коммерция и др. За подробным описанием задач, требующих интеграции разнородных данных, можно обратиться к [6].

После завершения концептуальной интеграции разнородных источников возможно решение новых задач, использующих интегрированный ресурс, например создание информационно-поисковой системы по культурному наследию.

Современные разработки в области семантической интеграции широко применяют дескриптивные логики [7, 8], но редко прибегают к использованию лингвистических технологий [9, 10]. Очевидно, что использование одних только лингвистических технологий неоправданно в случае, когда источники данных имеют некоторую структуру. Перспективность комбинирования двух этих подходов отмечается, например, в [11].

Далее представлен подход, основанный на использовании онтологии по культуре для интеграции разнородных описаний из музейных баз данных.

2.1. Постановка задачи интеграции разнородных источников. Далее под источником данных понимается информационный объект, состоящий из двух компонентов: содержимое источника (непосредственно сами данные) и метаданные (схема или модель источника). Такое представление соответствует стандарту MetaObject Facility[3], разработанному Object Management Group для описания произвольных моделей, который включает 4 уровня: уровень информации (1), уровень модели данных (2), уровень метамодели (3) и уровень метаметамодели (4). Допустим, что на уровнях 3 и 4 исходные источники данных однородны или могут быть представлены как однородные. Для определенности на уровне 3 подразумевается использование некоторой дескриптивной логики. Уровень 4 в этом случае соответствует языку описания синтаксиса и семантики конструкций соответствующей логики. Метаданные источника данных представлены как некоторая терминология S, то есть система понятий (концептов или классов) и их связей (ролей или свойств). Содержимое источника представляется как множество индивидуальных экземпляров (домен интерпретации Δ^I данной терминологии). Не ограничивая общности, будем считать, что все исходные источники имеют один и тот же домен интерпретации (он может быть представлен объединением доменов интерпретации исходных источников), но каждый источник имеет свою терминологию, или *схему*.

Поставим задачу построения межсхемных отображений.

Пусть дана исходная схема S и результирующая (глобальная) схема T. Через C и R обозначим, соответственно, атомарные концепт и роль в схеме S. Необходимо построить отображение $\varphi : S \to T$ такое, что

$$\forall C \in S.C^I \subseteq (\varphi(C))^I\,, \quad \forall R \in S.R^I \subseteq (\varphi(R))^I\,. \tag{1}$$

При этом в схеме T не существует концепта C^*, не эквивалентного $\varphi(C)$, и роли R^*, не эквивалентной $\varphi(R)$, для которых, соответственно, выполняется:

$$C^I \subseteq (C^*)^I \subseteq (\varphi(C))^I\,, \quad R^I \subseteq (R^*)^I \subseteq (\varphi(R))^I\,. \tag{2}$$

Здесь $I = (\Delta^I,\ f^I)$ – интерпретация, состоящая из домена интерпретации Δ^I и функции интерпретации f^I. Применение оператора $(\cdot)^I$ состоит в вычислении для данного аргумента значения функции f^I, которая каждому концепту C ставит в соответствие некоторое подмножество из домена интерпретации Δ^I, а каждой роли R – подмножество из $\Delta^I \times \Delta^I$. Условие (1) необходимо, чтобы индивиды, соответствующие концепту в исходной схеме, также принадлежали его образу в результирующей схеме, а условие (2) требует, чтобы построенное отображение было минимальным. Результат отображения атомарного концепта (роли) необязательно является атомарным концептом (ролью). В общем случае он представляется запросом над атомами результирующей схемы.

Таким образом, построение отображения φ состоит в выражении концептов и ролей схемы S через запросы над схемой T. Для выполнения условий (1) и (2) достаточно выражать концепты и роли из S с помощью отношения эквивалентности.

Для проверки условий (1) и (2) необходимо иметь некоторую интерпретацию I. В идеальном случае эксперт, хорошо понимающий смысл концептов и ролей в схемах, заменяет каждый атомарный элемент схемы S запросом над схемой T. При большом числе разнородных исходных источников этот процесс усложняется. Возникает необходимость моделировать интерпретацию концептов исходной схемы, используя автоматическую процедуру. Для этого предлагается сделать следующее:

1) Связать концепты глобальной схемы T с понятиями тезауруса данной предметной области. Этот шаг был выполнен при отображении тезауруса на онтологию CRM.

[3] www.omg.org/mof

2) Для каждого атомарного концепта X (для атомарной роли X – аналогично) из исходной схемы S построить список, содержащий те понятия тезауруса, которые встретились в текстовом содержимом экстенсионала данного атома. Полученный список определяет интерпретацию атомарного концепта в терминах информационно-поискового языка (ИПЯ) тезауруса.

3) Поскольку понятия тезауруса связаны также с концептами схемы T, использовать связи между понятиями тезауруса и глобальной схемой для автоматического выделения тех концептов схемы T, которые войдут в искомое представление атома X.

Такая процедура интерпретации имеет недостатки, поскольку опирается на содержимое источника, а оно может быть неточным, недостаточным или, наоборот, избыточным. С другой стороны, данная процедура интерпретации позволяет отбросить отображения, ошибочные или не имеющие смысла для данного исходного источника. Эта актуальная проблема редко рассматривается в системах интеграции данных [12]. Данный подход может быть расширен и применяться и в случае, когда экстенсионал некоторого исходного понятия имеет выражение в виде числовых последовательностей, например результатов измерения некоторой величины. Для этого достаточно либо снабдить каждое понятие из исходной схемы текстовой аннотацией, отражающей имя данной величины, либо расширить прикладную онтологию так, чтобы появилась возможность обработки числовых величин с некоторой размерностью.

Для применения описанного подхода необходимо привести все исходные источники данных к единому формализму описания. В качестве такого единого формализма, как указывалось выше, выбран формализм дескриптивных логик (или логик описания).

Опишем способ, с помощью которого структура БД может быть переведена в концептуальную схему, а содержимое таблиц – в интенсионал этой схемы.

Имена таблиц и столбцов БД трактуются как имена концептов (далее – классов) исходной схемы и формируют ее интенсионал, а содержимое таблиц и столбцов представляется как экземпляры соответствующих классов и является экстенсионалом схемы. Вхождение столбца в таблицу трактуется как наличие связи между классом, соответствующим таблице, и классом, соответствующим столбцу. Наличие связи между таблицами также трактуется как связь между соответствующими таблицам классами.

Предлагается моделировать интерпретацию класса концептуальной схемы БД на основе текстового содержимого, как это описано выше. Содержимое столбца, то есть набор текстовых значений, хранящихся в ячейках, задает текстовое представление соответствующего класса схемы.

Основное предположение состоит в том, что по совокупности текстовых представлений экземпляров класса можно однозначно определить интенсионал данного класса (иными словами определить его семантику) и использовать эту информацию для поиска близких классов в других схемах.

Далее описан алгоритм, основанный на предлагаемом подходе к интерпретации значения концептов в разнородных схемах и методике скрытого семантического анализа.

2.2. Алгоритм связывания элементов схем на основе их смыслового содержимого. Итак, исходные источники данных можно представить в виде некоторых концептуальных схем, и каждый класс такой схемы можно представить как текстовый документ (лексический образ экстенсионала класса). Задача поиска близких по смыслу классов в различных схемах сводится к поиску близких

текстовых документов. Для определения близости документов, представленных списком терминов, в области информационного поиска разработано множество подходов [13]. Далее предлагается использовать технику латентного (скрытого) семантического анализа LSA[4]. Каждый документ представляется списком терминов, который получается индексированием текста документа с помощью тезауруса. После подсчета числа вхождений того или иного понятия в документе строится вектор, соответствующий документу.

Матрица D, составленная из векторов-документов, как правило, очень разрежена и неудобна для оценки близости документов на основе скалярного произведения векторов. В методе LSA применяется сжимающее отображение, использующее сингулярное разложение матрицы D и понижающее размерность исходного пространства понятий. Важным свойством этого отображения является то, что после уменьшения размерности пространства векторы-документы, близкие в исходном пространстве, становятся еще ближе, а удаленные друг от друга векторы – еще дальше. Скалярное произведение векторов в пространстве уменьшенной размерности позволяет судить о скрытой семантической близости исходных документов, а следовательно, и о близости классов различных концептуальных схем.

Оценка значений координат вектора (например, его модуля) позволяет говорить о специфичности соответствующего документа. Относительно столбцов БД эта информация может использоваться для отделения столбцов описания, комментариев или примечаний (где все термины тезауруса могут появляться одинаково часто) от специфических столбцов, характерных для таблиц-справочников, в которых используется достаточно узкий набор терминов (например, Материал или Техника).

Более подробно способ адаптации техники LSA и соответствующий алгоритм описаны в [14]. Далее рассматриваются некоторые результаты работы алгоритма.

2.3. Результаты экспериментов. Для проведения экспериментов по оценке близости классов из различных схем были отобраны несколько десятков записей из трех музейных баз данных: Этнографического музея Казанского университета (ЭМКУ), Всероссийского реестра музеев (ВРМ) и Национального музея Республики Татарстан (НМРТ). Концептуальные схемы были сгенерированы из таблиц баз данных автоматически, как было указано выше. Общее число документов, полученных на основе классов-столбцов исходных схем, составило 52. К числу документов добавляются также ДБ тезауруса. Эти документы представляются списками терминов, входящих в соответствующий блок. Добавление ДБ к общему числу документов сделано для улучшения качества работы метода LSA, поскольку новые документы группируют термины по смыслу, что существенно влияет на семантическую близость других документов, в которых эти термины также встречаются. Число документов, добавленных на основе дескрипторных блоков, – 236. Общее число терминов индексирования, полученных из тезауруса, – 6500. Таким образом, размерность матрицы составила 288×6500. Сравнение близости документов проводилось в трех случаях: после уменьшения размерности пространства терминов до 10, 100 и 200. Результаты существенно не зависели от новой размерности пространства терминов (10, 100 и 200).

Получены как положительные результаты, позволяющие говорить о применимости метода на практике, так и отрицательные (см. табл. 1).

Отрицательные результаты объясняются неточностью индексирования документов и общим списком терминов для таких полей, как Описание. Несмотря на

[4]http://lsa.colorado.edu/

Табл. 1

Результаты применения метода LSA (отрицательные результаты отмечены звездочкой)

Класс (имя столбца)	Семантика столбца	Список возможных отображений соответствующего класса
material	Материал	Вещь_E70* Концептуальный Объект_E28 Материал_E57 Продукт Деятельности Человека_E24*
OPIS	Описание, комментарий	Вещь_E70 Продукт Деятельности Человека_E24 Деятельность_E7 Юридическое Лицо_E40 Документ_E31 Концептуальный Объект_E28 Материал_E57 Группа_E74
ethnogr_prinadl	Этнографическая принадлежность	Продукт Деятельности Человека_E24* Группа_E74
postupl_istchn	Источник поступления	Юридическое Лицо_E40
postupl_sposob	Способ поступления	Деятельность_E7
technique	Техника	Материал_E57* Продукт Деятельности Человека_E24* Деятельность_E7 Концептуальный Объект_E28

наличие отрицательных результатов, в каждой строке есть хотя бы один подходящий по смыслу класс результирующей схемы, и общее число вариантов невелико.

Проведенные эксперименты показали, что лексический состав разработанной онтологии по культуре позволяет связывать близкие по смыслу классы из различных исходных схем. Особенность подхода состоит в том, что для поиска близких по смыслу классов можно использовать небольшое число записей из исходных источников. В проводимых экспериментах использовалось менее 200 записей (строк) из всех трех БД. Векторы-документы могут быть построены также на текстовых примечаниях к столбцам таблицы.

2.4. Формат определения межсхемных отображений. На основе формата определения отображения, предложенного в [15], была разработана онтология для определения отображения. Эта онтология позволяет более гибко (в виде отображения между запросами над исходной и результирующей схемами) задавать отображения между двумя различными схемами и хранить их как независимый набор утверждений.

Описания отображений могут быть повторно использованы при подключении нового источника данных с похожей структурой. Онтология отображения состоит из трех групп классов: Домены, Диапазоны и Пути. Домены могут использоваться в качестве начала Путей, которые, в свою очередь, заканчиваются в Диапазонах. В каждой из групп есть классы, соответствующие и сущностям исходной схемы, и сущностям результирующей схемы (например, Исходный Домен или Результирующий Путь). Основным отношением в онтологии отображения является свойство mapsTo, которое каждому экземпляру исходной сущности ставит в со-

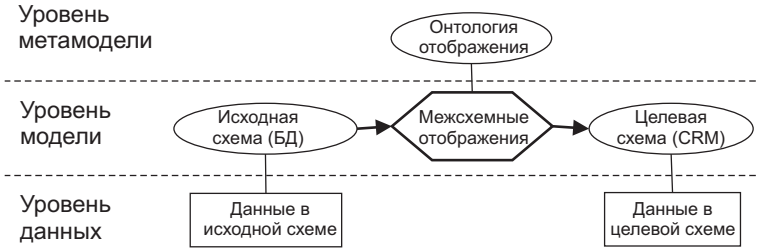

Рис. 2. Логическое представление процесса отображения

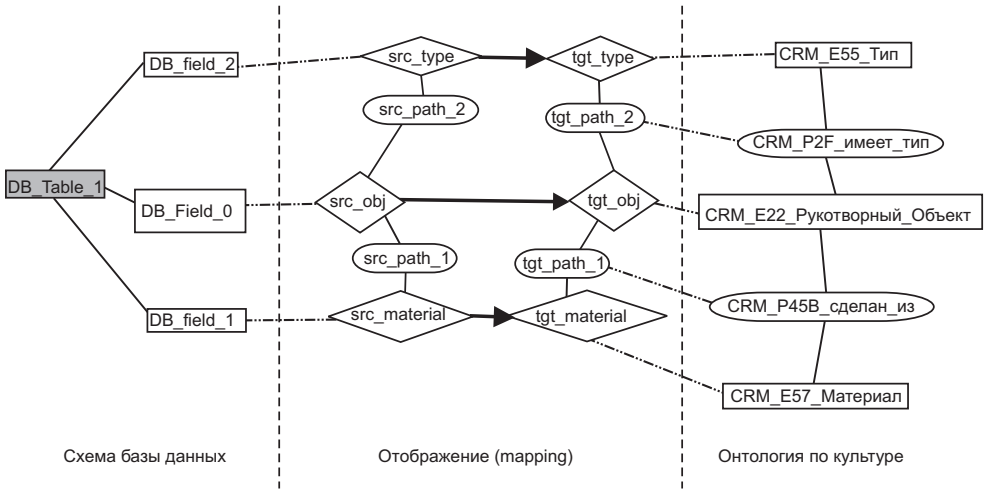

Рис. 3. Визуальное представление отображения. Темными линиями в центре выделены связи «mapsTo» между исходной схемой (слева) и результирующей (справа)

ответствие экземпляр некоторой результирующей сущности. Кроме того, вводится дополнительное свойство indexedWith. Это свойство позволяет на этапе построения отображения задать для каждого класса в результирующей схеме некоторый набор лексических единиц тезауруса, которыми при выполнении отображения будут индексированы текстовые представления новых экземпляров. С логической точки зрения онтология отображения находится на уровне метамодели по отношению исходной и результирующей схемам (рис. 2, 3).

После «выполнения» отображения экземпляры классов извлекаются из исходного источника и погружаются в контекст результирующей схемы. Новые экземпляры связываются друг с другом в соответствии с определенным отображением. Кроме того, некоторые новые экземпляры связываются с дескрипторами тезауруса.

3. Приложения онтологии по культуре. Поиск описаний музейных предметов

Основная цель работы, как отмечалось выше, состоит в предоставлении доступа к информации, извлеченной из разнородных источников данных. Одним из главных механизмов доступа является поиск информации по запросу. Стоит отметить,

что база данных[5] создаваемой системы имеет несколько особенностей, существенно влияющих на механизмы поиска. Опишем эти особенности.

3.1. Особенности интегрированного хранилища. Первая особенность (структурное отличие). Модель данных, используемая для хранения экземпляров онтологии, отличается от реляционной. Данные хранятся в базе знаний системы в виде набора связанных утверждений, троек вида (s, p, o), а не в виде кортежей некоторого отношения. Такая модель ближе к сетевой модели данных, чем к реляционной модели. Данное отличие влияет на форму запроса к базе знаний, который по своей структуре ближе к логической формуле, чем к SQL-предложению. Как следствие, отличие оказывает влияние на то, какой результат возвращается после выполнения запроса. В общем случае результатом выполнения запроса будет некоторое множество утверждений, извлеченных из базы знаний.

Вторая особенность связана с тем, что в роли субъекта или объекта утверждения могут быть понятия тезауруса, которые рассматриваются как семантические метки для значений тех или иных свойств экземпляров онтологии. Это отличие влечет возможность описывать образ желаемого результата, используя лексику естественного языка (ЕЯ), и извлекать из базы знаний утверждения, помеченные соответствующими понятиями. Наряду с возможностью описывать образ запроса в виде логической формулы над атомами онтологии появляется возможность задавать и обрабатывать запросы, заданные в виде предложений естественного языка. Первый способ задания запроса приводит к однозначной его интерпретации, но требует от пользователя хорошего знания всей онтологии, правил построения логических формул и имеет другие недостатки, описанные, например, в [16, p. 31–32]. Второй способ формулировки запроса более прост и интуитивно понятен. Однако при обработке такого запроса возможно появление множественных интерпретаций. Эта проблема широко известна в области информационного поиска. Результаты, извлеченные по ЕЯ-запросу, могут не полностью отвечать поисковой потребности пользователя, но часто это – единственный возможный способ сформулировать запрос. Близость между запросом и возвращенными таким образом данными необходимо оценивать, вычислять релевантность результата. Для сравнения двух форм задания запроса рассмотрим следующий пример.

1) Мебель, сделанная из дерева

2) SELECT ?x FROM <source_ontology.owl> WHERE ?x a ?physicalObjectClass .
?x <crm:P2F_имеет_тип> <culthes:Мебель> .
?x <P45F_сделан_из> <culthes:Дерево> ?physicalObjectClass rdfs:subClassOf <crm:E18_ Материальный_Объект>

Два данные запроса семантически эквивалентны, но второй более громоздкий и жестко привязан к понятиям (именам классов и свойств) онтологии верхнего уровня. Во второй запрос также включены понятия из тезауруса, такие как Мебель, Дерево, а остальные части запроса необходимы, чтобы увязать между собой эти понятия тезауруса и избежать неоднозначной интерпретации. Первый запрос может быть адресован любой информационной системе, поддерживающей ЕЯ-интерфейс, в том числе и к поисковой системе интернета, что позволяет сравнивать выдаваемые результаты или получать дополнительную информацию из сети.

Существует возможность задавать запрос на некотором собственном диалекте запросов. Этот случай соответствует некоторому промежуточному положению

[5]Далее будем именовать результирующее хранилище базой знаний, что соответствует определению, принятому в литературе по дескриптивной логике. Множество всех экземпляров, для которых определены классы и свойства онтологии, или так называемый Terminology Box, называется базой знаний – Assertion Box, или knowledge base.

между двумя крайними вариантами, описанными выше. Недостатком этого, как может показаться, компромиссного решения является практически полная несовместимость с имеющимися и появляющимися средствами обработки запросов.

Третья особенность состоит в том, что онтология, определяющая структуру базы знаний, является формальной логической теорией, состоящей из набора унарных и бинарных предикатов, заданных на множестве экземпляров, и набора аксиом, позволяющих выполнять логический вывод, например, с целью получения новых утверждений. Эта особенность является наиболее важной, поскольку дает возможность включать в результаты запроса новые факты, изначально отсутствовавшие в исходных источниках данных. Основные формы логического вывода (вывод по транзитивности, вывод по иерархии понятий и иерархии свойств) поддерживаются стандартными средствами[6]. Существует также возможность определить предметно-ориентированную систему правил вывода.

Перечисленные особенности выделяют разрабатываемую систему как из класса документальных информационных систем (систем извлечения информации из текста, поиска документов и т. п.), так и из класса информационных систем для хранения и обработки фактографической информации на основе реляционных баз данных.

3.2. Обработка запроса. Как отмечалось ранее, результатом выполнения запроса (в какой бы форме он ни задавался) будет набор утверждений или троек вида (subject, predicate, object), где под subject понимается некоторый ресурс, обладающий свойством predicate, значение которого равно object.

Поскольку обработка запроса, написанного в виде логической формулы, тривиальна, перейдем к описанию методов обработки ЕЯ-запросов в системе. Сначала сделаем два допущения.

1) В результате выполнения запроса из базы знаний должны извлекаться утверждения, как-либо относящиеся к понятиям из запроса.

2) Агент (человек или программа), формулирующий запрос, ожидает извлечения некоторой совокупности взаимосвязанных утверждений, близкой по смыслу к задаваемому запросу в целом.

Перед выполнением ЕЯ-запроса его необходимо интерпретировать в терминах общей онтологии. Это осуществляется извлечением из текста запроса понятий тезауруса. Запрос фактически заменяется списком понятий (которые в онтологии по культуре представляются как экземпляры). Далее, в соответствии с первым допущением, из базы знаний извлекаются все утверждения, в которых данное понятие запроса встречается либо в роли субъекта (subject), либо в роли объекта (object) некоторого свойства (predicate). Ясно, что такой подход никак не учитывает второго допущения и может привести к извлечению совокупностей утверждений, каждая из которых порождена определенным понятием запроса, но при этом нет никакой связи между утверждениями из разных совокупностей, нет цельности в интерпретации запроса. Для того чтобы учесть второе допущение, необходимо, во-первых, определить, как понятия запроса комбинируются между собой, образуя общий смысл запроса, и, во-вторых, достраивать результат до некоторой цельной совокупности взаимосвязанных утверждений. Имеет смысл (при отсутствии другой информации) представлять запрос конъюнкцией понятий, предполагая, что для каждого из понятий запроса в извлекаемой совокупности (результате) существует хотя бы одно утверждение, соотносящееся с этим понятием. В другом крайнем случае запрос можно представить дизъюнкцией понятий. В общем случае запрос представляется дизъюнкцией конъюнкций.

[6]http://pellet.owldl.com

3.3. Алгоритм обработки ЕЯ-запроса. Входные данные: текст запроса (строка символов), база знаний G (множество утверждений).

Выходные данные – подмножество утверждений, извлеченное из базы знаний.

Шаг 1. Предобработка текста запроса. Текст запроса индексируется понятиями тезауруса. Понятия, встретившиеся в тексте запроса, вносятся в список $\mathbf{t} = (t_1, \ldots, t_n)$. Если после индексирования список $\mathbf{t}$ оказался пустым, то вернуть пустой результат и выйти из алгоритма, иначе положить множество M результатов запроса равным пустому множеству и перейти к Шагу 2.

Шаг 2. Расширение запроса. Для каждого из понятий t_i из списка $\mathbf{t}$ построить множество T_i, в которое включить понятие t_i и все понятия, стоящие ниже t_i в иерархии понятий тезауруса по отношению ВР–НВ. Последовательно для каждого из множеств T_i положить множество извлеченных утверждений m_i равным пустому множеству и выполнить Шаг 2.1. После выполнения Шага 2.1 для всех множеств T_i перейти к Шагу 3.

Шаг 2.1. Если множество T_i пусто, то завершить обработку текущего множества T_i и выйти из Шага 2.1, иначе взять любой элемент t_j из множества T_i и выполнить для него следующее:

1) удалить элемент t_j из множества T_i;

2) положить список ресурсов *resources* состоящим из одного элемента t_j, список ресурсов для следующей итерации *newResources* равным пустому списку;

3) Если выполняется условие останова C, то перейти к Шагу 2.1, иначе:

а) для каждого утверждения вида (s, p, o) из базы знаний G такого, что s находится в списке *resources*, добавить o в список *newResources*, а утверждение (s, p, o) добавить в множество m_i;

б) для каждого утверждения вида (s, p, o) из базы знаний G такого, что o находится в списке *resources*, добавить s в список *newResources*, а утверждение (s, p, o) добавить в множество m_i;

в) положить список *resources* равным списку *newResources*, список *newResources* равным пустому списку и перейти к пункту 3).

Шаг 3. Вычислить результат M запроса как некоторую формулу F над множествами утверждений m_i. Выдать результат и выйти из алгоритма.

На этом описание алгоритма завершено.

Как видно, алгоритм имеет два параметра: условие останова C и формула F. Формула F полностью определяется дизъюнктивной формой, связывающей понятия внутри запроса: каждому понятию соответствует множество m_i, каждой логической связке соответствует некоторая операция над множествами. В самом общем случае $F : M_1 \times \cdots \times M_n \to M$, где $M_i \subseteq G$, $i = 1, \ldots, n$, и $M \subseteq G$. Условие останова C срабатывает, когда список ресурсов *resources* становится пустым. Кроме того, условие может ограничивать количество итераций внутри пункта 3), что позволяет задавать максимальную длину цепочек утверждений от понятий запроса до связанных с ними экземпляров онтологии. Для того чтобы исключить зацикливание (многократный проход по одним и тем же утверждениям), при добавлении нового ресурса в список *newResources* необходимо проверять, был ли данный ресурс хоть раз добавлен в список *newResources* на одной из предыдущих итераций внутри пункта 3). Внесение соответствующего изменения в описание алгоритма не представляет особой сложности.

В заключительной части статьи описывается архитектура прототипа программной системы для интеграции музейных описаний и поддержки семантического поиска в объединенной базе знаний. В основу разработки прототипа легли изложенные выше базовые аспекты технологии создания подобных систем.

4. Программная реализация технологии

4.1. Внешние функции системы. Основными функциями системы являются:

1) итеграция разнородных информационных источников в области музейной документации;

2) поиск в интегрированной базе знаний по запросу, заданному
   – на естественном языке;
   – формулой на языке SPARQL, составленной из понятий онтологии по культуре.

4.2. Структура системы. Структурно система состоит из ядра, включающего в себя онтологию верхнего уровня, расширенную тезаурусом, и программных модулей, поддерживающих выполнение внешних функций. Для представления онтологии в системе используется язык OWL. Функциональные модули реализуются на языке Java.

Ядро системы. В данном случае онтология CRM была представлена на языке OWL DL. Модель расширена путем подключения к ней тезауруса CULTHES.

Функциональные модули системы. Исходными данными для системы является набор XML-файлов, соответствующих некоторой единой XML-схеме и описывающих источник. По существу это могут быть таблицы БД, сохраненные в XML-формате, либо данные, извлеченные из HTML-страниц. Имена XML-элементов, атрибутов и их взаимосвязи определяют метаданные источника. Текстовое содержимое элементов и атрибутов соответствует уровню данных.

Модуль подключения источника данных реализует следующие функции.

• Обработка исходного набора XML-файлов и извлечение из них концептуальной схемы источника. Это реализуется с помощью XSL-трансформации, переводящей исходный набор XML-файлов в один OWL-файл, содержащий онтологию (схему или интенсионал источника) и экземпляры онтологии (экстенсионал). Каждый экземпляр онтологии имеет связь (rdfs:label) со своим текстовым представлением.

• Поддержка построения отображения (φ) концептуальной схемы источника на онтологию CRM. Эта функция по существу реализует метод связывания классов разнородных схем на основе техники LSA. После ее выполнения классы исходной концептуальной схемы автоматически связываются с одним или более классами результирующей схемы – онтологии CRM. Эти связи фиксируются как утверждения в онтологии отображения и далее могут редактироваться.

• Выполнение построенного отображения: использование исходных данных и отображения φ для наполнения базы знаний. Данная функция реализована с помощью генерации SPARQL-запроса к исходной схеме и использовании информации об отображении для создания множества связанных экземпляров в результирующей онтологии. На этом этапе также происходит связывание текстовых значений с понятиями тезауруса.

Модуль лингвистической обработки является своеобразным сервером, поддерживающим при необходимости выполнение функций других модулей. Основная его функция связана с индексированием некоторого заданного текстового фрагмента при помощи заданного набора понятий. Модуль также выполняет:

– очистку исходных данных;

– этапы автоматизированной лингвистической обработки текстовых данных – морфологический и терминологический анализы.

Модуль поиска обрабатывает внешние запросы, заданные либо формулой запроса на языке SPARQL, либо предложением на естественном языке, извлекает

из базы знаний соответствующие запросу утверждения и представляет результат пользователю.

4.3. Реализация прототипа системы. На настоящий момент сформировано ядро системы: на онтологию CRM отображены 8 основных фасетов русскоязычного тезауруса CULTHES. Общее число дескрипторов тезауруса, включенных в онтологию, составляет 6.5 тыс.

Реализован модуль подключения источника данных. Экспериментальный прототип тестируется на трех разнородных базах данных российских музеев: Национального музея Республики Татарстан, Музея Истории и Этнографического музея Казанского университета, а также на данных Всероссийского реестра музеев, полученных с портала www.museum.ru. Общее число предметов составило около 10 тыс. единиц хранения, а общее число утверждений в базе знаний – порядка 10^5.

Модуль лингвистической обработки практически полностью построен на свободно распространяемых компонентах, разработанных различными организациями и сообществами. Для обработки текстовых документов используются основные функции пакета GATE 4. В качестве морфологического анализатора русского языка использован модуль RuPOSTagger с сайта aot.ru. Для индексирования текстов понятиями тезауруса используется пакет Apolda 1.3. Однако использованных модулей оказалось недостаточно. В частности потребовалось создать надстройку для корректного извлечения двухсловных и трехсловных терминов на основе шаблона, задающего синтаксическое отношение между словами термина. Пример шаблона: «существительное + существительное (в родительном падеже)».

Модуль поиска реализует описанный выше алгоритм извлечения связанных совокупностей утверждений. По заданному на естественном языке запросу и базе знаний строится новая база знаний, содержащая соответствующие утверждения. Модуль также реализует функцию представления результатов поиска с помощью XSL-трансформации, применяемой к результирующему OWL-документу. Результаты поиска могут быть представлены в виде списка веб-страниц с описанием извлеченных по запросу объектов.

Узким местом алгоритма поиска в вычислительном аспекте оказалось то, что время его работы зависит от числа понятий, извлеченных из запроса. Поэтому при реализации важно учитывать, что алгоритм допускает очевидное распараллеливание по числу терминов запроса. Шаг 2 может выполняться на n компьютерах. После вычисления всех множеств T_i Шаг 2.1 также может независимо выполняться на n компьютерах. На шаге 3 выполняется сборка решения и возврат результата.

Заключение

Проведенные эксперименты показали, что предлагаемая технология может применяться на практике для объединения исходных источников разного типа (от слабоструктурированных веб-страниц, являющихся интерфейсом к внутренней базе данных сайта, до наборов связанных таблиц некоторой реляционной базы данных), и учитывает основные виды неоднородности. Отличительной чертой предлагаемого подхода является совместное использование формальной онтологии верхнего уровня и тезаурусов. Один из общих результатов проделанной работы состоит в том, что структурно схемы (метаданные) исходных источников данных похожи (часто это плоские таблицы), а сами данные в основном представляют текстовые описания. Конечно, это утверждение верно для слабо формализованных областей, какой, например, является область музейной документации. Поэтому целесообразно, с одной стороны, использовать онтологию верхнего уровня для явного представления

семантики связей, уложенных в плоские таблицы, а с другой – лингвистические ресурсы для представления смысла текстового содержимого.

Важно отметить еще одно перспективное приложение разработанной нами технологии. В случае, когда БД генерируется из набора веб-страниц, после построения интегрированного хранилища появляется возможность снабдить исходные страницы семантическими аннотациями (метаданными), которые могут быть использованы другими поисковыми агентами сети, разделяющими общую онтологию.

Близкие работы в области интеграции данных связаны с развитием теоретической и технологической базы интеграции на основе онтологий (проекты MAFRA [17]) и созданием веб-порталов для доступа к культурному наследию (финский проект Finnish Museums on the Semantic Web и голландский – MultimediaN).

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проекты № 06-07-89219-а и № 07-07-12039-офи).

Summary

V.V. Ivanov. Ontology-based approach to creation of an information system on cultural heritage.

The article is devoted to a set of technological solutions developed in order to create a modern information system in culture heritage domain. A crucial aspect of the technology is in using ontologies from two points of view: formal and linguistic. The technology allows extension to similar subject domains. A process of development of an applied ontology in cultural heritage domain is discussed with respect to two problems: integration of heterogeneous data sources and intelligent information retrieval. Two applications of the developed ontology are described. In conclusion a structure of a demonstration program tool is provided.

Литература

1. *Doerr M., Hunter J., Lagoze C.* Towards a core ontology for information integration [Электронный ресурс] // J. Digital Information. – 2003. – V. 4, No 1. – Режим доступа: www.cs.cornell.edu/lagoze/papers/Core_Ontology.pdf, свободный.
2. Art and Architecture Thesaurus: in 5 v. – N. Y.: Oxford Univ. Press, 1994. – 413 p.
3. *Добров Б.В., Лукашевич Н.В., Соловьев В.Д.* Тезаурус по архитектуре и искусству как средство формализации описаний музейных предметов [Электронный ресурс] // Web journal of Formal, Computational and Cognitive Linguistics. – 2006. – Режим доступа: http://fccl.ksu.ru/issue_spec/docs/aat_index.doc, свободный.
4. *Кронгауз М.А.* Семантика. – М.: Рос. гос. гуманит. ун-т, 2001. – 399 с.
5. *Baader F., Calvanese D., McGuinness D., Nardi D., Patel-Schneider P.* The Description Logic Handbook : Theory, Implementation and Applications. – Cambridge: Cambridge Univ. Press, 2003. – 555 p.
6. *Rahm E., Bernstein P.A.* A Survey of Approaches to Automatic Schema Matching // The International J. on Very Large Data Bases – 2001. – V. 10, No 4. – P. 334–350.
7. *Calvanese D., De Giacomo G., Lenzerini M.* Description logics for information integration // Kakas A.C., Sadri F. (Eds.). Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II. Lecture Notes in Computer Science. – Springer, 2002. – V. 2408 – P. 41–60.
8. *Preece A.D., Hui K.-J., Gray W.A., Marti P., Bench-Capon T.J.M., Jones D.M., Cui Z.* The KRAFT architecture for knowledge fusion and transformation // Knowledge Based Systems. – 2000. – V. 13, No 2–3. – P. 113–120.

9. *Wache H.* Towards rule-based context transformation in mediators // Conrad S., Hasselbring W., Saake G. (eds.) International Workshop on Engineering Federated Information Systems (EFIS 99). – Germany, Infix-Verlag, 1999. – P. 107–122.

10. *Mena E., Kashyap V., Sheth A., Illarramendi A.* Observer: An approach for query processing in global information systems based on interoperability between pre-existing ontologies // Proc. 1st IFCIS Int. Conf. on Cooperative Information Systems. – 1996. – P. 14–25.

11. *Poesio M.* Domain modelling and NLP: Formal ontologies, Lexica, Or a bit of both // Appl. Ontology. – 2005. – V. 1, No 1. – P. 27–33.

12. *Wache H., Vögele T., Visser U. et al.* Ontology-based integration of information – a survey of existing approaches // Stumme G., Maedche A., Staab S. (Eds.) Proceedings of the IJCAI'2001 Workshop on Ontologies and Information Sharing, Seattle, USA, 4–5 August 2001. – CEUR-WS, 2001. – V. 47. – P. 108–117.

13. *Van Rijsbergen C. J.* Information Retrieval. – London: Butterworths, 1979. – 208 p.

14. *Соловьев В.Д., Иванов В.В.* Создание и валидация онтологии в области культуры на базе онтологии верхнего уровня и тезауруса // Моделирование процессов: Тр. Казан. науч. семинара «Методы моделирования». – Казань: Изд-во КГТУ, 2007. – Вып. 3. – С. 135–153.

15. *Kondylakys H., Doerr M., Plexousakis D.* Mapping language for information integration [Электронный ресурс]. – Technical Report 385, ICS-FORTH. – 2006. – Режим доступа: http://www.ics.forth.gr/isl/publications/paperlink/ Mapping_TR385_December06.pdf, свободный.

16. *Jackson P., Mouliner I.* Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization. – John Benjamins Publishing Company, 2002. – 224 p.

17. *Maedche A., Motik B., Silva N., Volz R.* MAFRA – a mapping framework for distributed ontologies // Proc. of the 13th Internat. Conf. on Knowledge Engineering and Knowledge Management (EKAW 2002). – Siguenza, Spain, 2002. – P. 235–250.

Поступила в редакцию
30.07.07

Иванов Владимир Владимирович – аспирант кафедры теоретической кибернетики Казанского государственного университета.

E-mail: *vivanov@ksu.ru*