

СИСТЕМЫ ДОСТУПА К ДАННЫМ, ОСНОВАННЫЕ НА ОНТОЛОГИЯХ

© 2012 г. М.Р. Когаловский
Институт проблем рынка РАН
117418 Москва, Нахимовский проспект, 47
E-mail: kogalov@cemi.rssi.ru
Поступила в редакцию 15.01.2012 г.

Исследования, направленные на обеспечение семантического доступа к базам данных, имеют многолетнюю историю и начались еще на ранних стадиях развития технологий баз данных. К сожалению, они все еще не привели к созданию широко распространенных индустриальных технологий. В последнее десятилетие деятельность консорциума W3C в области Семантического Веба и создание стандартов языков описания онтологий вызвали новую волну активности в разработках инструментария для систем семантического доступа к базам данных и нового класса систем баз данных, называемых системами доступа к данным, основанными на онтологиях (OBDA-системам). В таких системах онтологии используются в качестве концептуальной схемы предметной области и основы пользовательского интерфейса для SQL-систем баз данных. Предложенные в последние годы подходы не обеспечивают “окончательного” решения проблемы. Тем не менее, удалось создать языки описания онтологий, позволяющие достигнуть приемлемого компромисса между их выразительностью, которая остается достаточной для многих приложений, и вычислительной сложностью рассуждений на онтологиях и обработки запросов данных, хранимых в больших базах данных. Созданы предпосылки для появления индустриальных технологий разработки систем указанного класса. В предлагаемой работе приводится обзор полученных в последние годы основных результатов исследований и разработок в рассматриваемой области.

1. ВВЕДЕНИЕ¹

Достижения в области создания формальных языков описания онтологий для Всемирной паутины и их стандартизации консорциумом W3C, разработки инструментальных средств для описания и поддержки онтологий привели к активной экспансии онтологий в разработки веб-приложений, текстовых систем, систем баз данных и баз знаний. Открылись новые возможности для создания систем, обеспечивающих пользователям семантический доступ к информационным ресурсам. В технологиях баз данных формируется направление, в котором предусматривается использовать для этой цели онтологии в качестве

высокоуровневых концептуальных схем предметной области и основы пользовательских интерфейсов SQL-систем баз данных, позволяющих формулировать запросы данных в терминах онтологий и вместе с тем осуществлять рассуждения (reasoning) на онтологиях. Механизмы основанного на онтологии интерфейса становятся при этом посредником между пользователем с его информационными потребностями, выраженными в терминах онтологии, и традиционной системой базы данных. Такие системы называют *системами доступа к данным, основанными на онтологиях* (Ontology-Based Data Access Systems, OBDA-Systems, OBDA-системами).

В наших работах [4, 5] рассматривался вопрос о часто встречающихся на практике подменах

¹Работа поддержана грантом РГНФ 11-02-12026-в.

концептуальной схемы предметной области ее онтологией, детально обсуждались различия между ними. Такая подмена характерна и для OBDA-систем. Предлагаемые в этой области подходы используют язык описания онтологий в качестве языка концептуального моделирования предметной области.

Проблематике систем доступа к данным, основанных на онтологиях, посвящено большое количество публикаций в последние годы. В этой нашей работе рассматриваются подходы к созданию таких систем с использованием онтологий, представленных дескриптивными логиками (Descriptive Logic, DL) [83].

В 1990-х годах был выполнен ряд проектов в американских университетах, а также коллективами, финансируемыми DARPA и Европейской комиссией, целью которых было создание языков описания онтологий Веб на основе дескриптивных логик. Гибрид созданных в результате этих проектов языков – DAML и OIL послужил прототипом языка описания онтологий OWL, одобренного консорциумом W3C (2004) в качестве одного из ключевых стандартов технологий Семантического Веба. Позднее (2009) была одобрена новая версия этого стандарта – язык OWL2 [57, 58] вместе с его профилями (EL, QL и RL) [59] – подязыками, в которых выразительные возможности OWL2 ограничиваются для достижения эффективности рассуждений в некоторых важных областях применения. Стандартизация языка описания онтологий способствовала довольно широкому его применению, активизации использования описываемых его средствами онтологий в разработках систем баз данных и информационных систем.

Исследования в области дескриптивных логик проводятся уже более четверти века. Они особенно активизировались в связи с разработками языков описания онтологий в последнее десятилетие. Среди исследований этого периода выделяется большой комплекс работ группы сотрудников итальянских университетов Free University of Bozen-Bolzano и Universit'a di Roma "La Sapienza". Ими предложено семейство дескриптивных логик DL-Lite [22, 23, 27], одна из которых – DL-LiteR послужила основой профиля OWL2

QL [59]. Предложены также конструктивно проработанные и экспериментально проверенные на исследовательских прототипах подходы к созданию систем интеграции данных и OBDA-систем на основе логик указанного семейства [23, 25, 26, 28, 29].

В этой статье обсуждение OBDA-систем опирается, главным образом, на указанные работы. Термины *доступ к данным, основанный на онтологиях* (Ontology-Based Data Access, OBDA), и *системы доступа к данным, основанные на онтологиях* (Ontology-Based Data Access Systems, OBDA-Systems), были введены именно их авторами.

Нужно отметить, что группой итальянских исследователей предложен также подход к интеграции данных из множества источников с использованием логик указанного семейства. Поэтому при обсуждении OBDA-систем в указанном цикле работ предполагается, что благодаря возможности применения методов интеграции данных источник данных в OBDA-системе может рассматриваться в общем случае как единая SQL-система базы данных.

В данном обзоре обсуждаются предыстория и предпосылки создания OBDA-систем в технологиях управления структурированными данными, основные идеи и принципы их реализации. При этом акцент в обсуждении делается на технологических его аспектах. Поскольку в качестве источников информационных ресурсов в OBDA-системах рассматриваются базы данных, указанные вопросы обсуждаются именно в контексте технологий баз данных.

Остальная часть статьи организована следующим образом. Прежде всего, в разделе 2 обсуждается концепция OBDA-системы и требования к ней. Далее в разделе 3–5 приводится краткий обзор тех направлений теории и технологий баз данных, достижения в которых создали предпосылки для рождения обсуждаемого класса систем. Прежде всего, это имеющие давнюю историю и обсуждаемые в разделе 3 и 4 технологии семантического доступа пользователей к базам данных. Их эволюция потребовала, в свою очередь, развития методологий, создания концептуальных моделей данных и языков концептуального моде-

лирования предметной области. При этом большое внимание исследователей привлекали поддержка абстрактной модели предметной области, представленной ее концептуальной схемой, непосредственно механизмами СУБД и ее использование как основы пользовательского интерфейса системы базы данных. Реализация таких возможностей обеспечивает сближение систем баз данных с системами, основанными на знаниях. В последнее время появился ряд публикаций, посвященных анализу истории, эволюции, состояния и перспектив концептуального моделирования. Среди них наиболее достойны внимания, по нашему мнению, фундаментальная монография [56], подготовленный по материалам Дагштульского семинара по эволюции концептуального моделирования сборник статей [45], а также обзоры [71, 88]. Мы ограничимся здесь кратким обсуждением этих вопросов в той степени, в которой это необходимо для целей данной статьи.

В разделе 5 обсуждается другое имеющее более чем сорокалетнюю историю направление исследований, предопределившее создание дескриптивных логик – основы OBDA-систем. Это – получившее в последние годы новый импульс для развития направление, связанное с использованием логического аппарата в технологиях баз данных. Первые его шаги связаны с рождением реляционного подхода в области баз данных. Позднее были предложены идеи дедуктивных и дедуктивных объектно-ориентированных баз данных. В последние годы главное внимание в этом направлении привлекают формальные языки представления знаний, в частности языки описания онтологий. Интерес к ним особенно активизировался в рамках деятельности по созданию технологий и стандартов Семантического Веба. Особенности дескриптивных логик, играющих важную роль в этих разработках, и возможности их применения в OBDA-системах обсуждаются в разделе 6.

На основе материалов предыдущих разделов в разделе 7 обсуждаются принципы организации и функционирования OBDA-систем. В разделе 8 рассматривается созданный для их разработки инструментарий, приводятся сведения о прототипах инструментальных средств для

создания и поддержки онтологий, а также для реализации OBDA-систем. Наконец, в разделе 9 дается оценка достоинств и ограниченности предложенного подхода к созданию систем этого класса. В заключении подводятся итоги обзора и обсуждаются перспективы распространения технологий OBDA-систем.

2. OBDA-СИСТЕМЫ И ТРЕБОВАНИЯ К НИМ

Как уже указывалось, термин *доступ к данным, основанный на онтологиях*, был введен авторами семейства логик DL-Lite (см. например [25]), которые предложили их не только для описания онтологий, для чего они первоначально разрабатывались, но и для создания систем интеграции данных, а также OBDA-систем. Заметим, что наряду с термином *Ontology-Based Data Access System* в литературе другими авторами используются также близкие термины: *Accessing Data Mediated by an Ontology*, *Ontology-driven Information Systems*, *Ontology based Data Management*.

OBDA-система рассматривается в работах авторов семейства DL-Lite как *надстройка над множеством существующих источников структурированных данных*, позволяющая предоставить пользователям системы концептуальное представление содержащихся в них данных, возможность рассуждений и доступа к данным в терминах этого представления. При этом в качестве концептуальной схемы интегрированного источника данных предлагается использовать *формальную онтологию предметной области*, описанную средствами дескриптивной логики семейства DL-Lite. Одновременно предложен подход к решению на этой основе проблемы семантической интеграции источников данных [26]. Поэтому предлагается в общем случае рассматривать в качестве интегрированного источника данных OBDA-системы реляционную SQL-систему базы данных.

Важное предположение заключается в том, что система базы данных может уже существовать и разрабатываться независимо от онтологии, которую предстоит создать и связать с этой системой. Процесс создания OBDA-системы начинается с разработки

онтологии предметной области, данные которой хранятся в SQL-системе базы данных.

В качестве наиболее значимых возможностей OBDA-систем рассматриваются:

- предоставление развитых формальных выразительных средств для представления базы данных и спецификации запросов;
- обеспечение декларативности запросов в терминах такого представления;
- наличие механизмов для рассуждений на онтологии, а также для обработки сформулированных в терминах онтологии запросов данных в системе базы данных; последнее, естественно, влечет необходимость отображения запроса, сформулированного в терминах онтологии, в запрос, специфицированный средствами языка SQL;
- способность осуществлять рассуждения и обработку запросов данных с приемлемой производительностью.

Наряду с этим, использование онтологии в качестве концептуальной схемы позволяет обеспечить ряд других полезных возможностей. В частности, обеспечивается более абстрактное представление базы данных, чем при использовании традиционных моделей данных, и оно не связано с “логической” структурой базы данных. Появляется возможность использовать в запросах явно не определенные в системе базы данных (скрытые) отношения. Может быть проверено качество данных. В результате может обнаруживаться неожиданная неполнота или противоречивость данных в базе данных. В качестве побочного эффекта осуществляется упорядочение и документирование терминологии организации, использующей OBDA-систему.

Для практически применимой реализации указанных выше возможностей OBDA-система должна удовлетворять нескольким важным требованиям. Перечислим основные из них.

Одна из главных функций OBDA-системы – обеспечение рассуждений на поддерживаемой в ней онтологии. Хотя эта онтология создается согласованно с используемой независимо

разработанной системой базы данных, она строится в соответствии с *гипотезой открытого мира* в отличие от базы данных, создаваемой в соответствии с *гипотезой замкнутого мира*. Поэтому при обработке запросов в терминах онтологии может обнаруживаться *неполнота или противоречивость данных* в базе данных. Это обстоятельство должно учитываться при обработке запросов данных.

Другое важное требование – должна обеспечиваться *приемлемая сложность* выполнения рассуждений, а также обработки сформулированных в терминах онтологии запросов к базам данных большого объема, с которыми приходится иметь дело в реальных приложениях.

Далее, как уже указывалось, в OBDA-системах в качестве *концептуальной схемы*, поддерживаемой пользовательским интерфейсом, предполагается использовать онтологию. Сформулированный в терминах такой концептуальной схемы запрос данных должен отображаться в среду реляционной SQL-системы базы данных, которая будет, в конечном счете, обрабатывать конвертированные запросы. Важно, чтобы накладные расходы на выполнение такого отображения не были значительными.

Наконец, поскольку в качестве концептуальных схем в OBDA-системах предполагается использовать формальные онтологии, необходимо, чтобы используемая для их представления логика позволяла выражать основные модельные элементы, свойственные традиционно используемым языками концептуального моделирования. Наиболее распространенными из них в настоящее время являются ER-диаграммы и диаграммы классов языка UML [43].

Авторы концепции OBDA-системы разработали для описания онтологий семейство дескриптивных логик DL-Lite, которые ценой некоторых ограничений на выразительные возможности позволяют удовлетворить указанные требования.

Прежде чем перейти теперь к обсуждению принципов организации и функционирования OBDA-систем, а также имеющихся инструментальных средств для их разработки,

представляется необходимым предварительно привести краткий обзор эволюции тех областей теории и технологий баз данных и информационных систем, которые обусловили рождение концепции OBDA-системы. К этим областям относятся, как уже отмечалось, концептуальное моделирование предметной области, использование формальной логики в технологиях баз данных, а также создание дескриптивных логик.

3. КОНЦЕПТУАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

На ранней стадии развития технологий баз данных проектирование баз основывалось на интуиции и опыте разработчика. Не существовало каких-либо широко признанных методологий и инструментальных средств для этой цели. В конце 1960-х – начале 1970-х гг. технологии баз данных оформились как значимая самостоятельная ветвь информационных технологий. Началось производство СУБД общего назначения, активизировались практические разработки систем баз данных. Стали востребованными методологии проектирования баз данных и инструментарий для этой цели. Начали формироваться CASE-технологии, обеспечивающие, в частности, функции проектирования и реализации систем баз данных. С появлением персональных компьютеров широкое распространение получили средства быстрой разработки приложений (Rapid Application Development, RAD), основанные на 4GL. Такие средства стали непременным функциональным компонентом СУБД, предназначенных для этой аппаратной платформы. В настоящее время существует развитая индустрия и рынок программных продуктов, поддерживающих различные CASE-технологии. Без их использования сегодня немислимы проектирование и разработка любой сложной системы базы данных или информационной системы.

В соответствии со сложившимися методологиями проектирования баз данных, начальным этапом этого процесса является формирование спецификации абстрактного представления предметной области, неза-

висимого от информационных технологий, которые предполагается использовать для реализации разрабатываемой системы. Такая спецификация представляется с помощью подходящих выразительных средств – *языков концептуального моделирования*, а этот этап называется *концептуальным моделированием* предметной области.

Как указывается в [44], концептуальное моделирование – это деятельность, обеспечивающая создание концептуальных моделей, т.е. моделей, которые описывают задачи независимо от технологии и стратегии, используемых для их решения. При этом моделируются сущности предметной области, их концептуальные структуры, характерные отношения между ними и их поведение. Иначе говоря, концептуальное моделирование – это процесс выявления, анализа и описания релевантных его целям сущностей предметной области, взаимосвязей между ними, ограничений, которым они должны удовлетворять, а также их поведения. Результатом концептуального моделирования является абстрактное описание предметной области, которое называют *концептуальной моделью предметной области*. Степень формальности концептуальной модели зависит от характера языка, используемого для ее представления. В технологиях баз данных эту модель называют *концептуальной схемой*.

Заметим, что концептуальная схема представляется в терминах не экземпляров, а *типов сущностей* предметной области и *типов связей* между ними, а также налагаемых на них ограничений целостности. При использовании объектной парадигмы при концептуальном моделировании предметной области базовыми модельными элементами являются *объекты*, обладающие *атрибутами*, значения которых характеризуют их состояние, и *поведением*, которое описывается методами (операциями), а также *бинарными связями объектов*. Концептуальная схема предметной области представляется в этом случае совокупностью типов объектов и типов связей между ними. В ней определяется также *иерархия отношений наследования* между типами объектов. Концептуальная схема является, таким образом, *интенциональной моделью* предметной области.

В традиционных технологиях разработки систем баз данных концептуальная схема предметной области используется только *на стадии проектирования* базы данных. Синтезированная концептуальная схема отображается в среду СУБД, выбранной для реализации системы. Результат отображения представляется средствами языков определения данных этой конкретной выбранной СУБД. При этом формируются новые модели предметной области, которые, в отличие от абстрактных моделей, зависят от информационных технологий, выбранных для реализации. В работах шведской школы (см. например [47, 48, 81, 82]), в которых в 1970-е годы был предложен *инфологический подход* к проектированию информационных систем, такие модели называются *даталогическими*. В трехуровневой архитектуре системы базы данных ANSI/X3/SPARC [16], эти модели называются *концептуальной, внутренней и внешними схемами базы данных*, в соответствии с теми уровнями архитектуры, на которых они поддерживаются. Инструменты для представления схем базы данных и оперирования данными в их терминах называются, соответственно, даталогическими *концептуальной, внутренней и внешними моделями данных*. Эти инструменты поддерживаются механизмами архитектурных уровней системы базы данных.

Концептуальная схема базы данных является результатом отображения концептуальной схемы предметной области в среду данной СУБД и описывает представление базы данных в целом в терминах даталогической концептуальной модели данных, поддерживаемой этой СУБД. *Внутренняя схема базы данных* описывает представление базы данных в среде хранения. Наконец, *внешние схемы базы данных* описывают представление фрагментов базы данных для использующих их приложений. Все названные модели предметной области также являются *интенциональными*, а наполнения базы данных, соответствующие этим схемам, представляют ее *экстенциональные модели*. Даталогические модели предметной области используются *на стадии функционирования* системы базы данных. Они поддерживаются

системой в процессе ее функционирования. Интенциональные даталогические модели и выражаются средствами языков описания данных используемой СУБД, т.е. средствами *моделей данных*, реализованных в данной СУБД. При этом интерфейсы пользователей системы поддерживают представление базы данных в виде *внешних схем*.

Предпринимавшиеся ранее попытки повышения семантического уровня пользовательских интерфейсов систем баз данных заключались в использовании концептуальных схем предметной области в качестве внешних схем в системах баз данных. По существу, эту идею и реализуют авторы концепции OBDA-систем. Однако при этом не создаются СУБД, поддерживающие концептуальную схему предметной области непосредственно на внешнем уровне архитектуры. Вместо этого используются традиционные SQL-СУБД и создаются самостоятельные механизмы поддержки концептуальной схемы предметной области, а также механизмы отображения этой схемы и связанных с нею операционных возможностей в модель данных внешнего уровня архитектуры СУБД. В данном случае эта модель воплощается языком SQL.

Наряду с концептуальной схемой предметной области в последние годы в разработках систем баз данных используют другое абстрактное ее представление, называемое *онтологией предметной области*. В то время как концептуальная схема описывает структуру и, возможно, поведенческие аспекты предметной области, онтология определяет *понятийную систему* предметной области, т.е. совокупность ее понятий (концептов) и отношений между ними. Соотношение между этими абстрактными моделями предметной области подробно обсуждалось в наших работах [4, 5]. Главное различие между ними состоит в том, что концептуальная схема предметной области отображает релевантные целям системы сущности и связи между ними, а онтология – понятия (концепты) предметной области и семантические отношения между ними. Онтология, таким образом, терминологически дополняет описание предметной области, представленное ее концептуальной схемой.

Между тем, это различие часто не осознается. Можно привести множество примеров. Так, по мнению автора [58], “цель концептуального моделирования – описать множество концептов, используемых в предметной области для определения концептуальной схемы памяти, которую должна иметь программная система этой предметной области”. В соответствии с таким пониманием, в последние годы онтологии предметной области часто используют на практике в качестве ее концептуальных схем. Таким образом поступают, в частности, и авторы концепции OBDA-системы. Следует обратить внимание на название упомянутой монографии. Хотя в работе основательно обсуждаются существо и инструментарий концептуального моделирования предметной области, в ее названии речь идет о моделировании информационных систем. Аналогичную подмену объекта моделирования можно найти в статье [88]. В этой статье обсуждаются проблемы концептуального моделирования предметной области в разработках сложных программных систем. Однако вместо концептуальной модели предметной области автор говорит о модели программной системы. Кроме того, автор не замечает различия между концептуальной моделью (концептуальной схемой) предметной области и концептуальной схемой базы данных. Это обстоятельство порождает у него неправомерное мнение о якобы имеющейся путанице в архитектурной модели ANSI/X3/SPARC.

4. ЯЗЫКИ КОНЦЕПТУАЛЬНОГО МОДЕЛИРОВАНИЯ И КОНЦЕПТУАЛЬНЫЕ МОДЕЛИ ДАННЫХ

Выразительными средствами для представления результатов концептуального моделирования предметной области являются разнообразные *языки концептуального моделирования*. Они позволяют описывать структуру предметной области и/или происходящие в ней процессы. Существует большое разнообразие языков концептуального моделирования, различающихся их уровнем формальности, функциональностью и базовыми парадигмами моделирования. В работе [88] предложена классификация языков этого назначения, предусматривающая три их категории: языки,

базирующиеся на сущностях (*entity-oriented*), коммуникациях (*communication-oriented*) и поведении (*behavior-oriented*). Примерами языков первой категории являются нотация ER-модели данных [30] или диаграмма классов языка UML [43], второй категории – диаграммы потоков данных (Data Flow Diagram, DFD) Йордана/ДеМарко [33, 87] или диаграммы стандарта IDEF1 [42], а третьей – нотации потоков работ – сети Петри, диаграммы деятельности UML [43], а также нотация для моделирования бизнес процессов (Business Process Modeling Notation, BPMN) [20].

Эта классификация языков концептуального моделирования – отнюдь не исчерпывающая. Она не учитывает, в частности, степень формальности языков, не определяет места объектно-ориентированных языков. Мы не будем рассматривать ее здесь более подробно.

Языки концептуального моделирования, используемые в технологиях баз данных для представления концептуальной схемы предметной области, основаны на *концептуальных моделях данных*. Важная задача таких моделей – удерживать в концептуальной схеме предметной области по возможности больше семантики. Эта задача может решаться, в частности, путем использования развитых механизмов абстракции, повышения семантической нагрузки на типы связей, более строгого разделения концептуального и даталогического представления предметной области.

Модели данных, обладающие относительно развитыми возможностями этого рода, называются *семантическими* [10, 18, 60]. Разработки таких моделей данных представляли собой обширную сферу активных исследований в период 1970-х–1980-х годов. Большое влияние на них оказали работы Дж. Смита, Д. Смит и Д. МакЛеода по абстракции данных [50, 77–79], обсуждение проблем концептуального моделирования на симпозиуме по абстракции данных, базам данных и концептуальному моделированию [65], а также на состоявшемся позднее симпозиуме по концептуальному моделированию [17], форум для обсуждения проблем концептуального моделирования, который предоставила работающая уже более трех десятилетий ежегодная

международная конференция по концептуальному моделированию (International Conference on Conceptual Modeling) [36], до 1995 года называвшаяся *International Conference on the Entity Relationship Approach*.

В этот период был предложен многочисленный ряд концептуальных моделей данных, базирующихся на различных парадигмах моделирования. Среди них можно назвать семантическую бинарную модель Абриаля [14], модель Сенко и др. DIAM [75], модель сущностей-связей (Entity-Relationship Data Model, ER-модель данных) Чена [30], расширенную реляционную модель Кодда RM/T [32], инфологические модели Лангефорса и Сундгрена [47–48, 81–82], модели Шмида и Свенсона [73], IAM Бубенко [19], SDM Хаммера и МакЛеода [40], ORM/NIAM Хэлпина [39], функциональную модель DAPLEX Шипмана [76] и ряд других. Некоторые ранние семантические модели данных рассматриваются в классической монографии Цикритзиса и Лоховски [10]. Моделированию семантики в базах данных посвящена весьма основательная монография Цаленко [9].

Начали также предприниматься попытки использования формальной логики для целей концептуального моделирования предметной области, которые привели к созданию нового пласта технологий баз данных – *дедуктивных баз данных*. Активизации исследований в этом направлении в значительной мере стимулировалась развитием логического программирования и появлением языка Пролог. Стремление к интеграции технологий логического программирования и технологий баз данных привело к созданию языка запросов для систем дедуктивных баз данных Datalog [37, 86], в котором были использованы идеи Пролога и его синтаксис, преодолены некоторые его слабые стороны. Datalog стал платформой теоретических исследований, проводимых в области дедуктивных баз данных.

Дальнейшая эволюция концептуальных моделей данных связана с применением в этой области объектной парадигмы моделирования. Учитывая доминирующее использование объектных моделей данных в концептуальном моделировании предметной области в настоящее

время, остановимся на этом классе моделей немного подробнее.

В конце 1980-х гг. объектный подход, получивший к этому времени довольно широкое распространение в разработках программного обеспечения, распространился на сферу технологий баз данных и способствовал повышению уровня семантики систем баз данных. Появились первые объектно-ориентированные СУБД. Объектная концептуальная схема предметной области в таких системах может поддерживаться практически без какой-либо трансформации с точностью до способа ее описания. Развитию объектных технологий баз данных в значительной мере способствовала деятельность консорциума ODMG (*Object Data Management Group*, первоначально *Object Database Management Group*), образованного в 1991 году. Опираясь на стандарты CORBA (1991) консорциума OMG (Object Management Group) и языка SQL, ODMG разработал ряд версий стандарта объектной модели данных с комплексом воплощающих ее языков. Принятием последней версии этого стандарта ODMG 3.0 [84] консорциум завершил свою деятельность. Создание стандарта ODMG и стандарта CORBA стимулировало производство инструментальных средств объектных технологий и их практическое использование. Рынок программного обеспечения обогатился рядом объектно-ориентированных СУБД, поддерживающих стандарт ODMG.

Активные разработки объектных систем баз данных, в свою очередь, создали предпосылки для развития методов объектного анализа и проектирования (OA&D) информационных систем. В компании Rational, объединившей усилия крупнейших специалистов по методологиям OAD – Буча, Рамбо и Якобсона, был создан объектный унифицированный язык концептуального моделирования UML (*Unified Modeling Language*). Этот язык стал стандартом OMG (1997), а позднее и стандартом ISO/IEC [43]. Поддержка языка UML стала необходимым условием конкурентоспособности CASE-инструментов, выпускаемых индустрией программного обеспечения.

Популярности объектной парадигмы моделирования способствует ряд ее достоинств.

Действительно, поддержка индивидуальности объектов позволяет моделировать поведение объектов и эволюцию их во времени, поскольку объекты сохраняют индивидуальность при обновлениях. Возможности классификации объектов позволяют поддерживать отношение наследования между классами. Концепция наследования дает возможность отображать семантические отношения между классами, совместно и повторно использовать некоторые фрагменты программного кода. Объектная модель ODMG позволяет конструировать сложные объекты и коллекции объектов, поддерживать сложные структуры связей между ними. Использование объектных СУБД для реализации систем баз данных дает возможность существенно упростить отображение концептуальной схемы предметной области, созданной средствами объектного CASE-инструмента, в *дatalogическую среду*. Не случайно, что на основе объектного подхода создано большое количество методологий проектирования.

Вместе с тем, объектные модели данных не лишены и некоторых слабых сторон. В частности, значительную роль в объектных моделях играет навигационный доступ к данным, что не соответствует декларативному характеру объектной парадигмы моделирования реальности. Объектные языки запросов не обладают дедуктивными возможностями. Наконец, объектным моделям данных недостает формальной основы, в отличие от реляционных моделей.

Мы уже упоминали некоторые попытки создания концептуальных моделей данных, обеспечивающих более высокий уровень семантики в моделировании предметной области. Первоначально семантические концептуальные модели данных создавались как инструменты *концептуального проектирования* баз данных. Представленная средствами такой модели данных концептуальная схема предметной области, как уже отмечалось, должна быть отображена в *дatalogическую среду*, т.е. в среду той модели данных, которую поддерживает используемая СУБД. Однако модели данных, поддерживаемые традиционными коммерческими СУБД функционально

более бедны по сравнению с семантическими концептуальными моделями данных. В силу этого при отображении концептуальной схемы предметной области в *дatalogическую среду* в значительной степени утрачивается семантика предметной области.

Поэтому возник интерес к использованию семантических концептуальных моделей данных в системах баз данных *и на стадии функционирования этих систем* как основы пользовательских интерфейсов, иначе говоря, в качестве *внешних моделей данных* в системах баз данных. Стали проводиться исследования и разработки таких технологий, которые бы обеспечивали комплексное использование семантических моделей данных как инструмента концептуального моделирования предметной области, так и моделей данных внешнего уровня системы базы данных, и тем самым позволяли бы повысить семантический уровень их пользовательских интерфейсов. Семантическая концептуальная модель данных используется при этом и на стадии проектирования системы, и на стадии ее функционирования.

Созданию таких семантических моделей данных и воплощающих их языков, реализации прототипов систем управления данными и их окружения, пользовательские интерфейсы которых строились на основе таких моделей, был посвящен ряд исследовательских проектов. Ограничимся лишь несколькими примерами. Прежде всего, это проекты, в которых предусматривалась поддержка модели сущностей-связей на внешнем уровне архитектуры СУБД (см. например [31]), язык FORAL [74], проекты Taxis [54] и Telos [53, 55], реализованные в СССР проекты Института кибернетики (Киев) [6], ОМЕГА-1 [7], КОМОД [8], АТЛАНТ [2]. Упомянем также разработанный позднее язык СИНТЕЗ [3], а также СУБД, пользовательский интерфейс которой поддерживает UML-представление базы данных и компонент стандарта UML – язык ограничений Object Constraint Language (OCL) в качестве языка запросов [1]. К этому направлению технологий примыкают и дедуктивные базы данных.

В последние годы в качестве языков концептуального моделирования, особенно

в реализациях экспериментальных проектов для научных исследований, стал применяться язык RDF [69] вместе с языком запросов SPARQL [80] для доступа к RDF-спецификации предметной области. Совокупность этих языков определяет полнофункциональную (включающую как дескриптивные, так и операционные средства) семантическую модель данных. Более широкому использованию языка RDF для концептуального моделирования будет способствовать осуществляемая в консорциуме W3C разработка нового стандарта описания отображения реляционных данных в RDF [11]. В ряде исследовательских проектов в качестве языка концептуального моделирования стали использоваться также версии языка описания онтологий консорциума W3C – OWL и OWL2.

Подобные проекты обозначили сближение технологий баз данных и систем, основанных на знаниях. Новый импульс развитию этой тенденции придают результаты исследований в области дескриптивных логик, их использование как языков описания онтологий, принятие стандартов этих языков, а также результаты, полученные на этой основе в области разработки технологий доступа к данным, основанного на онтологиях.

Завершая краткий обзор эволюции концептуальных моделей данных и воплощающих их языков концептуального моделирования, следует отметить, что в процессе эволюции технологий баз данных из всего многообразия таких моделей широкое практическое применение в промышленных CASE-технологиях нашли пока лишь реляционная модель, модель сущностей-связей, а также объектная модель, воплощенная подмножеством языка UML – диаграммой классов. Последняя из них наиболее распространена в настоящее время. Однако уже существуют программные средства, позволяющие использовать в качестве языков концептуального моделирования предметной области язык RDF, а также языки описания онтологий OWL2 и др. Использование языков описания онтологий для целей концептуального моделирования предметной области в технологиях баз данных начало практиковаться в последние годы.

5. ЛОГИКА И БАЗЫ ДАННЫХ

Представляется теперь уместным обратить внимание на использование в технологиях баз данных аппарата математической логики, развитие которого привело к созданию дескриптивных логик. Логика этого класса стала основой ключевых стандартов Семантического Веба, а также ряда других информационных технологий, в частности, развивающихся технологий доступа к данным, основанного на онтологиях, которым посвящена данная статья.

Развитию использования аппарата логики в технологиях баз данных способствовало появление известных публикаций Кодда по реляционной модели данных. За прошедшие с тех пор четыре десятилетия роль логики в этой области значительно расширилась. Логика стала не только инструментом исследования реляционной модели данных и методов проектирования реляционных баз данных, но и основой важного направления в технологиях баз данных – дедуктивных и объектно-дедуктивных баз данных, а в последние годы также и основой языков описания онтологий, OBDA-систем, систем интеграции данных, ряда приложений систем баз знаний.

Создание языка Пролог (1972) и первого компилятора этого языка привели к рождению логического программирования. Программы на Прологе оперировали лишь небольшими объемами данных, хранимых в оперативной памяти. Дальнейшее развитие логического программирования вызвало необходимость интеграции этого направления с технологиями баз данных. Естественным выбором для такой интеграции стали реляционные базы данных, также основанные на аппарате логики. Совместное использование средств логического программирования и реляционных баз данных позволило сочетать формальный аппарат вывода и возможности работы с большими объемами данных.

Для целей такой интеграции были предложены идеи расширения реляционного подхода в базах данных, адекватного ее потребностям. Реализующие эти идеи системы баз данных стали называться *системами дедуктивных баз данных*. Такие системы могут рассматриваться

как реляционные системы продвинутого типа, а традиционные реляционные системы базы данных как частный случай систем дедуктивных баз данных.

В системах дедуктивных баз данных база данных состоит из литеральных предикатов (*экстенционал базы данных*) и правил (*интенционал базы данных*). Отношение в реляционных базах данных может рассматриваться как множество значений некоторого литерального предиката, а реляционные операторы языка SQL – как логические выражения. Поэтому системы реляционных баз данных действительно являются частным случаем систем дедуктивных баз данных.

Исследования дедуктивных баз данных особенно интенсивно проводились в 1970-х–1980-х годах. В эти годы был проведен ряд представительных научных конференций и рабочих семинаров, опубликован ряд фундаментальных изданий, посвященных этой проблематике. Детальному анализу развития этого направления в теории и технологиях баз данных посвящен ряд широко известных работ (см. например, [37, 38, 51, 52, 67, 68]).

В процессе исследований технологий дедуктивных баз данных выявились серьезные проблемы, ограничивающие возможности их практического использования. Это проблемы разрешимости и сложности вычислений, оптимизации запросов и обеспечения приемлемого уровня производительности, сложности распараллеливания вычислений и контроля ограничений целостности, теоретические проблемы, связанные с отрицанием и немонотонными рассуждениями и др.

Предпринимались попытки решения этих проблем. Важным шагом в теоретическом направлении явилось создание языка Datalog [37, 86], который использует многие идеи Пролога, с синтаксической точки зрения являясь его подязыком. В языке Datalog преодолены некоторые слабые стороны Пролога, и он более приспособлен к использованию в качестве языка запросов в системах дедуктивных баз данных. В отличие от Пролога, Datalog является декларативным, в программах на этом языке (запросах)

несущественен порядок следования правил, в нем предусматривается множественная обработка фактов. В правилах Datalog, в отличие от Пролога, могут использоваться в качестве термов только литералы и переменные, но не функциональные символы. Программы Datalog всегда разрешимы.

Значительный вклад в развитие теории и технологий дедуктивных баз данных был внесен рядом крупных исследовательских проектов. Наиболее известны среди них LDL в MCC (Microelectronics and Computer Technology Corp., Texas) и Nail! (Стэнфорд) [86]. Их авторами были предложены идеи, обеспечивающие существенное повышение производительности систем дедуктивных баз данных, созданы реализующие их прототипы.

Эти и многие другие исследования в области дедуктивных систем баз данных имели важные итоги. Как уже отмечалось, системы дедуктивных баз данных расширили функциональные возможности традиционных реляционных систем, позволив использовать совместно с системой базы данных средства логического вывода. Благодаря этому такие системы стали существенным шагом от систем баз данных к системам, основанным на знаниях. Спецификация запросов в форме логических программ обеспечивает в дедуктивных системах декларативность пользовательских запросов.

Такие языки логического программирования, как Пролог, довольно функционально богаты и сложны для пользователя, что затрудняет эффективную оптимизацию программ на этом языке. Поэтому не случайно большинство усилий в области дедуктивных баз данных было посвящено исследованию свойств программ Datalog и их оптимизации. Вместе с тем, ориентация на язык Datalog существенно ограничила выразительные возможности логического программирования ради снижения вычислительной сложности и достижения разрешимости программ.

Исследования, посвященные развитию Datalog, продолжают до настоящего времени. Создан ряд расширений языка, обладающих более богатыми выразительными возможностями. Например, в [21] предложено семейство таких расширений Datalog[±], применимое как

для описания онтологий, так и запросов на онтологиях и в базах данных. В расширениях этого семейства выразимы логики семейства DL-Lite и F-Logic [46], сохраняется свойственная Datalog разрешимость и обеспечивается приемлемая сложность.

Важным дальнейшим шагом в развитии технологий дедуктивных баз данных на пороге 1990-х гг. стала интеграция технологий дедуктивных и объектных баз данных. Она осуществлялась в двух разных направлениях – обогащение объектной парадигмы моделирования дедуктивными возможностями и дополнение объектных возможностей в дедуктивные системы. Первое из них развивалось значительно интенсивнее и привело к созданию нового класса систем баз данных – систем *дедуктивных объектно-ориентированных баз данных (Deductive Object-Oriented Database, DOOD)* [12, 13, 49].

В системах этого класса удалось объединить логические основы дедуктивного подхода с модельными возможностями объектной парадигмы: обеспечить поддержку концепции объекта, индивидуальности объектов, конструирования сложных объектов, типизации объектов, методов, основанных на правилах, инкапсуляцию методов, перегрузку свойств, полиморфизм, представление иерархии классов, множественное наследование и др. В таких системах поддерживается объектная схема, описывающая структурные и поведенческие аспекты базы данных. Пользователям предоставляется декларативный язык дедуктивных правил, позволяющий осуществлять вывод на множестве хранимых фактов. Ценным теоретическим результатом создания дедуктивных объектно-ориентированных баз данных является логическая формализация объектно-ориентированного подхода.

В 1990-е годы было создано большое количество языков, воплощающих модели данных для DOOD: O-logic (1986, пересмотренная версия – 1993), C-logic (1989), LOGRES (1990), LLO (1991), COMPLEX (1992), ORLOG (1992), отечественный язык СИНТЕЗ (1993), LIVING IN LATTICE (1993), Datalog-meth (1993), CORALCC (1993), Noodle (1993),

DTL (1993), F-Logic (1995), Gulog (1995), Rock & Roll (1995), ROL (1996), DatalogCC (1997), IQL (1998), ROL2 (1998, 1999), Chimera (1998), DO2 (1998) и др. Одним из наиболее известных среди них является F-Logic Кифера и Лаузена [46]. Этот язык, первоначально созданный для дедуктивных объектно-ориентированных баз данных, использовался также в ряде исследовательских проектов как средство формального описания онтологий. Однако в этой области значительно большую популярность приобрели *дескриптивные логики*, которые стали основой стандартов языка описания онтологий Веба консорциума W3C и профилей этого языка.

Технологии дедуктивных баз данных и DOOD не получили широкого практического применения, хотя некоторые их идеи воплощены в СУБД, выпускаемых индустрией программного обеспечения. Однако они сыграли важную роль в развитии теории баз данных и стали основой для дальнейшего развития применений логики в технологиях баз данных и информационных систем.

6. ДЕСКРИПТИВНЫЕ ЛОГИКИ И ОВДА-СИСТЕМЫ

Другая актуальная в настоящее время ветвь применения аппарата логики в системах баз данных и информационных системах формировалась в рамках исследований, посвященных представлению структурированных знаний и рассуждениям о них. Для этих целей различными коллективами была разработана популярная в настоящее время разновидность логик, названных в 1980-х годах *дескриптивными логиками (Description Logics, DL)* [83].

Создание дескриптивных логик – результат многолетних исследований, связанных с использованием формальной логики в информационных системах и системах баз данных. Дескриптивные логики – это семейство формализмов для структурированного представления знаний, созданное в результате попыток привнесения формальной семантики в структурированные, не основанные на логике, языки представления знаний – семантические сети и языки фреймов.

Логики этого семейства представляют собой языки представления знаний, реализующие компромисс между выразительными возможностями и вычислительной сложностью рассуждений. Они являются разрешимыми фрагментами логики первого порядка с некоторыми расширениями, обогащающими выразительную силу языков и существенно не влияющими на вычислительную сложность.

Дескриптивные логики позволяют описывать понятия (*концепты*) предметной области с помощью выражений, которые строятся из атомарных понятий (*унарных предикатов*) и атомарных ролей (*бинарных предикатов*) с использованием конструкторов понятий и ролей, предоставляемых данной логикой.

Важно при этом, что дескриптивные логики обладают формальной основанной на логике семантикой. Они обладают также важными для практических применений свойствами, достигнутыми на основе компромисса между степенью выразительности и разрешимостью.

Благодаря их свойствам дескриптивные логики стали использоваться как языки представления терминологических знаний, позволяющие осуществлять рассуждения об описанных их средствами понятиях предметной области.

Основанная на дескриптивной логике база знаний состоит из двух частей. *Терминологическая часть*, называемая TBox (от англ. *Terminological*), аналогична по своим функциям схеме базы данных. Она описывает понятия (*классы*) предметной области и их роли (*бинарные отношения между ними*).

Вторая часть базы знаний, *часть утверждений*, называемая ABox (от англ. *Assertional*), содержит утверждения, описывающие конкретную ситуацию в предметной области, устанавливая свойства ее индивидов. Эта часть аналогична по своим функциям наполнению базы данных.

Наиболее привлекательной областью применения дескриптивных логик стало их использование в качестве формальных языков описания онтологий для семантических информационных систем. В 1980-е–начале 1990-х годов был разработан и реализован целый ряд дескриптивных логик: KL-

ONE (1985), KRYPTON (1983), LOOM (1987), BACK (1988), K-REP (1991), CLASSIC (1991), KRIS (1991). Начиная с середины 1990-х годов, стали создаваться логики, обеспечивающие большую выразительность и при этом вычислительную сложность, не превышающую полиномиальной. Они позволили реализовать механизмы рассуждений (резонеры) для сложных онтологий, обладающие приемлемой производительностью. К их числу можно отнести: FaCT (1998), RACER (2001), CEL (2005), KAON 2 (2005).

Появление Веб в середине 1990-х годов вызвало потребности в языке описания онтологий для этой информационной среды. Деятельность в этом направлении особенно активизировалась после провозглашения концепции Семантического Веба. В этом направлении наиболее значимым результатом стало создание языков описания онтологий OIL (Ontology Interchange Language, проект Европейской комиссии) и DAML (DARPA Agent Markup Language, проект DARPA), гибрид которых DAML+OIL стал прототипом принятого консорциумом W3C стандарта языка описания онтологий OWL (2004). Диалект DL языка OWL основан на дескриптивной логике.

В последние годы дескриптивные логики рассматриваются как формализм не только для описания онтологий, но и для создания систем семантической интеграции данных, систем баз данных с пользовательским интерфейсом, который обеспечивает доступ к данным, основанный на онтологиях, и рассуждения на онтологиях (OBDA-систем), а также для ряда других приложений.

Полученные к настоящему времени основные результаты в области OBDA-систем базируются на дескриптивных логиках семейства DL-Lite [22, 23, 27]. Это семейство разработано на основе обеспечивающего разрешимость и вычислимость запросов диалекта OWL DL языка OWL.

В конце 2009 года консорциум W3C одобрил продвинутую версию стандарта языка описания онтологий – OWL2 [57, 58]. В нем появились полезные новые конструкции, обогащающие выразительные возможности, но не нарушающие, тем не менее, разрешимость

и вычислимость в тех рамках, в которых они обеспечивались прежней версией языка. Как отмечалось ранее, для OWL2 разработаны его профили OWL2 EL, OWL2 QL и OWL2 RL [59], также одобренные консорциумом. Эти подязыки OWL2 за счет ограничений его выразительной силы обеспечивают определенные преимущества в аспекте реализации в некоторых конкретных областях применения по сравнению с полным языком.

В контексте данной статьи интерес представляет профиль OWL2 QL, ориентированный на приложения, которые имеют дело с большим объемом экземпляров данных (иначе говоря, с большим ABox) и в которых важной задачей являются эффективные рассуждения на онтологиях. OBDA-системы относятся именно к этой категории приложений.

Прототипом профиля OWL2 QL послужила дескриптивная логика DL-Lite_R семейства DL-Lite. В спецификации профиля OWL2 QL [59] утверждается, что другие более выразительные логики семейства DL-Lite могут поддерживаться над OWL2 QL, но могут потребоваться дополнительные ограничения на структуру онтологий, представляемых их средствами.

Целью разработки семейства логик DL-Lite являлось обеспечение приемлемой для реальных практических приложений вычислительной сложности рассуждений (*reasoning*) на TBox и обработки запросов на ABox большого объема, сводимых к конъюнктивным запросам или их объединениям, которые могли бы отображаться в среду традиционной SQL-системы базы данных и обрабатываться используемой в ней СУБД. Такие возможности были достигнуты благодаря компромиссу между выразительной силой логик этого семейства, с одной стороны, и вычислительной сложностью рассуждений на онтологии и обработки запросов данных – с другой.

Логики семейства DL-Lite обеспечивают полиномиальную сложность рассуждений относительно размера TBox и сложность LogSpace относительно размера ABox (*сложность по данным*). Важным их достоинством является возможность осуществлять рассуждения на TBox независимо от ABox и выполнения обработки запросов на ABox

независимо от TBox. При этом обработка запросов на ABox сводится к выполнению, в конечном счете, запросов средствами реляционной SQL-СУБД. Благодаря этому может осуществляться оптимизация запросов с помощью механизмов, которыми располагают для этих целей традиционные SQL-СУБД.

Авторам семейства DL-Lite удалось показать [28], что логика DL-Lite_R этого семейства представляет собой максимальный фрагмент языка OWL DL, обладающий указанными свойствами. При использовании ее в качестве прототипа в профиле OWL2 QL для согласованности со стандартом OWL2 внесены некоторые изменения. Исключена поддержка предположения об уникальности имен (*Unique Name Assumption*), поскольку оно не поддерживается в OWL2. В профиле не выражается утверждение идентификации (*Identification Assertion*), характеризующее функциональность ролей и атрибутов, хотя оно выразимо в логике-прототипе. Такое решение аргументируется необходимостью обеспечить для профиля хорошие характеристики сложности рассуждений. Вместе с тем, дополнены не влияющие на вычислительную сложность утверждения свойств ролей, поддерживается типизация данных языка OWL2, которой также нет в логике-прототипе.

Следует отметить, что для логик семейства DL-Lite при некоторых расширениях их выразительных возможностей сохраняются указанные оценки сложности. Примером может служить логика DL-Lite_{A,id}, детально рассмотренная в [23]. В этой работе на основе указанной логики полностью воспроизведена теория построения OBDA-систем, которая обсуждалась авторами семейства в их ранних работах на основе более ограниченных логик семейства.

В объемной статье [23], содержащей комплексное обсуждение всех полученных ими результатов по OBDA-системам, авторы показывают на примере DL-Lite_{A,id}, являющейся некоторым расширением логики DL-Lite_R, что эта логика позволяет выразить основные модельные возможности диаграммы классов UML, за исключением методов, ассоциированных с классами. На этом основании

утверждается, что логики семейства обладают достаточно развитыми возможностями концептуального моделирования. Поэтому в исследованиях OBDA-систем онтологии, представленные средствами указанных логик, предложено использовать в качестве концептуальных схем предметной области, на которых базируется пользовательский интерфейс – надстройка над SQL-системой базы данных.

При сопоставлении возможностей концептуального моделирования логик семейства DL-Lite и языка UML вопрос о моделировании поведенческих аспектов предметной области является довольно важным. Моделирование поведения является неотъемлемым элементом объектной парадигмы моделирования. Для этой цели используются методы, определяемые для классов. Авторы [23] утверждают, что такая возможность диаграммы классов, как методы, нужна, скорее, для программной инженерии, чем для концептуального моделирования, и поэтому при оценке логик DL-Lite возможность выражать методы их средствами можно не учитывать. Этот вопрос возник в связи с тем, что определяемые для классов UML методы не выразимы в логиках семейства DL-Lite. Они не выразимы также в языке OWL2 и его профиле OWL2 QL.

Нужно заметить, однако, что приведенный аргумент авторов цитируемой работы представляется неправомерным. Он, по сути, противопоставляет модельные потребности сферы концептуального моделирования и программной инженерии. Однако концептуальное моделирование является необходимой составной частью программной инженерии. Это подчеркивалось еще на симпозиуме [17], в материалах которого констатировалось, что концептуальное моделирование – это сфера общих интересов для области искусственного интеллекта, программирования и технологий баз данных.

В заключение этого раздела нужно отметить, что поддержка представленных средствами профиля OWL2 QL онтологий на интерфейсе для SQL-СУБД допускает эффективную обработку запросов данных, выразимых как конъюнктивные запросы или их объединения, в приложениях с большим

объемом баз данных. Однако ради достижения приемлемых оценок сложности в этом профиле не допускается использование ряда полезных для доступа к базам данных конструкций, которые предусмотрены в языке OWL2. К их числу относятся, в частности, использование спецификации ключей для классов, ограничений кардинальности, спецификации перечислимых множеств значений для индивидов и литералов, некоторых форм кванторов существования и всеобщности и др. Именно по этой причине ограничивается допустимый вид запросов данных. Выразительная сила языка SQL при этом используется в весьма ограниченной степени.

7. ПРИНЦИПЫ ОРГАНИЗАЦИИ И ФУНКЦИОНИРОВАНИЯ OBDA-СИСТЕМ

С формальной точки зрения, OBDA-систему можно рассматривать как триплет следующего вида: $O = \langle T; M; D \rangle$, где: T – это $TBox$ используемой онтологии, D – это ее $ABox$, представляемый SQL-базой данных, M – множество утверждений отображения между T и D .

Поскольку в качестве источника данных в OBDA-системах предполагается использовать реляционные SQL-базы данных, система должна допускать только запросы данных, переписываемые в логику первого порядка и тем самым отображимые в язык SQL. В случае неполноты данных в базе данных SQL-запросы становятся неразрешимыми. Компромиссный вариант выразительной силы языка представления запросов данных в OBDA-системе заключается в отображимости его во фрагмент языка SQL, позволяющий формулировать конъюнктивные запросы и их объединения (*Union of Conjunctive Queries, UCQs*).

Источником данных в OBDA-системе является $ABox$ онтологии, контент которого составляют абстрактные объекты предметной области, возвращаемые пользователю вместе с их свойствами в ответах на запросы. Контент $ABox$ соответствует $TBox$ онтологии в том смысле, что представленные в нем концепты, роли и атрибуты определены в $TBox$.

Предполагается также, что источник данных имеет значительный объем и хранится как база

данных, управляемая реляционной SQL-СУБД. Контент ABox в системе не материализован, он является, таким образом, *виртуальным* – необходимые для обработки запросов его элементы порождаются системой базы данных на стадии исполнения.

Архитектура OBDA-системы вполне согласуется со сложившимися представлениями о многоуровневой архитектуре систем баз данных [16], обеспечивающей реализацию принципа *независимости данных*. В данном случае предусматриваются следующие архитектурные компоненты:

- пользовательский интерфейс, который поддерживает онтологии как концептуальные схемы предметной области и одновременно внешние схемы базы данных (в терминах ANSI/X3/SPARC), обеспечивает функции резонера и предоставляет средства для спецификации запросов данных;
- механизм отображения запросов данных в среду реляционной SQL-системы базы данных на основе определенного на стадии разработки *связывания онтологии и системы базы данных*;
- традиционная SQL-система базы данных, управляемая какой-либо из распространенных СУБД.

Такая архитектура обеспечивает для пользователя инкапсуляцию реального репозитория данных – SQL-системы базы данных, существование которой в “недрах” OBDA-системы становится для него прозрачным, и он не обязан о ней знать.

Отображение запросов, сформулированных в терминах онтологии, в среду системы базы данных формируется, как уже указывалось, на основе отображения между онтологией и SQL-схемой базы данных, предварительно определенного проектировщиком конкретной OBDA-системы. Описание этого отображения состоит из совокупности *утверждений отображения* двух видов: *типизированных утверждений отображения* и *утверждений отображения “данные-объекты”*. Типизированные утверждения отображения определяют соответствие типов онтологии элементам схемы базы данных. Утверждения

отображения “данные-объекты” описывают отображение данных базы данных в экземпляры концептов, ролей и атрибутов онтологии.

Сложность связывания онтологии с системой базы данных состоит в том, что онтологии основаны на *гипотезе открытого мира*, а источник данных (система базы данных) – на *гипотезе замкнутого мира*. В базе данных могут отсутствовать данные, которые запрашиваются пользователем в терминах онтологии. При этом в традиционных технологиях баз данных факт, не представленный в базе данных, считается ложным, в отличие от интерпретации такой ситуации в среде онтологии, где для такого факта рассматриваются три альтернативы – он может быть истинным, ложным или неизвестным.

Другая возникающая здесь проблема – *несоответствие импеданса*. Эта проблема в технологиях баз данных впервые проявилась при создании интерфейсов прикладного программирования реляционных СУБД для традиционных языков программирования. В реляционных базах данных операции над строками таблиц имеют множественный характер. В то же время операции над данными в традиционных языках программирования не являются множественными. В OBDA-системах возникновение проблемы несоответствия импеданса обусловлено тем, что в источнике данных хранятся структурированные данные различных типов – числа, строки, булевские значения и т.п., а запросы данных формулируются в терминах объектов, представляющих в онтологии концепты, их роли и атрибуты, а также отношения между ними. Поэтому механизмы отображения должны конструировать из значений данных в источнике идентификаторы тех абстрактных объектов, которые составляют ABox в онтологии. Для решения этой проблемы авторы концепции OBDA-систем используют подход, заимствованный в работах по дедуктивным объектно-ориентированным базам данных [41]. Принципы и техника отображения, а также предложенный для OBDA-систем язык описания отображений подробно описаны в [23, 54].

Рассмотрим теперь, как обрабатываются

запросы данных в OBDA-системах. Этот процесс осуществляется в три этапа. На первом этапе запрос над TBox переписывается в запрос в FOL, который может выполняться над ABox, материализованным в реляционной системе базы данных. Можно сказать, что задача формирования ответа на запрос (*Query Answering* – термин систем баз знаний) сводится при этом к вычислению запроса (*Query Evaluation* – термин систем баз данных). Важно заметить, что TBox при этом не используется. Для логики DL-Lite_R предложенный ее авторами алгоритм переписывания запроса представлен в [28]. Более эффективный алгоритм переписывания рассматривается в работе [61].

На втором этапе осуществляется отображение полученного FOL-представления запроса в среду реляционной системы базы данных. Это отображение сводится к препроцессорному переписыванию запроса в язык SQL. При этом могут быть получены только конъюнктивные запросы (*Conjunctive Queries, CQ*) или их объединения (*Union of Conjunctive Queries, UCQs*). Полученный SQL-запрос может включать только конструкции Select-Project-Join и, возможно, Union.

Наконец, на третьем этапе осуществляется собственно обработка запроса в материализованном ABox – в SQL-системе базы данных. Значения данных, полученные в результате выполнения запроса, как указывалось выше, трансформируются далее в абстрактные объекты, составляющие ABox онтологии, и возвращаются пользователю.

Получены оценки вычислительной сложности функционирования рассмотренного здесь класса OBDA-систем. В случае использования логик DL-Lite для представления онтологии известны следующие оценки: для рассуждений на TBox – полиномиальная сложность относительно размера TBox, для обработки запросов данных на ABox (*сложность по данным*) – LogSpace относительно объема данных. Оценка для запросов на ABox следует из того, что они сводятся в OBDA-системах, имеющих рассмотренную архитектуру, к обработке SQL-запросов в реляционной системе базы данных. Эти оценки остаются в силе и для систем, использующих профиль QWL2 QL.

8. ИНСТРУМЕНТАРИЙ ДЛЯ РЕАЛИЗАЦИИ OBDA-СИСТЕМ

Для создания OBDA-систем имеется ряд инструментальных средств, созданных авторами семейства дескриптивных логик DL-Lite, а также другими исследовательскими группами. Некоторые из них имеют более широкую сферу применения. Кратко рассмотрим некоторые из этих инструментов.

Система QuOnto (QUerying ONTOlogies) [15, 66] – резонер для дескриптивных логик семейства DL-Lite. Он обладает также средствами обработки запросов данных при большом объеме A-Box. Обеспечивает совместимость с OWL2 и SPARQL. Механизм рассуждений в системе хорошо оптимизирован. Запросы данных могут быть адресованы к “внутренним” данным, которые поддерживаются в собственном репозитории QuOnto, или к данным, содержащимся в SQL-системе базы данных. Система QuOnto располагает драйверами для ряда широко распространенных СУБД: Oracle, DB2, SQL Server, MySQL, Postgres и др., позволяя тем самым представить A-Box в виде обычной SQL-базы данных. QuOnto реализована на языке Java, имеет интерфейсы прикладного программирования, в том числе, расширенные DIG-интерфейсы.

DIG-интерфейс [35] – созданная коллективом исследователей дескриптивных логик, называемым DL Implementation Group (DIG), широко принятая спецификация стандарта интерфейса для взаимодействия в сетевой среде различных приложений с резонерами на дескриптивных логиках. Спецификация DIG определяет основанный на дескриптивной логике SHOIQD-н язык описания онтологий (*concept language*), а также допускающий расширения минимальный набор сообщений, которыми обмениваются резонеры, снабженные DIG-интерфейсами (их называют *DIG-резонерами*), и их клиенты. По существу, это – протокол доступа клиентов к DIG-резонерам. Большинство известных редакторов онтологий и резонеров поддерживают этот интерфейс. Последняя разработанная его версия – DIG 2.0 [85].

OBDA-расширение для DIG [34] – первоначально разработано для версии DIG 1.1 и

получило развитие в последующих версиях DIG-интерфейса. Дополняет функциональность DIG-интерфейса возможностями спецификации параметров источников данных и отображения запросов на онтологии в запросы к источникам данных для резонеров OBDA-систем.

DIG-QuOnto сервер [23, 66] – реализация расширения интерфейса DIG 1.1 для системы QuOnto, которое позволяет выполнять рассуждения с помощью ее резонера на онтологиях и осуществляет переписывание запросов данных в объединение конъюнктивных SQL-запросов. Для того чтобы использовать также имеющиеся в QuOnto развитые возможности отображения онтологий в среду систем баз данных, управляемых широко распространенными SQL-СУБД, клиенты сервера должны использовать OBDA-расширение DIG.

OBDA Plugin for Protégé [23, 70] – плагин с открытым кодом для популярного редактора онтологий Protégé. Расширяет его возможности и превращает Protégé в инструмент для OBDA-систем. При этом обеспечиваются не только проектирование онтологий с отображением их в источники SQL-данных, но и возможности спецификации и выполнения рассуждений на онтологиях (в текущей версии редактора специфицированных на OWL2), а также объединений конъюнктивных SQL-запросов данных путем взаимодействия с резонерами, разработанными для архитектуры OBDA-систем. Расширенный таким образом Protégé становится развитым инструментом разработки OBDA-систем и может использоваться как клиент, например, для DIG-QuOnto.

ROWLkit [66] – первая реализация профиля OWL2 QL. Представляет собой систему с графическим пользовательским интерфейсом, обеспечивающую рассуждения над онтологиями, описанными средствами OWL2 QL, и их верификацию, а также обработку запросов данных в терминах онтологий. Использует сервисы QuOnto, дополненные средствами, необходимыми для оперирования онтологиями OWL2 QL. Может воспринимать онтологии OWL2 QL как входные данные благодаря наличию интерфейса OWL API. Система ROWLkit реализована в Java. Для хранения ABox в ней используется in-Memory

СУБД H2 Java. Но ROWLkit может оперировать также данными, хранимыми во внешней памяти.

QToolKit [66] – графический интерфейс для системы QuOnto. Он позволяет представлять онтологии DL-Lite и осуществлять рассуждения на них средствами резонера QuOnto, в полной мере используя его возможности. Для хранения ABox предусматривается внутренний репозиторий данных QuOnto.

MASTRO [64, 72] – реализованный на Java инструмент для разработки OBDA-систем. Является функциональным расширением системы QuOnto. Онтологии в MASTRO специфицируются средствами логики DL-Lite_{A,id} семейства DL-Lite. Предусмотрены возможности спецификации онтологий, их верификации (проверки непротиворечивости), выполнения запросов данных и рассуждений на онтологиях. Разработана версия с DIG-интерфейсом, называемая *DIG-MASTRO* [63]. Эта версия совместно с *OBDA Plugin for Protégé* позволяет обеспечить полный цикл разработки и функционирования OBDA-систем.

Средствами MASTRO была реализована крупная банковская система. В работе [72] содержится детальный анализ опыта этой разработки, который позитивно оценивается авторами. В этом проекте онтология включает: 79 концептов, 33 роли, 37 атрибутов концептов, около 600 аксиом и примерно 50 ограничений. Разработка этой онтологии была довольно трудоемкой. Некоторые сложности вызвало описание отображения онтологии в среду системы базы данных. Этот процесс требует, естественно, основательного знания источника данных и построенной онтологии. На практике реализованная система проявила достаточно высокую производительность. Системный инструментарий позволил проверить качество данных. Были обнаружены неполнота данных и некоторые содержащиеся в них противоречия.

Таким образом, в распоряжении разработчиков OBDA-систем уже имеется различный инструментарий, в том числе и адаптированный к стандартам W3C.

9. OBDA-СИСТЕМЫ: ДОСТОИНСТВА И ОГРАНИЧЕННОСТЬ ПОДХОДА

Обсуждаемый в данной работе подход

к созданию OBDA-систем обладает рядом достоинств. Вместе с тем, ему свойственна и некоторая ограниченность. В некоторых своих возможностях он близок к другим подходам в рассматриваемой области.

Дескриптивные логики, используемые в OBDA-системах в качестве языков концептуального моделирования предметной области, родились в результате синтеза выразительных средств логики первого порядка и структурных языков представления знаний (семантические сети, фреймы). Ранее создание дедуктивных объектно-ориентированных баз данных, а также ряда мощных формальных языков концептуального моделирования и языков описания онтологий, таких как Синтез [3], F-Logic [46] и ряда других упоминавшихся выше, стало результатом этого же процесса.

Создатели семейства дескриптивных логик DL-Lite, предложенного в качестве основы таких систем, главной своей задачей считали не столько развитие их выразительной силы, сколько нахождение приемлемого для практических приложений компромисса между их выразительной силой и вычислительной сложностью описываемых их средствами рассуждений в терминах онтологий, а также запросов данных на ABox. Компромисса удалось достигнуть за счет ограничения выразительности логик этого семейства. Так, средствами этих логик не выразимы, например, рекурсивные запросы, которые допускаются в дедуктивных базах данных, а теперь выразимы и средствами действующей версии стандарта языка SQL. На ABox в логиках семейства DL-Lite допустимы только запросы, сводимые к дизъюнктивным запросам в SQL-базе данных или к их объединениям. Выразительная сила языка SQL в системах баз данных, над которыми надстраивается такой онтологический уровень представления концептуальной схемы, используется при этом в весьма ограниченной степени. Тем не менее, выразительная сила логик семейства DL-Lite остается достаточной для многих практических приложений.

Как и системы дедуктивных баз данных, OBDA-системы позволяют осуществлять рассуждения на интенционале (TBox) и за-

просы данных на экстенционале базы знаний (ABox), представляемой онтологией. К числу их достоинств следует отнести высокий уровень абстрактности представления данных в базе данных, с которым имеют дело пользователи системы, развитые выразительные средства запросов данных, их декларативность, приемлемую производительность обработки запросов и рассуждений на онтологии.

Важным достоинством подхода авторов рассмотренной концепции OBDA-систем является их архитектура. Как уже указывалось, OBDA-система состоит из традиционной SQL-системы базы данных с надстройкой, обеспечивающей поддержку онтологии, которая играет роль концептуальной схемы, рассуждения на TBox и спецификацию запросов данных на ABox, а также отображение этих запросов в среду SQL-системы базы данных. Отображение между заданной онтологией и конкретной SQL-схемой базы данных явно описывается, а механизмы СУБД обычным образом обеспечивают оптимизацию и обработку запросов.

Такая архитектура обеспечивает возможность автономной и независимой разработки вновь создаваемой SQL-системы базы данных по отношению к созданию онтологии. При создании крупных систем это позволяет осуществить разделение труда и первоначальное распараллеливание разработки системы базы данных и онтологии. Созданная онтология на следующих этапах "связывается" с системой базы данных. При этом проверяется качество данных в базе данных – обнаруживаются возможные неожиданная неполнота данных (относительно онтологии) и их противоречивость. На стадии исполнения могут быть использованы в запросах скрытые, явно не специфицированные в схеме базы данных, отношения между данными, определенные или продуцируемые из онтологии.

Возможность использования ранее и независимо разработанной SQL-системы базы данных как составной части создаваемой OBDA-системы позволяет обеспечить этой системе базы данных новую жизнь в среде, предоставляющей пользователю семантический интерфейс доступа.

Оценивая достоинства созданных технологий OBDA-систем, следует, наконец, отметить еще одно важное для их практического распространения обстоятельство. Благодаря принятию консорциумом W3C нового стандарта языка описания онтологий Веба – OWL2 и его профилей, системы рассматриваемого класса могут теперь базироваться на стандартизированной основе: профиле OWL2 QL как языке описания онтологий и языке баз данных SQL. Обеспечивается также интероперабельность OBDA-систем с приложениями Семантического Веба. В такой ситуации подход авторов рассмотренной здесь концепции OBDA-системы, фактически базирующийся на уже существующем стандарте языка описания онтологий, имеет значительно больше шансов на широкое практическое использование, чем другие подходы в этой области, например, основанные на Datalog[±] – упоминавшемся семействе расширений языка Datalog.

10. ЗАКЛЮЧЕНИЕ

Завершая данный обзор, можно констатировать, что проведены интенсивные исследования и создана теоретическая основа OBDA-систем, обладающих практически приемлемыми характеристиками вычислительной сложности рассуждений на онтологиях и обработки запросов данных на A-Vox большого объема при ограниченной, но достаточной для многих практических приложений выразительной силе языка описания онтологий.

Предложенные языки описания онтологий позволяют выразить с некоторыми исключениями основные элементы языков концептуального моделирования, реализованных в коммерческих программных продуктах (диаграмм классов UML, ER-диаграмм). Благодаря этому они могут практически использоваться для концептуального моделирования предметной области при разработке многих приложений и для использования онтологий в качестве концептуальных схем, на которых базируются семантические пользовательские интерфейсы создаваемых OBDA-систем.

Разработаны исследовательские прототипы ряда инструментальных средств для создания OBDA-систем, а также успешно функционирующие действующие системы, основанные

на дескриптивных логиках семейства DL-Lite. Некоторые из этих инструментов основаны на профиле OWL2 QL – стандарте консорциума W3C, созданном на основе одной из логик этого семейства.

Таким образом, возникли предпосылки для широкого практического применения технологий OBDA-систем. Семантические возможности концептуальных схем, обеспечиваемые рассмотренным подходом, в некоторой мере ограничены. Тем не менее, можно с большой степенью уверенности предположить, что в близкое время индустрия программного обеспечения начнет выпуск тиражируемых инструментальных программных средств для разработки OBDA-систем.

СПИСОК ЛИТЕРАТУРЫ

1. *Гринева М.Н., Кузнецов Д.* UQL: язык запросов к интегрированным данным в терминах UML // Программирование. 2002. № 4. С. 9–19.
2. *Замулин А.В.* Язык программирования Атлант: (Предварит. сообщ.). Новосибирск: ВЦ СО АН СССР, 1986. 46 с.
3. *Калиниченко Л.А.* СИНТЕЗ: язык определения, проектирования и программирования интероперабельных сред неоднородных информационных ресурсов (вторая редакция). Институт проблем информатики Российской академии наук. Сентябрь 1993.
4. *Когаловский М.Р., Калиниченко Л.А.* Концептуальное моделирование в технологиях баз данных и онтологические модели. Труды симпозиума “Онтологическое моделирование: состояние и направления исследований и применения”. Звенигород. 21–22 мая 2008. М.: ИПИ РАН, 2008.
5. *Когаловский М.Р., Калиниченко Л.А.* Концептуальное и онтологическое моделирование в информационных системах // Программирование. МАИК “Наука”/Интерпериодика. 2009. № 5.
6. *Михновский С.Д.* Технология концептуального проектирования интегрированных баз данных. Тез. докл. IV Всесоюзн. конф. “Системы баз данных и знаний”. Секция 3 “Применение СУБД и СУБЗ”. Калинин, 1989.
7. *Савинков В.М., Вейнеров О.М., Казаров М.С., Жадан Н.В., Блюм В.М.* Основные принципы организации экспериментальной системы автоматизации проектирования баз данных

- ОМЕГА-1. Тезисы докл. Второй Всесоюз. конф. “Банки данных”. Секция 3 “Методы и системы проектирования баз данных”. Ташкент, 1983.
8. *Фурсин Г.И.* Теория и практика создания банков данных. Киев: Головное издательство издательского объединения “Вища школа”, 1987.
 9. *Цаленко М.Ш.* Моделирование семантики в базах данных. М.: Наука. Гл. ред. физ.-мат. лит., 1989.
 10. *Цикритзис Д., Лоховски Ф.* Модели данных: Пер. с англ. М.: Финансы и статистика, 1985.
 11. A Direct Mapping of Relational Data to RDF. W3C Working Draft 24 March 2011. <http://www.w3.org/TR/rdb-direct-mapping/>
 12. *Abiteboul S.* Deductive and Object-Oriented Databases. Logic Programming, Proceedings of the Joint International Conference and Symposium on Logic Programming, (JICSLP 1992). November 1992. MIT Press 1992. P. 26–29.
 13. *Abiteboul S.* Towards a deductive object-oriented database language. Data & Knowledge Engineering. V. 5. I. 4. October 1990. P. 263–287.
 14. *Abrial J.R.* Data Semantics. In: J. Klimbe and K. Koffeman (eds.). Database Management, North-Holland, 1974.
 15. *Acciarri A., Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Palmieri M., Rosati R.* QUONTO: QUerying ONTOlogies. Proc. of the 20th National Conf. on Artificial Intelligence. V. 4. Pittsburgh, Pennsylvania, 2005. P. 1670–1671.
 16. ANSI/X3/SPARC Study Group on Data Base Management Systems. Interim Report. SIGFDT Bull, ACM. V. 7. № 2. (1975).
 17. On Conceptual Modelling, Perspectives from Artificial Intelligence, Databases, and Programming Languages / Eds by M.L. Brodie, J. Mylopoulos, J.W. Schmidt. Springer-Verlag, 1984.
 18. *Bubenko J.* Data Models and their Semantics. In: Data Design. Infotech State of the Report Series. V. 8. № 4. 1980.
 19. *Bubenko J.A. Jr.* IAM: An Inferential Abstract Modeling Approach to Design of Conceptual Schema. Proceedings of the ACM SIGMOD Intern. Conf., Toronto, Canada, August 3–5, 1977.
 20. Business Process Model and Notation (BPMN). Version 2.0. OMG Document Number: formal/2011-01-04. <http://www.omg.org/spec/BPMN/2.0>
 21. *Cali A., Gottlob G., Lukasiewicz T.* Datalog Extensions for Tractable Query Answering over Ontologies. In: Roberto De Virgilio, Fausto Giunchiglia, Letizia Tanca (Eds.). Semantic Web Information management. A Model-Based Perspective. Springer-Verlag, 2010.
 22. *Calvanese D.* Ontologies and Databases. Tutorial. Reasoning Web Summer School 2009. September 3–4, 2009. Bressanone, Italy. <http://www.inf.unibz.it/calvanese/teaching/2009-09-ReasoningWeb-school-ontologies-dbs/ReasoningWeb-2009-ontologies-dbs.pdf>
 23. *Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Poggi A., Rodriguez-Muro A., Rosati A.* Ontologies and Databases: The DL-Lite Approach. Semantic Technologies for Informations Systems. 5th Int. Reasoning Web Summer School (RW 2009). LNCS, V. 5689, 2009.
 24. *Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Poggi A., Rodriguez-Muro M., Rosati R., Ruzzi M., Savo D.F.* The MASTRO system for ontology-based data access. Semantic Web 2 (2011) 43–53. <http://iospress.metapress.com/content/4332331681716631/fulltext.pdf>
 25. *Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Poggi A., Rosati R.* Ontology-Based Database Access. Proc. of the 15th Italian Conf. on Database Systems (SEBD 2007), 2007.
 26. *Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Rosati R.* Conceptual Modeling for Data Integration. <http://www.inf.unibz.it/calvanese/papers/calv-et-al-book-mylopoulos-2009.pdf>
 27. *Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Rosati R.* DL-Lite: Tractable Description Logics for Ontologies. In Proc. of AAAI 2005. P. 602–607. 2005.
 28. *Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Rosati R.* Tractable reasoning and efficient query answering in description logics: The DL-Lite family. JAR, 39(3):385–429, 2007.
 29. *Calvanese D., Lembo D.* Ontology-Based Data Access. Tutorial. 6th Int. Semantic Web Conference (ISWC 2007), Busan, South Korea. Nov. 12th, 2007.
 30. *Chen P.P.* The Entity-Relationship Model. Toward a Unified View of Data. ACM TODS, 1(1): 1976. Русск. пер.: Питер Пин-Шен Чен. Модель “сущность-связь” – шаг к единому представлению данных. СУБД, 3/1995.
 31. *Chiang T.C., Bergeron F. R.* A Data Base Management System with an E-R Conceptual Model. In: Peter P. Chen (Ed.): Entity-Relationship Approach

- to Systems Analysis and Design. Proc. 1st International Conference on the Entity-Relationship Approach. North-Holland 1980. P. 467–476.
32. *Codd E.F.* Extending the Database Relational Model to Capture More Meaning. ACM TODS, V. 4. № 4. December 1979. Русск. пер.: Кодд Э.Ф. Расширение реляционной модели для лучшего отражения семантики. СУБД, 4–5/1996.
 33. *DeMarco T.* Structured Analysis and Systems Specification. Englewood Cliffs, NJ: Prentice-Hall, 1979.
 34. DIG 1.1 – OBDA. <http://obda.inf.unibz.it/dig-11-obda/>
 35. DL Implementation Group. <http://dl.kr.org/dig/>
 36. ER Web Site. <http://www.er.byu.edu/>
 37. Logic and Databases / Eds by Gallaire H., Minker J. Plenum Press, New York, Apr. 1978.
 38. *Gallaire H., Minker J., Nicolas J.-M.* Logic and databases: A deductive approach. ACM Computing Surveys, 16(2):153–185, June 1984.
 39. *Halpin T.* Object-Role Modeling (ORM/NIAM) In: Handbook on Architectures of Information Systems / Eds by P. Bernus, K. Mertins, G. Schmidt. Springer-Verlag, 1998. <http://www.orm.net/pdf/springer.pdf>
 40. *Hammer M., McLeod D.* Database description with SDM: a semantic database model, ACM TODS. V. 6. № 3. September 1981.
 41. *Hull, R.* A survey of theoretical research on typed complex database objects. In: Paredaens, J. (ed.) Databases. P. 193–256. Academic Press, London (1988).
 42. IEEE Standard 1320.2–1998 for Conceptual Modeling Language – Syntax and Semantics for IDEF1X97 (IDEFobject)
 43. ISO/IEC 19501:2005 Information technology – Open Distributed Processing – Unified Modeling Language (UML) 1.4.2 (2005).
 44. *Kaschek R.* Introduction. Issues of a Conceptual Modeling Agenda. In [45].
 45. The Evolution of Conceptual Modeling / Eds by R. Kaschek, L. Delcambre. LNCS 6520, Springer-Verlag, 2011.
 46. *Kifer M., Lausen G.* F-Logic: A Higher-Order Language for Reasoning about Objects, Inheritance, and Schema. SIGMOD Record. V. 18. № 2. 1989.
 47. *Langefors B.* Infological model and information user views. Information Systems. № 5. 1980.
 48. *Langefors B.* Information systems. In: Information Processing-74, IFIP Congress-74. Amsterdam: North-Holland. 1974.
 49. *Liu M., Dobbie G., Tok Wang Ling.* A Logical Foundation for Deductive Object-Oriented Databases. National University of Singapore. ACM Transactions on Database Systems, V. 27. № 1. March 2002. P. 117–151
 50. Abstraction in databases / Eds by McLeod D., Smith J.M. Proceedings of the 1980 Workshop on Data abstraction, databases and conceptual modeling, June 23–26, 1980, Pingree Park, Colorado, US. SIGMOD Record. V. 11. № 2. February 1981.
 51. *Minker J.* Logic and databases: a 20 Year Retrospective. Proc. of the Intern. Workshop on Logic in Databases. Lecture Notes In Computer Science; V. 1154. Springer, 1996. <http://prism.cs.umd.edu/papers/Retrospective/Retrospective.ps.Z>
 52. *Minker J.* Perspectives in deductive databases. J. of Logic Programming, 5:33–60, 1988.
 53. *Mylopoulos J.* Conceptual Modelling and Telos. Eds by P. Loucopoulos, R. Zicari. Conceptual Modelling, Databases and CASE: An Integrated View of Information Systems Development, Wiley, 1992. <http://www.cs.toronto.edu/jm/2507S/Readings/CM+Telos.pdf>
 54. *Mylopoulos J., Borgida A., Greenspan S., Wong H.K.T.* Information System Design at the Conceptual Level – the TAXIS Project. IEEE Database Engineering Bulletin. V. 7. № 4. December 1984.
 55. *Mylopoulos J., Borgida A., Jarke M., Koubarakis M.* Telos: Representing Knowledge About Information Systems. ACM Transactions on Information Systems, October 1990.
 56. *Olive A.* Conceptual modeling of information systems. Springer-Verlag, 2007.
 57. OWL 2 Web Ontology Language Document Overview. W3C Recommendation 27 October 2009.
 58. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C Recommendation 27 October 2009.
 59. OWL 2 Web Ontology Language Profiles. W3C Recommendation 27 October 2009.
 60. *Peckham J., Maryanski F.* Semantic Data Models. ACM Computing Surveys. V. 20. № 3. September 1988.
 61. *Prez-Urbina H., Motik B., Horrocks I.* Efficient Query Answering for OWL 2. The Semantic Web – ISWC 2009. LNCS. 2009. V. 5823/2009. P. 489–504
 62. *Poggi A., Lembo D., Calvanese D., De Giacomo D., Lenzerini D., Rosati R.* Linking Data to Ontologies. JODS–2008

63. *Poggi A., Rodriguez-Muro M., Ruzzi M.* Ontology-based database access with DIG-Mastro and the OBDA Plugin for Protege (Demo). In Proceeding of the Workshop OWLED 2008. Washington DC, USA, 1–2 April 2008.
64. *Poggi A., Ruzzi A.* Ontology-based data access with MASTRO. http://www.webont.org/owled/2007/PapersPDF/submission_33.pdf
65. Proc. of the Workshop on Data Abstraction, Databases and Conceptual Modelling. Pingree Park, Colorado, June 23–26, 1980. SIGMOD Record. V. 11. № 2. February 1981.
66. QUOnto Querying ONTOlogies. <http://www.dis.uniroma1.it/quonto/q=node/26>
67. *Ramakrishnan R., Ullman J.D.* A Survey of Research on Deductive Database Systems. Stanford University. Journal of Logic Programming, 23(2):125–149, May 1995. <http://dbpubs.stanford.edu:8090/pub/showDoc.Fulltextlang=en&doc=1995-14&format=pdfcompression=&name=1995-14.pdf>
68. *Ramamohanarao K., Harland J.* An Introduction to Deductive Database Languages and Systems. VLDB Journal, 3, 107–122 (1994). Invited contribution.
69. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, 10 February 2004.
70. *Rodriguez-Muro M., Lubyte L., Calvanese L.* Realizing Ontology Based Data Access: A Plug-in for Protege. <http://www.inf.unibz.it/lubyte/pub/iimas08.pdf>
71. *Roussopoulos N., Karagiannis D.* Conceptual Modeling: Past, Present and the Continuum of the Future. In: A.T. Borgida et al. (Eds.): Mylopoulos Festschrift, LNCS 5600. P. 139–152. Springer-Verlag, 2009.
72. *Savo D.F., Lembo D., Lenzerini M., Poggi A., Rodriguez-Muro M., Romagnoli V., Ruzzi M., Stella G.* MASTRO at Work: Experiences on Ontology-based Data Access. Proc. 23rd Int. Workshop on Description Logics (DL2010), CEUR-WS 573, Waterloo, Canada, 2010.
73. *Schmid H.A., Swenson J.R.* On the Semantics of the Relational Data Model. ACM SIGMOD Conf., 1975.
74. Senko M.E. DIAM II: The Binary Infological Level and Its Database Language – FORAL. Proc. of the SIGPLAN 1976 Conf. on Data: Abstraction, Definition and Structure, Salt Lake City, Utah, USA, March 22–24, 1976.
75. *Senko M.E., Altman E.B., Astrahan M.M., Fehder P.L.* Data Structures and Accessing in Data Base Systems. IBM System J. V. 12. № 1. 1973.
76. *Shipman D.W.* The Functional Data Model and the Data Language DAPLEX. ACM SIGMOD Conf., 1979.
77. *Smith J.M., Smith D.C.P.* Database Abstraction: Aggregation. Comm. of ACM, 20(6), June 1977.
78. *Smith J.M., Smith D.C.P.* Database Abstraction: Aggregation and Generalization. ACM TODS, 2(2): 1977. Русск. пер.: Джон М. Смит, Диана К. Смит. Абстракции баз данных: Агрегация и обобщение. СУБД, 2/96.
79. *Smith J.M., Smith D.C.P.* Principles of Database Conceptual Design, LNCS, 132, 1982. Русск. пер.: Дж. Смит, Д. Смит. Принципы концептуального проектирования баз данных. В сб.: Требования и спецификации в разработке программ. / Пер. с англ. под ред. В.Н. Агафонова. М.: Мир, 1984.
80. SPARQL Query Language for RDF. W3C Recommendation 15 January 2008. <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>
81. *Sundgren B.* An infological approach to data bases. Urval № 7. National Central Bureau of Statistics, Stockholm, Sweden. 1973.
82. *Sundgren B.* Data base design in theory and practice. Toward an integrated methodology. Proc. of VLDB 1978.
83. The Description Logics Handbook. Theory, Implementation and Applications. Ed. by F. Bader, D. Calvanese, D. MacGuinness, D. Nardi, P. Patel-Schneider. Cambridge University Press, 2003.
84. The Object Data Standard: ODMG 3.0. Ed. By R.G.G. Cattell and D.K. Barry. Academic Press, 2000.
85. *Turhan A.-Y., Bechhofer S., Kaplunova A., Liebig T., Luther M., Moller R., Noppens O., Patel-Schneider P., Suntisrivaraporn B., Weithoner T.* DIG 2.0 – Towards a Flexible Interface for Description Logic Reasoners. <http://www.owl-link.org/publications/dig20-OWLED06.pdf>
86. *Ullman J.* Principles of Database and Knowledge Base Systems. V. II. The New Technologies. Principles of Computer Science Series. Computer Science Press, Incorporated, Rockville, Maryland, 1989.
87. *Yourdon E., Constantine L.* Structured Design. New York: YOURDON Press, 1975.
88. *Wieringa R.* Real-World Semantics of Conceptual Models / Eds by Kaschek R., Delcambre L. The Evolution of Conceptual Modeling. LNCS 6520, Springer-Verlag, 2011.