УДК 004.62

Чистякова И.С.
Институт программных систем НАН Украины

РОЛЬ ОНТОЛОГИЙ В ИНТЕГРАЦИИ ДАННЫХ В СЕМАНТИЧЕСКОМ ВЕБЕ

Работа посвящена проблеме интеграции данных в Семантическом Вебе. Рассматривается процесс интеграции, основные его составляющие, а именно: выработка схем интеграции, выработка отображений между моделями, выработка способов манипулирования.

Робота присвячена проблемі інтеграції даних в Семантичному Вебі. Розглядається процес інтеграції, головні його складові, а також вироблення схем інтеграції, вироблення відображень між моделями, вироблення засобів маніпулювання.

This paper is devoted to the problem of data integration in the Semantic Web. The process of integration, its main components, namely, construction of integration schemes, the development of mappings between models, the development of ways of manipulation are considered.

***Ключевые слова*:** *онтология, интеграция данных, семантическая интеграция, модель данных.*

Онтологии в Семантическом Вебе

Для реализации любой современной семантической технологии необходим соответствующий информационный контент, то есть данные, представленные в требуемом семантическом формате. В настоящее время большое количество онтологий можно найти в интернете, в том числе и в свободном доступе. Существует множество информационных ресурсов, предоставляющих онтологии для свободного использования.

В научном мире структурировать уже готовые онтологии пытался не один исследователь [10]. Следует отметить, что согласно анализу множества работ, не существует единой, общепринятой классификации, поскольку онтологии сильно различаются по ряду параметров, и исследователи выделяют различные основания для их классификации. При этом, между собой они могут быть взаимозаменяющими и взаимодополняющими. Рассмотрим некоторые примеры.

Э.Хови[6] выстраивает подробную классификацию различных характеристик онтологий. Можно упомянуть о том, что существует разбиение онтологий по количеству и качеству понятий, включаемых в них. Таким образом, выделить три вида:

- онтологии верхнего уровня;
- онтологии среднего уровня;
- онтологии нижнего уровня.

Онтологии верхнего уровня обычно насчитывают примерно 100-500 концептов. В них включены наиболее абстрактные категории, обладающие свойством универсальности. Они составляют некоторый базовый набор концептов для характеристики окружающего мира, и строятся обычно теоретиками, философами. Зачастую они полностью абстрактны. Составление аксиом в данном типе онтологий с высоким уровнем обобщения достаточно сложно и требует некоторого воображения. Преимуществом таких онтологий является возможность их использования во многих областях и даже во многих языках [6, 10].

Онтологии среднего уровня. Здесь элементов обычно больше (500 – 100000 концептов). Они представляют мир в целом, и в общем случае это неаксиоматизированная область. Сложность заключается в необходимости вывода слишком большого количества аксиом, где выходом является использование методов автоматизированного вывода аксиом из уже существующих онтологий. Построением таких онтологий чаще всего занимаются когнитологи и лингвисты [6, 10].

Онтологии нижнего уровня или так называемые онтологии предметной области наиболее обширные, обычно они насчитывают около 200 – 2 000 концептов. Они описывают конкретные предметные области с их спецификой. При этом круг решаемых задач и

вопросов, на которые онтология отвечает, ограничен выбранной областью. Для нее возможно построение большого количества аксиом и правил. В большинстве случаев строится экспертами области знания или при их содействии. В связи с большой спецификой каждой отдельной предметной онтологии ее повторное использование зачастую возможно только в рамках предметной области [6, 10].

С точки зрения предмета концептуализации исследователи выделяют:

- прикладные онтологии;
- онтологии области знания;
- общие (родовые) онтологии;
- репрезентационные онтологии (речь идет об онтологиях метауровня, включающих в себя репрезентационные первоэлементы).

Онтологии могут быть также разделены на *одноязычные* и *многоязычные*. Уже существует ряд онтологий, ориентированных на представление знаний на нескольких языках, например, EuroWordNet, MikroKosmos и некоторые другие. Сложность создания таких онтологий обычно заключается в том, что возможно наличие различий в понятийных системах разных языков [10].

С.А.Коваль [8] предлагает различать *безэкземплярные* и *экземплярные* онтологии. Как понятно из названия данного типа, безэкземплярные онтологии отличаются отсутствием конкретных экземпляров. На нижних уровнях иерархии таких онтологий находятся понятия. Эта особенность онтологии накладывает некоторый отпечаток и на вводимые в данной онтологии отношения.

Таким образом, существует множество вариантов классификации онтологий, но они не всегда являются четкими и последовательными. В последнее время все более широкое распространение получают так называемые адаптивные онтологии.

Адаптивная онтология – такая онтология, которая, кроме своих основных функций, наполняет семантической информацией пользовательские интерфейсы.

С точки зрения [11] смысловой нагрузке в этом случае подвергается пользовательская информация ресурса, например: названия атрибутов и подсказки, навигация и просмотр структур и деревьев, структуры меню, автоматическое завершение при вводе данных, контекстные выпадающие списки выбора, проверка правописания и т.д. Любой значимый эксперт в предметной области или в области управления знаниями может способствовать развитию и улучшению этих онтологий. Таким образом, мы определяем положительную тенденцию – смещение акцента от ИТ к самим знаниям.

На сегодняшний день в открытом доступе существует множество онтологий. В качестве примера можно привести следующие.

1. **Винная онтология (Wine)** [12]. Является примером онтологии, сопровождающей официальные W3C OWL стандарты. Для презентации особенностей языка OWL, она демонстрирует на собственном примере их употребление в домене вина и еды. Благодаря исчерпывающему и сбалансированному использованию различных OWL выражений, винные онтологии квалифицируются как указатель шкалы поддержки OWL. Кроме того, при использованни ее в качестве прототпа для онтологий большего размера (клонирование ресурсов), можно ожидать, что будет приведено значимое понимание свойств расширения алгоритма рассуждений.

2. **Открытые биомедицинские онтологии (OBO)** [12]. Являются коллекцией хорошо структурированных словарей для открытого использования сквозь различные биологические и медицинские домены. Оригинальный формат OBO является представлением, основанным на прямых ациклических графах, которые могут быть напрямую преобразованы в OWL. В настоящее время проект OBO охватывает более чем 60 отдльных онтологий со своими индивидуальными размерами, варьирующимися от тысяч до миллионов кортежей.

3. **EClassOWL** [12]. Является OWL-Lite представлением eClass – продуктов и сервисов стандарта категоризации. EClass включает продукты и понятия услуги, свойства продуктов, значения перечисляемых типов данных, иерархия понятий продукта, отражающие перспективы покупок организации, рекомендации, какие свойства должны быть использованы для соответствующих типов продуктов и рекомендации, какие значения разрешены для различных свойств. Онтология eClassOWL, была построена для охватывания как можно большей первоначальной семантики, в то же время хорошо уживаясь внутри ограничений OWL-DL.

Это далеко не полный список онтологий, которые применяются в современных информационно-поисковых системах. Существуют полноценные информационные

ресурсы, на которых представлено множество реальных онтологий, которые можно использовать.

В настоящее время количество актуальных онтологий возросло до таких размеров, что возникает новая проблема – создание специальных реестров, которые должны быть машиночитаемыми и машинообрабатываемыми. В процессе наших исследований мы обнаружили, что в данный момент прилагаются большие усилия для создания не машиночитаемых ресурсов, для ознакомления с которыми можно обратиться по этим адресам [15, 16, 17, 18, 19].

Для создания, изменения, управления структурой и содержимым онтологии существуют специальные редакторы. Полноценный их обзор можно найти в следующей работе [9]. Здесь мы лишь приведем несколько примеров существующих редакторов.

1. **Protégé** [9] – локальная, свободно распространяемая Java-программа, разработанная группой медицинской информатики Стенфордского университета (первая версия – 1987). Программа предназначена для построения (создания, редактирования и просмотра) онтологий прикладной области. Её первоначальная цель – помочь разработчикам программного обеспечения в создании и поддержке явных моделей предметной области и включение этих моделей непосредственно в программный код. Protégé включает редактор онтологий, позволяющий проектировать онтологии разворачивая иерархическую структуру абстрактных или конкретных классов и слотов. Структура онтологии сделана аналогично иерархической структуре каталога. На основе сформированной онтологии, Protégé может генерировать формы получения знаний для введения экземпляров классов и подклассов. Инструмент имеет графический интерфейс, удобный для использования неопытными пользователями, снабжен справками и примерами.

2. **OilEd** [9] – автономный графический редактор онтологий, разработан в Манчестерском университете в рамках европейского IST проекта On-To-Knowledge. Инструмент основан на языке OIL (сейчас адаптирован для DAML+OIL, в перспективе – OWL), который сочетает в себе фреймовую структуру и выразительность дескриптивной логики (Description Logics) с сервисами рассуждения. Это позволило обеспечить понятный и интуитивный стиль интерфейса пользователя и преимущества поддержки рассуждения (обнаружение логически противоречивых классов и скрытых отношений подкласса).

3. **WebOnto** [9] разработан для Tadzebao – инструмента исследования онтологий и предназначен для поддержки совместного просмотра, создания и редактирования онтологий. Его цели – простота использования, предоставление средств масштабирования для построения больших онтологий. Для моделирования онтологий WebOnto использует язык OCML (Operational Conceptual Modeling Language). В WebOnto пользователь может создавать структуры, включая классы с множественным наследованием, что можно выполнять графически. Все слоты наследуются корректно. Инструмент проверяет вновь вводимые данные контролем целостности кода OCML. Инструмент имеет ряд полезных особенностей: сохранение структурных диаграмм, раздельный просмотр отношений, классов, правил и т.д. Другие возможности включают совместную работу нескольких пользователей над онтологией, использование диаграмм, функций передачи и приёма и др.

Определение проблемы интеграции

Интеграция данных является одним из наиболее востребованных направлений в современной информационной индустрии. За все годы существования интернет-пространства в нем скопилось большое количество информации, объем которой с каждым днем возрастает в геометрической прогрессии, а релевантность – в арифметической. Это порождает множество проблем связанных с использованием и хранением данных информационного пространства. Огромные объемы разнородных данных в гетерогенных источниках представляют информацию различными способами и имеют разнообразное функциональное назначение. Интеграция и совместное использование информации из множества таких источников данных является сложной задачей, остающейся неизменно актуальной на протяжении последних десятилетий.

Можно выделить несколько порождающих причин гетерогенности:

1) Различные модели данных. Согласно разным сведениям от 75% до 90% (в зависимости статистического источника) информации хранится в РБД. Однако, на

оставшиеся проценты приходится немалое количество данных, хранящихся в структурах, которыя определяются совершенно другими моделями данных со своей специфической семантикой. В этих условиях не представляется возможным иметь согласованный доступ одновременно ко всем источникам информации.

2) Различные способы хранения данных (файлы, БД, хранилища и т.д.). Физическая организация хранения информации создает дополнительные препятствия к ее использованию. Интеграция данных должна предоставить единый логический формат организации данных таким образом, что независимо от способа их физического хранения, конечный пользователь имеет единый механизм доступа к содержимому.

3) Существенная распределенность данных. Источники информации изолированы друг от друга, каждый из которых подчиняется концепции «замкнутого мира». Такой подход значительно затрудняет введение принципиально новых понятий различных предметных областей, порождает дублирование данных, что приводит к увеличению объема, но уменьшению релевантности искомой информации. Интеграция данных способна устранить такую изолированность источников друг от друга, тем самым способствуя согласованному использованию уже существующих данных, устранение дубликатов, а также оперативному возникновению новой информации.

4) Неполнота и противоречивость данных. Отсутствие семантической составляющей современных источников порождает проблему неполноты сведений каждого из них в отдельности. А при рассмотрении совокупности этих источников возникает проблема противоречивости. Интегрция данных призвана устранить эти недостатки путем введения единого семантического контекста для всех информационных ресурсов, хранящихся в интегрированных источниках.

5) Различные сопсобы оперирования данными (манипулирование, поиск, выборка и т.д.). Существующие возможности поисковых систем общего назначения не позволяют обеспечить эффективный поиск информации. Каждая модель данных предполагает существоввания своих собственных средств манипулирования, что порождает их разнообразие, приводящее к гетерогенности данных.

Ввиду всего вышесказанного становится очевидной важность решения комплексной проблемы интеграции.

Проблема интеграции данных заключается в таком логическом объединении данных, принадлежащих разнородным источникам, которое обеспечивает единое представление и оперирование этими данными. Система интеграции данных позволяет освободить пользователя от необходимости самостоятельно отбирать источники, в которых находится интересующая пользователя информация, обращаться к каждому источнику по отдельности и вручную сопоставлять и объединять данные из различных источников.

Акцентируя внимание на разнородности данных, следует прояснить это понятие. Данные разнородны не с точки зрения их физического хранения, а с точки зрения модели их представления. То есть, вне зависимости от места их расположения и способа их хранения, для решения проблемы интеграции важную роль играет модель представления данных со своей специфической семантикой, которая предоставляет механизмы организации работы с данными для конечного пользователя.

В работе [4] авторы выделили следующие признаки неоднородности данных:

1. *Модель*. Структурные различия моделей данных порождают схематическую гетерогенность.
2. *Синтаксис*. Порождается с помощью различных языков описания моделей данных.
3. *Семантика*. Порождается различным определением данных в различных контекстах.

При этом, каждый из этих признаков может присутствовать независимо от двух остальных, например, семантическая гетерогенность может возникать даже в том случае, если схематическая и синтаксическая разнородности отсутствуют (именование концептов и т.д.).

В связи с тем, что далее мы будем много раз говорить о моделях данных, следует дать определение этому понятию.

Данные – представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе. Данные, могут

подвергаться обработке, и результаты обработки фиксируются в виде новых данных.

Модель данных – интегрированный набор понятий для описания и обработки данных, связей между ними и ограничений, накладываемых на данные в некоторой организации.

Цель построения модели данных заключается в представлении данных в понятном виде.

Можно по-разному характеризовать понятие модели данных. С одной стороны, модель данных – это способ структурирования данных, которые рассматриваются как некоторая абстракция в отрыве от предметной области. С другой стороны, модель данных – это инструмент представления концептуальной модели предметной области и динамики ее изменения.

На этапе выработки схем интеграции данных, модель является представлением "реального мира" объектов и событий, а также существующих между ними связей. Это некоторая абстракция, в которой акцент делается на самых важных и неотъемлемых аспектах ПО, а все второстепенные свойства игнорируются. Модель должна отражать основные концепции, представленные в таком виде, который позволит проектировщикам и пользователям обмениваться конкретными и недвусмысленными мнениями о роли тех или иных данных в организации.

Модель данных можно рассматривать как сочетание указанных ниже компонентов [7].

- *структурная часть* (набор правил, определяющих типы и характеристики логических структур данных);
- *управляющая часть*, определяющая типы допустимых операций с данными (описываются правила составления структур более общего типа из структур более простых типов, сюда относятся операции обновления и извлечения данных, а также операции изменения структуры экземпляра модели);
- *набор ограничений* поддержки целостности данных, гарантирующих корректность используемых данных. Сюда входят возможные действия над структурами и правила их выполнения, включающие:

- средства контроля относительно простых условий корректности ввода данных (ограничения);
- средства контроля сколь угодно сложных условий корректности выполнения определенных действий (правила).

Модели данных подразделяются на три категории[13, 14]:

1) *Объектные* (object-based) модели данных (описание данных на концептуальном и внешнем уровнях).

При создании объектных моделей данных используются следующие понятия:

- *сущность* – это отдельный концептуальный элемент ПО
- *атрибут* – это свойство, которое описывает некоторый аспект объекта и значение которого следует зафиксировать.
- *связь* – это ассоциативное отношение между сущностями.

Наиболее общие типы объектных моделей данных:

- ER-модель (Entity-Relationship model);
- семантическая модель (онтология);
- функциональная модель;
- объектно-ориентированная модель (расширяет определение сущности с целью включения в него не только атрибутов, которые описывают состояние объекта, но и действий, которые с ним связаны, т.е. его поведение)

2) Модели данных (record-based) *на основе записей* (описание данных на концептуальном и внешнем уровнях).

В модели на основе записей база данных состоит из нескольких записей фиксированного формата, которые могут иметь разные типы. Каждый тип записи определяет фиксированное количество полей, каждое из которых имеет фиксированную длину.

Существуют три основных типа логических моделей данных на основе записей:

- реляционная модель данных (relational data model);
- сетевая модель данных (network data model),
- иерархическая модель данных (hierarchical data model).

3) *Физические* модели данных (описание данных на внутреннем уровне).

Физические модели данных описывают то, как данные хранятся в компьютере, представляя информацию о структуре записей, их упорядоченности и существующих путях доступа. Физических моделей данных не так много, как логических, а самыми популярными среди них являются обобщающая модель (unifying model) и модель памяти кадров (frame memory).

Возвращаясь к проблематике интеграции, следует обратить внимание на исследования [1] приведенные в нем определения, можно утверждать, что не существует одной единой проблемы интеграции. В то время, как ее основной целью является обеспечить гомогенное, унифицированное представление данных различных источников, конкретная задача интеграции может зависить от множества факторов. Среди них: архитектурное представление информационной системы; содержимое и функциональность систем-компонентов; вид информации, которой оперируют системы-компоненты (числовые данные, мультимедийные данные; структурированные, полу-структурированные, неструктурированные данные); и т.д.

На сегодняшний день мы выделяем три основных составляющих проблемы интеграции данных:

1. выработка схем интеграции данных;
2. выработка отображений между моделями;
3. выработка способов манипулирования, суть которых раскрывается далее.

Выработка схем интеграции данных

Опираясь на исследования [1], а также фундаментальную работу [3] мы рассматриваем 2 типа схем интегрции данных: P2P (peer-to-peer) схема (ещё её называют одноранговой) и централизованая схема.

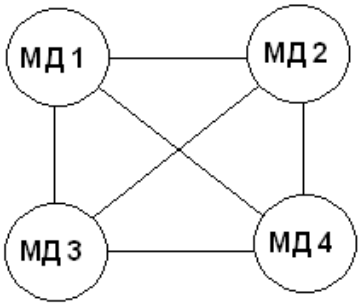

Рис. 1. Одноранговая (P2P) схема интеграции данных

В P2P схеме (рис.1) не существует глобальных точек контроля. В основе каждого узла, принимающего участие в схеме, лежит своя модель данных. Каждый узел равноправен и может принимать запросы пользователя к информации, распределенной по всей системе. Преимущества этой схемы заключаются в следующем: где бы ни был выполнен запрос на информацию, в какой из точек данные ни находились бы, узел, принявший запрос, имеет прямой и непосредственный доступ к каждой точке системы, вследствие чего ему открывается абсолютно вся информация, хранящаяся в ней. Существенным недостатком можно назвать следующее: при добавлении нового узла в схему, необходимо установить соответствия с ним существующих узлов. При небольшом объеме это сделать нетрудно, но с последующим увеличением количество точек, возрастает количество взаимодействий, которые требуется установить внутри схемы, возрастает сложность этих взаимодействий, увеличивается трудоемкость работы проектировщика. Система, основаная на такой схеме становится все более громоздкой и хрупкой, гетерогенность моделей налагает дополнительные сложности на установление связей друг с другом, исходя из особенностей собственных структурных отличий, а также особенностей своих компонент.

Данные недостатки породили развитие другого подхода – централизованой схемы. На сегодняшний день она является наиболее успешной для решения комплексной проблемы интеграции данных. Применяется во многих системах и лежит в основе подходов к выработке отображений между моделями системы, а также к разработке способов манипулирования.

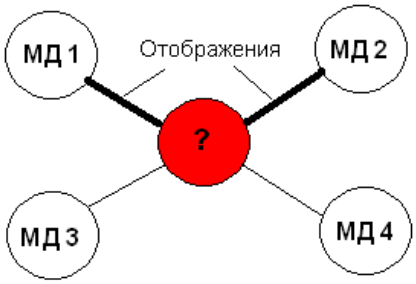

Рис. 2. Централизированая схема интеграции данных

В централизованой схеме (рис.2) обычно присутствует одна глобальная точка контроля. В основе этого узла лежит своя модель данных. В работе [3] ее называют глобальной схемой, а все остальные модели – локальными схемами, или схемами источников. Мы также будем придерживаться этой терминологии в дальнейшем. Основная роль глобальной схемы - предоставление пользователю единого интерфейса для доступа к информации, хранящейся в реальных источниках данных.

Преимуществом такой системы является возможность объединининения любого количество узлов без существенных потерь, т.к. сами локальные схемы могут взаимодействовать между собой любым доступным способом. Главным остается связь с глобальной схемой, обеспечивающей единое **согласованое** представление данных пользователю и предоставление централизованого поиска. Критическим моментом централизованой схемы остается разработка отображений между моделями, а именно схемами источников и глобальной схемой. При рассмотрении подходов взаимодействия глобальной и локальных схем будут рассмотрены недостатки каждого из подходов, которые в целом являются недостатками всех централизованой схемы интеграции данных.

В своих исследованиях, при выработке схем мы остановились именно на втором подходе, а именно на централизованой схеме интеграции данных. Развивая далее эту тему, возникает вопрос, а какую же модель данных выбрать в качестве глобальной схемы? Рассмотрев современные модели данных, наиболее подходящей для выполнения задачи предоставления пользователю единого согласованого представления данных, является онтологическая модель или онтология.

В свое время было сформулировано понятие **семантической интеграции данных** как процесса использования концептуального представления данных, а также их взаимоотношений для ликвидации возможных неоднородностей [4].

Мы уточнили это определение следующим образом.

Семантическая (онтолого-ориентированная) интеграция данных – использование онтологии в качестве объединяющей модели для:

- описания и поддержания отображений между различными моделями данных;
- унифицированного манипулирования данными.

Использование онтологий для семантической интеграции данных аргументируется следующими факторами:

- онтология является самой развитой моделью данных;
- онтологии обладают более развитой семантикой;
- онтологии предоставляют самые мощные механизмы вывода;
- онтологии имеют четкую формальную спецификацию (дескриптивная логика).

Мы понимаем онтологию в ее стандартном, классическом определении, которые было сформулировано много лет назад, а именно,

Онтология – это формальная, явная спецификация согласованной концептуализации [5].

Онтология, как модель данных, представляется следующими компонентами:

1. Структура.
   - классы – концептуальное представление некоторых общих понятий;
   - индивиды – конкретные экземпляры класса;
   - свойства – позволяют утверждать общие факты о классах и специфические факты об индивидах.
2. Ограничения целостности.
   - отношения - таксономия классов, таксономия свойств, принадлежность индивида классу, область определения и область значений свойства (способы, с помощью которых классы и индивиды могут быть связаны друг с другом);
   - правила – способ задания других видов ограничений, которые не поддерживаются отношениями.
3. Операции
   - теоретико-множественные операции на классах и свойствах (объединение, пересечение, дополнение);
   - ограничения свойств по существованию и общности (квантификация свойств);
   - численные ограничения свойств (функциональные, количественные, качественные);
   - и другие.

Как видно из всего выше сказанного, используя централизованую схему интеграции данных для решения комплексной проблемы интеграции, онтология наилучшим образом подходит в качестве глобальной схемы, что позволяет в качестве локальной схемы использовать любую модель данных. Вопрос взаимодействия внутри такой системы относится к следующей общей проблеме интеграции – выработке отображений между моделями.

Выработка отображений между моделями

Рассмотрим абстрактную систему интеграции данных, основанную на архитектуре центральной схемы. Задача такой

системы, называемой также посредником, заключается в том, чтобы предоставить интегрированный доступ к множеству распределенных, разнородных, автономно разработанных источников, без необходимости централизовано хранить всю информацию из источников. Система предоставляет пользователю возможность формулировать запросы на выборку информации из таких источников в терминах глобальной схемы данных (общей системы понятий), которая проектируется «сверху» исходя из интересующих пользователя аспектов предметной области.

При этом в каждом источнике информация может представляться в терминах собственной схемы данных (системы понятий), соответственно, при включении источника в систему указывается некоторое семантическое отображение между терминами глобальной схемы данных и терминами различных схем данных источников.

В работе [3] дается следующее формализированное определение. Система интеграции данных (СИД) I представляется тройкой $\langle G,S,M\rangle$, где

- G – *глобальная схема*, описанная в языке L_G над алфавитом A_G. Алфавит содержит символы каждого элемента G (отношения, если G – реляционная, классы, если G – объектно-ориентированная). В на шем случае, алфавит содержит символы, соответствующие всем концептам и ролям онтологии.

- S – *схема источника*, описанная в языке L_S над алфавитом A_S. Алфавит содержит все символы источника.

- M – *отображения* между G и S, образованные набором утверждений в форме $q_S \approx q_G$, $q_G \approx q_S$, где q_S, q_G – два запроса одинаковой арности, сформулированных в языке $L_{M,G}$ и $L_{M,S}$ соответственно. Запись $q_S \approx q_G$ означеает, что каждый концепт источника, представленный запросом q_S соответствует концепту глобальной схемы, представленной запросом q_G (аналогичным образом трактуется утверждение $q_G \approx q_S$).

По утверждению автора, данное определение охватытвает все подходы, известные в литературе, но каждый специфический подход зависит только от характеристик отображений и выразительной мощности схем и языков формулирования запросов.

Предлагается два подхода, определяющие отображения в СИД. Они называются LAV (Local-as-view) и GAV(Global-as-view).

LAV

В СИД $I=\langle G,S,M\rangle$, основанной на LAV-подходе, отображение M связывает каждый элемент s схемы источника S с запросом q_G к схеме G. Другими словами, язык запросов $L_{M,S}$ разрешает только выражения, образованные одним символом алфавита A_S. Таким образом, LAV отображение – это набор утверждений, по одному на каждый элемент s из схемы S, в форме $s \approx q_G$.

С точки зрения моделирования, подход LAV основывается на идее, что каждый элемент источника s должен быть связан запросом q_G с соответствующим элементом глобальной схемы. Запрос формулируется в языке источника с последующим переформулированием в терминах G. Добавление нового источника сводится к обогащению набора отображений новыми утверждениями без прочих изменений.

GAV

В GAV-подходе отображения M связывают каждый элемент g схемы данных G запросом q_S с элементом источника S. Другими словами, язык запросов $L_{M,G}$ разрешает только выражения, образованные одним символом алфавита A_G. Таким образом, GAV-отображение – это набор утверждений, по одному на каждый элемент g из схемы G, в форме $g \approx q_s$.

С точки зрения моделирования, подход GAV основывается на идее, что каждый элемент g глобальной схемы должен быть связан запросом q_S с соответствующим элементом с выбранным источником данных. Отображение говорит нам, как нужно извлечь данные из источника, когда кто-то хочет оценить различные данные глобальной схемы.

Главным в подходе является обработка запросов, т.к. с их помощью система знает как использовать источники для извлечения данных. Однако, добавление нового источника является серьезной проблемой, т.к. некоторые элементы глобальной схемы должны быть переопределены.

У каждого их этих подходов есть свои преимущества и недостатки.

1) В подходе LAV сложно сформулировать запрос. Представление элемента в ГС одно, а запрос формируется в терминах ИД (алфавит ИД, язык ИД). Но добавление нового ИД не является проблемой, т.к. формулирование запросов – задача самого источника.

2) В подходе GAV легко сформулировать запрос, т.к. мы сразу знаем, какой запрос к ИД соответствует элементу ИД. Представление элемента едино, алфавит и язык формулирования запросов един. Но добавление нвого источника является проблемой, т.к. некоторые представления необходимо бедт переопределить для формулирования запросов и к новому источнику тоже.

3) В то время, как проектировщик LAV концентрируется на том, как представить данные источника в терминах ГС, проектировщик GAV решает проблему, как извлечь необходимые данные из предоставленных источников.

4) Подход LAV нужен для задач, в которых много разнородных ИД, но объем данных не сильно велик. Подход GAV нужен для задач с небольшим количеством источников, но с очень большим объемом данных.

Принципиально новым является суть понятия отображения. Оно представляет собой **запрос**, а не очередное отношение между элементами моделей. Это означает, что в основе взаимодействия между элементами моделей лежит некоторый логический аппарат конкретного языка формулирования запроса.

В своей работе [5] автор предлагает создавать так называемые «обертки» для каждого из источников системы. Они представляют собой локальные схемы, представленные в той же самой модели данных, что и глобальная схема. Предполагается, что каждый информационный источник «обернут» промежуточным компонентом-адаптером, который отвечает за выборку сведений из источника в рамках единой модели данных, а также за предоставление стандартного технического интерфейса для обращения к источнику (сетевой протокол, язык запросов). Пользователь не взаимодействует с источниками напрямую, а обращается к выделенному компоненту-посреднику, который отвечает за обслуживание пользовательских запросов и взаимодействие с источниками. «Обертывание» каждого источника информации в локальную онтологию позволяет развиваться онтологии-источнику вне зависимости от других онтологий. Следовательно, задача интеграции может быть упрощена и добавление или удаление источников можно легко поддерживать.

Выработка способов манипулирования

1) После выработки отображений между моделями данных возникает вопрос применимости таких систем, то есть каким образом можно манипулировать созданными глобальными схемами и управлять данными, расположенными внутри различных систем. «Обертывание» источников данных углубляет этот вопрос тем, что дает возможность использовать эти источники вне существующей системы, а также расширяет возможности манипулирования данными на уровне единой принятой модели.

2) Поскольку мы сводим всю комплексную проблему интеграции данных к онтологической модели, то ввиду этого остро возникает проблема манипулирования онтологиями. Решая эту проблему, мы решаем в полной мере общую проблему интеграции.

3) В проблеме манипулирования онтологий важны следующие два аспекта: выработка подходов по интеграции онтологий и определение множества операций манипулирования онтологиями, которые обсуждаются далее.

Интеграции онтологий

В работе [2] дается три определения интеграции онтологий:

1) ***Интеграция как повторное использование.***

В данном случае, интеграция онтологий рассматривается как процесс создания новой онтологии с помощью повторного использования уже существующих, доступных онтологий (путем сборки, расширения, специализации, адаптации).

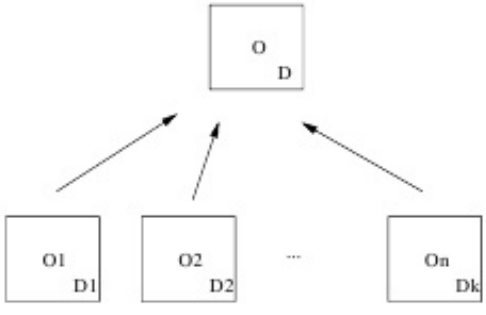

Рис. 3. Интеграция онтологий: повторное использование

В процессе интеграции существуют одна или несколько первоначальных онтологий $(O_1, O_2, ..., O_n)$, а также итоговая онтология O, которая образуется в результате процесса интеграции. Домены $(D_1, D_2, ..., D_k)$ могут отличаться от результирующего домена D, но между ними могут существовать связи. При этом, обычно $k = n$, но такое может быть не всегда, т.к. в процессе интеграции могут участвовать несколько различных онтологий принадлежащих одному и тому же домену. В результате процесса интеграции образуется онтология O такая, что аналогичной не существует. В противном случае одна из них должна будет повторно использовать другую.

2) ***Интеграция как объединение.***

В данном случае, интеграция онтологий рассматривается как процесс создания новой онтологии с помощью объединения нескольких онтологий в одну которая обобщает их все.

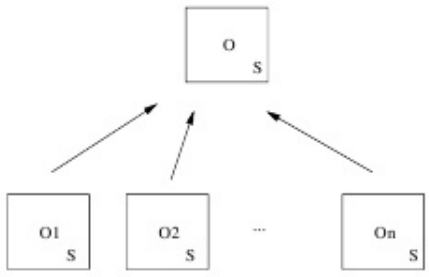

Рис. 4. Интеграция онтологий: объединение

В процессе интеграции участвуют несколько первоначальных онтологий онтологий $(O_1, O_2, ..., O_n)$, а также итоговая онтология O, которая образуется в результате процесса интеграции, которую в этом случае иногда называют объединением. Начальные онтологии принадлежат одному и только одному домену S, которому также принадлежит результирующая онтология. Целью данного процесса является создание более общей онтологии на заданном домене, собирая в единое целое знания нескольких онтологий этого домена. Уровень обобщенности первоначальных онтологий может отличаться.

3) ***Интеграция как использование в программном обеспечении.***

В данном случае, интеграция онтологий рассматривается как процесс создания программного приложения, основанного на использовании нескольких онтологий.

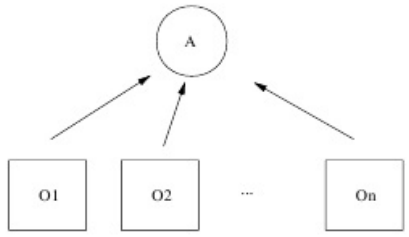

Рис. 5. Интеграция онтологий: использование

В процессе интеграции участвуют несколько первоначальных онтологий онтологий $(O_1, O_2, ..., O_n)$, но в результате не создается никакой новой онтологии. Некоторое приложение A просто использует готовые онтологии, а результат зависит от архитектуры и назначения самого приложения. Онтологии должны быть совместимы между собой по следующим критериям: язык описания, онтологические соглашения, уровень детализации, уровень обобщения, модульность, контекст и т.д.

Операции над онтологиями

Что касается операций манипулирования онтологиями, то можно выделить следующие два вида операций над онтологиями: сопоставление и оперирование. Сопоставление решает проблему установления различного рода (семантических) соответствий между онтологиями.

Оперирование – это набор унарных и бинарных операций создания новых онтологий из существующих. Мы кратко представим только операции сопоставления, как наиболее важные при решении проблемы интеграции онтологий.

Уточнение (refinement). Под уточнением онтологий понимают такое сопоставление онтологии A с другой онтологией B, что каждому понятию из онтологии A ставится в соответствие эквивалентное ему понятие в B. Примитивные понятия из онтологии A могут соответствовать непримитивным понятиям онтологии B.

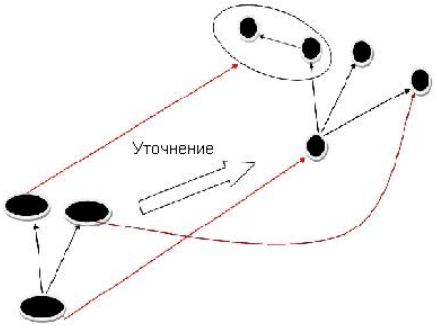

Рис. 6. Уточнение

Унификация (unification). Онтология приводится к некоему каноническому (эталонному) представлению. Для унификации должна задаваться исходная онтология, которая приводится к результирующей согласно заданной канонической онтологии. Задача унификации множества исходных онтологий становится актуальной при работе с гетерогенными онтологиями.

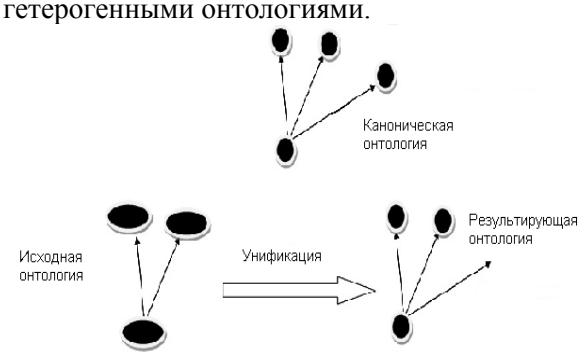

Рис.7. Унификация

Отображение (mapping). Отображение одной онтологии в другую – это функция преобразования одной онтологии в другую (способ перевода объектов одной онтологии в другую), либо сам результат такого преобразования. Часто это означает перевод между понятиями и отношениями. Отображение может быть частичным в том смысле, что не все понятия исходной онтологии отображаются в результирующую. В частности, это означает, что в исходной онтологии существует подонтология, для которой существует полное отображение.

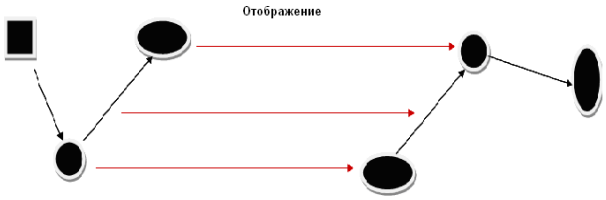

Рис. 8. Отображение

Согласование (alignment). Это процесс отображения онтологий в обоих направлениях. Согласование, как и отображение, может быть лишь частичным. Спецификация согласования называется артикуляцией (articulation).

Интеграция (integration). Это процесс поиска одинаковых частей двух разных онтологий, A и B, при разработке новой онтологии C, которая позволяет выполнить перевод между онтологиями A и B, и, таким образом, позволяет взаимодействие между двумя системами, где одна использует онтологию A, а другая - онтологию B. Новая онтология C может заменить онтологии A и B или может использоваться в качестве промежуточной онтологии для перевода между двумя онтологиями. Интеграция может меняться от согласования к унификации.

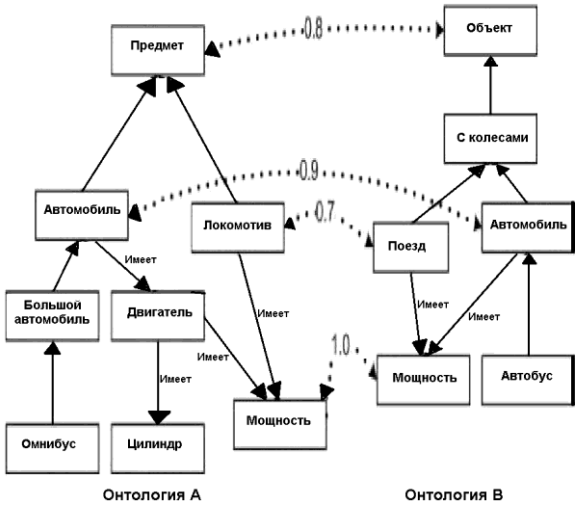

Рис. 9. Согласование

Наследование (inheritance). Означает, что онтология A наследует все из онтологии B. Она наследует все понятия, отношения и ограничения или аксиомы, и дополнительные знания, содержащиеся в онтологии, не внося при этом какой-либо несогласованности.

Выводы

Интеграция данных в информационном пространстве является большой научной проблемой. Существует множество подходов к её решению. Было выявлено три составляющие комплексной проблемы интеграции, в процессе рассмотрения которых мы остановились на централизированой схеме интеграции данных и онтологической модели, в качестве единой модели на роль глобальной схемы данных. Приведена аргументация данного выбора, дана характеристика онтологии как модели данных, проанализированы способы манипулирования онтологиями.

Список использованных источников

1. Ziegler P. Three Decades of Data Integration – All Problems Solved? / Patrick Ziegler, Klaus R. Dittrich // Report of Database Technology Research Group, Department of Informatics, University of ZurichWinterthurerstrasse 190, CH-8057 Zürich, Switzerland.

2. Pinto H.S.. Some Issues on Ontology Integration / H. Sofia Pinto // Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5). – Stockholm, Sweden, August 2, 1999. – P. 124 – 136.

3. Lenzerini M. Data Integration: A Theoretical Perspective / M. Lenzerini // Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2002). – N. Y.: ACM Press, 2002. – 233 p.

4. Cruz I.F. The Role of Ontologies in Data Integration./ Isabel F. Cruz, Huiyong Xiao // Jounal of Engineering Intelligent Systems. – 2005. – № 4. – P. 68 – 83.

5. Guarino N. Ontologies and Knowledge Bases: Towards a Terminological Clarification / Nicola Guarino, Daniel Oberle, and Steffen Staab // Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing. – 1995. – P. 25 – 32.

6. Hovy E. Ontologies: lecture 1, lecture 2, Issues of Content, lecture 3/ E. Hovy // Methods for Automated Ontology Building. Information Sciences. – London, 2004. – P. 136 – 148.

7. Бездушный А.А. Математическая модель системы интеграции данных на основе онтологий / А.А. Бездушный // Журнал «Вестник НГУ», серия «Информационные технологии». – Новосибирск, 2008. – Т.6, вып.2. – С. 15 – 40.

8. Безэкземплярные и экземплярные онтологии [Электронный ресурс]. – Режим доступа: http://skowal.narod.ru/research/ontology2007

9. Овдей О.М. Обзор инструментов инженерии онтологий./ Овдей О.М., Проскудина Г.Ю.// Электронные библиотеки. – Москва, 2004. – Т.7, вып. 4. – С. 86 – 102.

10. Константинова, Н.С. Онтологии, как системы хранения знаний Электронный ресурс. / Н.С. Константинова, О.А. Митрофанова. [Электронный ресурс]. – Режим доступа: http://www.ict.edu.ru/ft/005706/68352e2-st08.pdf

11. Слесарев Е.В Преимущества семантических технологий: практический аспект. / Е.В. Слесарев // Электроника и информационные технологии. – Москва. – 2012. – №1(12). – С. 57 – 64.

12. Auer S. Integrating Ontologies and Relational Data. Technical Reports (CIS) / Auer S., Ives Z.G.// Report of University of Pennsylvania, 2007. – 164 p.

13. Данные [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki/Данные

14. База знаний кафедры ИКТ. Лекция №07. Модели данных [Электронный ресурс]. – Режим доступа: http://wiki.auditory.ru/БД:Лекция_№07

15. Linked Open Vocabularies (LOV) [Electronic resource]. – Access mode: http://lov.okfn.org/dataset/lov/

16. 16. Protege Ontology Library. [Electronic resource]. – Access mode: http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library

17. The Open Biological and Biomedical Ontologies [Electronic resource]. – Access mode: http://www.obofoundry.org/

18. TONES ontology repository [Electronic resource]. – Access mode: http://rpc295.cs.man.ac.uk:8080/repository/

19. DAML Ontology Library [Electronic resource]. – Access mode: http://www.daml.org/ontologies/

Сведения об авторе:

Чистякова Инна Сергеевна – аспирантка Института программных систем НАНУ. Научные интересы: семантический веб, семантическая интеграция данных, онтологические модели данных.

E-mail: inna_islyamova@ukr.net