

# Системы доступа к данным, основанные на онтологиях

---

М.Р. Когаловский  
ИПР РАН

# План

---

- Введение
- Концептуальное моделирование, его задачи
- Инструментарий концептуального моделирования
- Логика и базы данных
- Онтологическое моделирование и дескриптивные логики
- Логики DL-Lite и стандарты W3C
- Онтологии в концептуальном моделировании
- OBDA-системы, их архитектура и функционирование
- Существующие программные средства OBDA-систем
- Пример практического использования
- Выводы
- Литература

# Введение - 1

---

- В 1990-е годы начались интенсивные исследования по созданию языков описания онтологий и инструментария для их разработки и поддержки
- Результаты этих исследований привели к активной экспансии онтологий в разработки разных классов информационных систем: Веб, текстовые системы, системы баз данных
- В докладе рассматриваются полученные в последние годы значимые для практического использования результаты, связанные с использованием онтологий в качестве концептуальных схем в реляционных системах баз данных
- Такие системы называются системами доступа к данным на основе онтологий (Ontology-Based Data Access Systems, OBDA-Systems, OBDA-системы)
- Этой теме посвящено много публикаций, но ключевые работы выполнены в двух итальянских университетах; поэтому доклад опирается при обсуждении OBDA в основном на эти работы.

# Введение - 2

---

- В OBDA-системе пользователь оперирует данными реляционной БД в терминах ее концептуального представления, описываемого онтологией, и имеет возможности для рассуждения на онтологии
- Возможности OBDA-систем:
  - предоставляют развитые выразительные средства для представления базы данных и спецификации запросов
  - обеспечивают декларативность запросов в их терминах
  - обладают механизмами для рассуждений на онтологиях, а также для обработки сформулированных в терминах онтологий запросов данных в реляционных БД
  - способны осуществлять рассуждения и обработку запросов данных с приемлемой производительностью
- В докладе обсуждается предыстория этого направления в технологиях управления структурированными данными, основные идеи и принципы их реализации
- Акцент делается на технологические аспекты проблемы.

# Моделирование предметной области: истоки

---

- На ранней стадии развития технологий баз данных (БД) проектирование БД основывалось на интуиции и опыте разработчика
- Не существовало каких-либо признанных методологий и инструментальных средств для этой цели
- В конце 1960-х – начале 1970-х гг. технологии БД сформировались как значимая ветвь ИТ, стали производиться СУБД общего назначения, активизировались практические разработки систем БД
- Стали востребованными *методологии проектирования БД* и реализующий их инструментарий
- Появились средства RAD, основанные на 4GL, начали формироваться CASE-технологии, ориентированные на разработку систем БД.

# Концептуальная схема ПО

---

- В соответствии со сложившимися методологиями проектирования БД, начальным этапом этого процесса является формирование спецификации *абстрактного представления предметной области (ПО)* с помощью подходящих выразительных средств
- Этот этап называется *концептуальным моделированием ПО*, а его результат - *концептуальной схемой ПО (КС)*
- КС является *абстрактной моделью ПО*, независимой от ИТ, которые предполагается использовать для реализации системы БД
- КС абстрагируется от некоторых свойств реальных сущностей и связей между ними, а также от ИТ, которые будут использованы для реализации системы
- КС - *интенциональная модель ПО*.

# Концептуальная схема и онтология ПО

---

- Наряду с концептуальной схемой ПО, абстрактное ее представление может быть определено в форме онтологической модели
- В то время как концептуальная схема описывает структуру и, возможно, поведение ПО, онтология выполняет иную задачу, определяя понятийный аппарат ПО
- Различие функций этих моделей ПО подробно обсуждалось на первом симпозиуме по онтологическому моделированию
- На практике, однако, часто используют онтологию в качестве КС
- Таким образом поступают и авторы концепции OBDA-систем.

# Концептуальная схема и даталогические модели ПО

---

- В стандартизованных технологиях создания систем БД КС используется только *на стадии разработки*
- Синтезированная КС отображается в среду тех ИТ, которые выбраны для реализации – представляется средствами конкретной выбранной СУБД
- При этом формируются новые модели ПО, которые зависят от выбранных ИТ и представляются в терминах *даталогических моделей данных (терминология шведской школы ИС)*
- В соответствии с терминологией ANSI/X3/SPARC, эти представления называются - *концептуальной, внутренней и внешними схемами БД*
- Эти модели ПО поддерживаются системой БД в процессе ее функционирования в терминах моделей данных используемой СУБД
- При этом пользовательские интерфейсы системы оперируют БД в терминах внешних схем.



# Языки концептуального моделирования

---

- Ранние выразительные средства для представления КС ПО – *языки концептуального моделирования*, хотя и позволяли описать некоторую семантику (ограничения целостности), не отличались высоким семантическим уровнем
- *Замечание:* Далее язык КМ = концептуальная модель данных (хотя язык может иметь более широкие функции)
- Так, в 70-е годы в качестве такого выразительного средства стали использовать реляционную модель данных благодаря тому, что описание БД в ее терминах абстрагируется от организации хранимых данных
- После публикации в 1976 г. статьи П. Чена стала широко использоваться модель данных сущностей-связей (*ER-модель данных*)
- Появились индустриальные технологии проектирования ИС, основанные на ER-модели.

# Семантические модели данных

---

- Наряду с индустриальным освоением ER-модели как средства концептуального моделирования ПО продолжались активные исследования, направленные на создание более развитых *моделей данных* для проектирования БД
- Такие модели названы *семантическими*, их цель - удерживать в концептуальной схеме ПО больше семантики
- Эта цель достигалась за счет:
  - более строгого разделения концептуального и даталогического представления ПО
  - повышения семантической нагрузки на типы связей
  - использования развитых механизмов абстракции
- На разработки таких моделей большое влияние оказали работы Дж. Смита и Д. Смит по абстракции данных
- Примеры: RM/T Кодда, семантическая сеть Абриаля, функциональная модель DAPLEX Шипмана, семантическая реляционная модель Шмида и Свенсона, модель SDM Хаммера и Маклеода [SDM], инфологическая модель Сундгрена ...

# Семантические модели и пользовательские интерфейсы-1

---

- Первоначально семантические модели данных создавались как более совершенные *инструменты проектирования* БД
- Но языки пользовательских интерфейсов, основанные на даталогических моделях данных, *семантически более бедны* по сравнению с языками концептуального моделирования
- Однако возник интерес к их использованию в системах БД *и на стадии функционирования этих систем* как основы пользовательских интерфейсов
- Стали проводиться исследования и разработки *языков концептуального моделирования* для такого комплексного использования с тем, чтобы повысить семантический уровень пользовательских интерфейсов систем БД.

# Семантические модели и пользовательские интерфейсы-2

---

- Был реализован ряд проектов по созданию таких семантических моделей данных и воплощающих их языков, реализованы прототипы систем, пользовательские интерфейсы которых строились на их основе
- Примеры – ERM, проект Taxis (1980), Telos (1990), ...
- Проекты такого рода были реализованы и в СССР: Михновский (1983), Вейнеров и др.(1985), Фурсин (1987), Замулин (1990); позднее был инициирован проект Синтез (первая публикация языка Синтез - 1991)
- Эти проекты обозначили сближение технологий баз данных и систем, основанных на знаниях
- Конференция «Системы баз данных и знаний» в СССР (1989, 1991).

# Объектный подход в системах БД

---

- В конце 1980-х гг. объектный подход (ОО-подход), получивший развитие в языках программирования, распространился и на сферу технологий БД и также способствовал повышению уровня семантики систем БД
- В объектных БД (OODB) объектная КС может поддерживаться непосредственно с точностью до способа ее описания
- Консорциум ODMG, опираясь на стандарт CORBA (1991) консорциума OMG, разработал, опубликовал и поддерживал стандарт объектной модели данных с комплексом воплощающих ее языков (ODMG-93, 1993; ODMG 2.0, 1997; ODMG 3.0, 2000)
- Принятие стандарта ODMG и стандарта CORBA стимулировало производство инструментальных средств объектных технологий и активное их практическое использование.

# Достоинства объектных моделей

---

- Индивидуальность объектов позволяет моделировать поведение данных и эволюцию их во времени (объекты сохраняют индивидуальность при обновлениях)
- Классификация объектов позволяет поддерживать отношение наследования между классами
- Наследование отображает семантическое отношение между классами, позволяет совместно и повторно использовать некоторые фрагменты программного кода
- Объектная модель ODMG позволяет конструировать сложные объекты, поддерживать сложные структуры связей между ними
- Использование ОО-СУБД дает возможность существенно упростить отображение КС, созданной средствами объектного CASE-инструмента
- На основе ОО-подхода создано большое количество методологий проектирования.

# Слабые стороны объектных моделей

---

- Возможность только навигационного доступа к данным, что не соответствует декларативному характеру парадигмы моделирования реальности
- Языки запросов OODB не обладают дедуктивными возможностями
- Объектным моделям данных недостает формальной основы как в реляционных системах.

# Объектный подход и язык UML

---

- Популярность ОО-подхода в программировании и в технологиях баз данных в 1990-е гг. привела к интенсивному развитию методов объектного анализа и проектирования (ОА&Д) в разработках программных систем и систем БД
- В компании Rational, объединившей усилия крупнейших специалистов по методологиям ОА&Д - Буча, Рамбо и Якобсона, был создан язык UML - инструмент для ОА&Д, широко распространенный в настоящее время язык концептуального моделирования
- Язык стал стандартом OMG (1997), стандартом ISO/IEC 19501:2005 Information technology – Open Distributed Processing – Unified Modeling Language (UML) 1.4.2 (2005)
- В настоящее время большинство коммерческих CASE-продуктов поддерживают язык UML
- Известны проекты, в которых пользовательский интерфейс СУБД поддерживает язык UML для представления КС и язык СЛ как язык запросов, например, проект ИСП РАН (2004).



# Логика и базы данных

---

- Появление публикаций Кодда способствовало развитию применений аппарата логики в технологиях БД
- Эта роль за прошедшие годы значительно возросла
- Логика стала не только аппаратом исследования реляционной модели данных и методов проектирования реляционных БД, но и основой важного направления в технологиях БД – *дедуктивных БД (DDB)*, а в последние годы – *основой языков описания онтологий, OBDA-систем и систем интеграции данных*
- Системы дедуктивных БД могут рассматриваться как *реляционные системы продвинутого типа*, а реляционные БД как частный случай DDB
- Как полагает Minker, первыми обнаружили связь между доказательством теорем и дедукцией в системах БД Green и Raphael (1968)
- Это привело к активизации исследований в данной области.

# Логическое программирование и БД

---

- Создание языка Пролог (1972) и первого компилятора языка стимулировали развитие *логического программирования (ЛП)*
- Это в свою очередь вызвало необходимость *интеграции ЛП и технологий БД*, т.к. программы на Прологе могли оперировать только данными в оперативной памяти
- Совместное использование средств ЛП и реляционных БД позволило иметь формальный аппарат вывода и возможности работы с большими объемами данных
- Были предложены идеи расширения реляционного подхода в БД, адекватного потребностям интеграции
- Системы баз данных нового типа стали называться системами *дедуктивных баз данных (DDB)*.

# Дедуктивные базы данных

---

- В DDB база данных состоит из литеральных предикатов (экстенционал БД) и правил (интенционал БД)
- В реляционных БД отношение – совокупность значений некоторого литерального предиката, а основные операторы SQL – это логические выражения
- Поэтому, действительно, системы реляционных БД являются *частным случаем DDB*
- В 1970-1980 гг. была проведена большая серия научных конференций и рабочих семинаров, посвященных проблематике DDB
- Опубликован ряд важных статей и зданий, например, работа Reiter о гипотезе замкнутого мира (Close World Assumption), *H. Gallaire and J. Minker, editors. Logic and Databases. Plenum Press, New York, April 1978 и др.*

# Проблемы дедуктивных баз данных

---

- *Проблемы*: производительность и оптимизация запросов, отрицание и немонотонные рассуждения, проверка ограничений целостности, распараллеливание вычислений, теоретические проблемы, связанные с проблемой сложности и разрешимости и т.д.
- Был создан язык *Datalog*, который использует многие идеи Пролога, позволяет определять правила и факты, а также формулировать запросы в DDB
- В правилах *Datalog* в отличие от Пролога могут использоваться в качестве термов только литералы и переменные, но не функциональные символы; программы *Datalog* *всегда разрешимы*
- В 1984 г. в проектах *LDL* в МСС (Microelectronics and Computer Technology Corp., Texas), *Nail!* в Стенфорде и *проект ECRC* (European Computer-Industry Research Center, Мюнхен) реализованы идеи, обеспечивающие повышение производительности DDB , созданы прототипы.

# Дедуктивные базы данных: итоги

---

- DDB позволяют использовать возможности ЛП; в этом отношении они дополняют возможности реляционных БД и представляют собой шаг к системам, *основанным на знаниях*
- Достоинства DDB = логический вывод + БД, декларативность пользовательских запросов
- Однако, с точки зрения БД, такие языки ЛП, как Пролог, слишком процедурны, сложны для пользователей и слишком функционально богаты, что затрудняет эффективную оптимизацию
- Большинство усилий в исследованиях DDB посвящено *оптимизации Datalog-программ*
- DDB обеспечивают ограниченные возможности использования ЛП
- DDB не нашли индустриального применения, но сыграли большую роль в развитии теории БД как исследовательский полигон и стали основой для дальнейшего расширения сферы применения логики в технологиях ИС.

# Объектно-ориентированные DDB-1

---

- Важный следующий шаг – интеграция технологий дедуктивных и объектных БД на пороге 1990-х гг. , создание нового класса систем - DOOD (Deductive Object-Oriented Database)
- Цель DOOD – объединить лучшее дедуктивного подхода и ОО-подхода - логические основы дедуктивного подхода с модельными возможностями объектной ориентации
- Такая интеграция в рамках единой системы обеспечивает:
  - язык дедуктивных правил, позволяющих осуществлять вывод на основе хранимых данных
  - поддержку объектной схемы, включающей структурные и поведенческие аспекты
- Главные возможности DOOD: поддержка концепции объекта, индивидуальность, сложные объекты, типизация, основанные на правилах методы, инкапсуляция методов, перегрузка свойств, полиморфизм, иерархия классов, множественное наследование ...
- DOOD обеспечивает логическую формализацию ОО-подхода.

# Объектно-ориентированные DDB-2

---

- Было создано большое количество языков (моделей данных) для DOOD:
  - O-logic (1986, пересмотренная -1993), C-logic (1989), IQL (1998), LOGRES (1990), LLO (1991), COMPLEX (1992), ORLOG (1992), LIVING IN LATTICE (1993), *Datalogmeth* (1993), CORALCC (1993), Noodle (1993), DTL (1993), F-logic (1995), Gulog (1995), Rock & Roll (1995), ROL (1996), DatalogCC (1997), ROL2 (1998, 1999), Chimera (1998), DO2 (1998)...
- Один из наиболее известных языков - F-logic (Кифер и Лаузен, Университет в Мангейме)
- Этот язык, созданный первоначально для DOOD, используется в ряде исследовательских проектов как средство формального описания онтологий, но большей популярностью для этой цели обладают дескриптивные логики.

# Дескриптивные логики

---

- Другая ветвь применения аппарата логики сформировалась в рамках исследований *представления структурированных знаний* и рассуждений о них
- Для этих целей было разработано популярное в настоящее время семейство логик, называемых *дескриптивными логиками* (*Description Logics, DLs*)
- В последние годы DLs рассматриваются как подходящий формализм описания онтологий, а также разработки систем семантической интеграции данных и систем БД с семантическими пользовательскими интерфейсами (*OBDA-систем*)
- DLs позволяют специфицировать классы сущностей и связи между ними, обеспечивают формальную семантику, а также механизмы вывода
- Логика этого семейства позволяют выразить некоторые главные формализмы объектных моделей данных, ER-модели; т.о. они обладают достаточно развитыми возможностями концептуального моделирования.



# Онтологии и дескриптивные логики

---

- В последние годы стали актуальными попытки использования онтологий в ИС различных классов, в частности, в системах БД и системах интеграции данных из множества источников
- Появился инструментарий для эффективного использования формальных онтологий – дескриптивные логики
- Соотношение онтологий с логикой и с другими языками в ИС [7]:
  - *с языками представления знаний*: онтологии являются схемами представления знаний
  - *с логикой*: логика – это инструмент для придания семантики онтологическим языкам
  - *с концептуальной моделью данных (точнее, с концептуальной схемой)*: концептуальные схемы являются специальными онтологиями, подходящими для концептуализации отдельной (конкретной) логической модели базы данных
  - *с языками программирования*: определения классов являются специальными онтологиями, подходящими для концептуализации отдельной (конкретной) структуры для вычислений.

# Стандарты OWL, OWL2 и профили

---

- Основой разработки дескриптивной логики для OBDA-систем с приемлемыми оценками вычислительной сложности стал диалект OWL DL стандарта OWL (2004) , обеспечивающий разрешимость и вычислимость запросов
- Было показано, что не вводя ограничений на функциональность этого языка, нельзя достигнуть желательного уровня сложности при обработке запросов данных
- В 2009 г. консорциум W3C принял новую версию стандарта языка описаний онтологий – OWL2
- В версии OWL2 появились полезные новые конструкции, обогащающие выразительные возможности и не нарушающие, тем не менее, разрешимость и вычислимость в тех рамках, в которых они обеспечивались прежней версией языка
- В стандарте OWL2 предусматривается три профиля: OWL2 QL, OWL2 EL, OWL2 RL
- Каждый профиль – это подязык (фрагмент) OWL2 DL, предназначенный для своей сферы применения и обеспечивающий в этой сфере лучшие вычислительные свойства по сравнению с OWL2 DL
- Для OBDA-систем предназначен профиль OWL2 QL.

# Профиль OWL2 QL

---

- Профиль OWL2 QL основан на дескриптивной логике  $DL-Lite_{A,id}$  семейства *DL-Lite*, которому был посвящен большой цикл исследований, представителей итальянской школы (D. Calvanese, G. Giacomo, D. Lembo, M. Lenzerini, A. Poggi, R. Rosati и др.)
- Логики этого семейства разрабатывались с главной задачей – обеспечение компромисса между выразительной силой основанных на них языков и сложностью обработки запросов
- В исследованиях OBDA-систем использовались именно логики семейства *DL-Lite*, позволяющие эффективно обрабатывать не только запросы на онтологии, но и запросы данных
- Теперь имеется стандартизованная версия DL, основанная на логике этого семейства – профиль OWL2 QL
- OWL2 QL включает большинство возможностей концептуального моделирования диаграммы классов UML, а также ER-модели
- Он позволяет эффективно поддерживать онтологический интерфейс для SQL-СУБД при обработке UCQs в приложениях с большим объемом данных.

# Семейство DL-Lite

---

- Цикл работ итальянской группы, опубликованных в 2005-2009 гг.
- DL-Lite специально разрабатывалась как основа языков описания онтологий, которые бы обеспечивали низкий уровень вычислительной сложности рассуждений в терминах онтологии и обработки сложных конъюнктивных запросов в больших БД
- Логики этого семейства обладают полиномиальной сложностью обработки запросов относительно размера TBox и LogSpace относительно размера ABox (*сложности по данным*)
- Предложенная логика позволяет осуществлять рассуждения на TBox независимо от ABox и обработку запросов на ABox выполнять независимо от TBox
- Обработка запросов на ABox может при этом выполняться средствами реляционной SQL-СУБД
- Благодаря этому оптимизация запросов может выполняться с помощью механизмов, которыми располагают такие СУБД
- Далее авторам удалось показать (2007), что DL-Lite<sub>R</sub> представляет собой максимальный фрагмент OWL DL с указанными свойствами.

# И снова семантический доступ к данным

---

- Достижения последних лет в разработке приложений логики в ИС, активные разработки в области создания языков описания онтологий и их стандартизация вновь актуализировали проблему создания систем, обеспечивающих семантический доступ к данным
- Стало возможно создание таких систем, в которых онтология играет роль высокоуровневой *концептуальной схемы*, поддерживаемой на пользовательском интерфейсе (над традиционной SQL-СУБД)
- Механизмы поддержки онтологии становятся посредником между пользователем с его информационными потребностями, выраженными средствами онтологии, и системой базы данных
- Достоинства подхода: высокий уровень абстрактности представления данных в БД, развитые выразительные средства запросов, декларативность, приемлемая производительность системы, запросы и на онтологии, и в базе данных
- В основных публикациях на эту тему такие системы называются *Ontology-Based Data Access Systems*.

# Онтологии в OBDA-системах

---

- Одним из результатов наметившейся тенденции использования онтологических языков для концептуального моделирования стало создание OBDA-систем
- Использование онтологий в качестве концептуальных моделей предметной области в OBDA-системах:
  - обеспечивает более абстрактное представление БД, чем при использовании традиционных моделей данных, не связанное с «логической» структурой БД
  - дает возможность использовать явно не определенные (скрытые) отношения
  - позволяет проверять качество данных, обнаруживать неожиданную неполноту данных в источниках
  - имеет побочный эффект - упорядочение и документирование терминологии организации, использующей OBDA-систему.

# Онтологии в концептуальном моделировании-1

---

- Важно оценить, в какой мере OWL и его профили позволяют выразить возможности моделирования, обеспечиваемые широко используемыми концептуальными моделями данных
- Относительно OWL2 QL утверждается, что его средствами можно выразить большинство модельных элементов диаграммы классов UML, а также ER-модели
- Этот вопрос, однако, нуждается в более серьезном изучении
- Известно, что прототипом OWL2 QL была логика  $DL-Lite_{A,id}$  семейства DL-Lite
- Исключены: Unique Name Assumption – нет в OWL2, Identification Assertion – функциональность ролей и атрибутов
- Дополнены: не влияющие на вычислительную сложность утверждения свойств ролей; поддерживается типизация данных OWL2, которой нет в логике-прототипе
- Можно утверждать, что модельные возможности OWL2 QL близки возможностям его прототипа.

# Онтологии в концептуальном моделировании-2

---

- В «библии» OBDA (Ontologies and Databases: The DL-Lite Approach, 2009 [5]) авторы действительно показывают возможности представления некоторых важных модельных элементов диаграммы классов UML средствами логики DL-Lite<sub>A,id</sub>
- Но: Авторы утверждают, что поскольку диаграмма классов рассматривается как средство **не программной инженерии**, *а концептуального моделирования (??)*, то можно при оценке модельных возможностей этой логики не рассматривать поведенческие аспекты языка – методы, ассоциируемые с классами (!?) – *Сравнить с т. зр. Workshop on Conceptual Modeling*
- В документах OWL2 также явным образом признается, что декларативный характер онтологических языков не позволяет моделировать методы ОО-подхода
- В качестве достоинства OWL2 отмечается бо'льшая его гибкость по сравнению с языками концептуального моделирования, позволяющая иметь дело с неполнотой информации, выводить не заданные явно отношения.



# Определение OBDA-системы

---

- Термин *Ontology-Based Data Access System* введен авторами семейства логик DL-Lite (*Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati*)
- OBDA-система рассматривается как сервис над множеством существующих источников данных, предназначенный для предоставления пользователю системы концептуального представления содержащихся в них данных
- Приводится и формальное определение:  
Ontology-Based Data Access System = триплет  $O = \langle T; M; D \rangle$ ,  
где:    Т - это Tbox  
          D – реляционная база данных  
          M – множество утверждений отображения между T и D
- В литературе используются также термины:  
*Accessing Data Mediated by an Ontology,*  
*Ontology-driven Information Systems,*  
*Ontology based Data Management.*

# Замечания относительно определения

---

- Авторы термина OBDA **отождествляют** *OBDA-системы* с *системами интеграции данных* (в определении говорится о множестве источников данных, хотя не упоминаются никакие механизмы интеграции)
- В других работах со ссылкой на результаты по интеграции данных, утверждается, что можно считать множество источников единой виртуальной реляционной SQL-БД
- Концептуальные схемы рассматриваются авторами как *специальный вид онтологий*:

*“Conceptual schema are special ontologies, suited for conceptualizing a single logical model (database)” [6].*

# Требования к OBDА-системам

---

- OBDА-системы должны обеспечивать рассуждения на онтологиях
- Должны обрабатываться также запросы *к большим объемам структурированных данных*, сформулированные в терминах онтологий
- Онтологии должны использоваться в OBDА-системах в качестве *концептуальной схемы*, поддерживаемой пользовательским интерфейсом
- Должна обеспечиваться *приемлемая сложность* этих операций
- Эффективность доступа к данным должна достигаться отображением запросов в среду реляционной СУБД, которая будет обрабатывать запросы и *создана независимо от онтологии*
- При этом накладные расходы использования интерфейса, поддерживающего онтологию, как надстройки над обычной системой БД не должны быть значительными
- DL, выбранная в качестве основы языка описания онтологий должна включать основные возможности, используемые в концептуальном моделировании.

# Как удовлетворить требования?

---

- Для удовлетворения перечисленных требований к OBDA-системам необходимо *найти компромисс* между выразительной силой языка и вычислительной сложностью запросов на онтологии и на БД
- На поиски этого компромисса и были направлены исследования DLs в последние годы
- Поскольку в качестве репозитория данных в OBDA-системах предполагается использовать реляционные БД, то система должна допускать запросы, переписываемые в логику первого порядка и тем самым отображимые в язык SQL
- В случае неполноты данных в БД SQL-запросы становятся неразрешимыми
- Компромиссный вариант выразительной силы языка запросов в OBDA-системе – отображимость во фрагмент языка SQL, позволяющий формулировать конъюнктивные запросы и их объединения (UCQs).

# Источники данных в OBDA

---

- Предполагается, что источником данных в OBDA является ABox онтологии, представленной средствами подходящей дескриптивной логики
- Предполагается также, что контент ABox составляют абстрактные объекты ПО, которые возвращаются пользователю с их свойствами в ответах на запросы
- Контент ABox соответствует TBox онтологии в том смысле, что используемые в нем концепты, роли и атрибуты определены в TBox
- Предполагается, что источник данных имеет значительный объем и хранится как база данных, управляемая реляционной СУБД
- Контент ABox в системе не материализован, он является *виртуальным* – его элементы порождаются системой БД при обработке запросов.

# Архитектура OBDA-систем

---

- Архитектура OBDA-системы вполне согласуется со сложившимися представлениями об архитектуре систем БД
- Принцип *независимости данных*, многоуровневая архитектура систем БД
- Уровни архитектуры OBDA:
  - механизм поддержки онтологий и рассуждений на онтологиях (резонер) с пользовательским интерфейсом
  - механизм отображения запросов и данных в среду реляционной БД
  - SQL-система базы данных
- Обеспечивается использование эффективных средств управления структурированными данными реляционной СУБД
- Сохраняется принцип независимости данных: SQL-система БД инкапсулируется для пользователя, он не обязан о ней знать.

# Отображение между онтологией и схемой БД

---

- Отображение определяется проектировщиком конкретной OBDA-системы
- Отображение описывается как совокупность утверждений отображения двух видов:
  - типизированные утверждения отображения
  - утверждения отображения «данные-объекты»
- Типизированные утверждения отображения определяют соответствие типов онтологии данным в БД
- Утверждения отображения «данные-объекты» описывают отображение данных БД в экземпляры концептов, ролей и атрибутов онтологии
- Сложность в том, что онтологии основаны на гипотезе открытого мира, а источник (БД) – на гипотезе замкнутого мира
- Другая проблема - несоответствие импеданса.

# Несоответствие импеданса

---

- Одна из проблем отображения запросов, определенных на уровне онтологии, в среду используемой SQL-СУБД состоит в несоответствии импеданса
- Существо этой проблемы в OBDA-системах:
  - в источнике данных хранятся структурированные данные различных типов – числа, строки, булевские значения и т.п.
  - в то же время запросы формулируются в терминах объектов, представляющих в онтологии концепты, их роли и атрибуты, отношения между ними
- Поэтому механизмы отображения должны конструировать из значений данных в источнике идентификаторы тех абстрактных объектов, которые составляют АВох в онтологии
- Для решения этой проблемы используются сколемовские функции, подход заимствован в работах Р. Хала (R. Hull).



# Обработка запросов данных

---

- Запросы данных обрабатываются в OBDA-системах в три этапа
- Первый этап:
  - запрос над TBox переформулируется в запрос в FOL, который может выполняться в ABox (в реляционной СБД);
  - задача Query Answering сводится при этом к Query Evaluation; TBox далее не используется; профиль OWL2 QL основан на DL-Lite<sub>A,id</sub>
  - для этой логики алгоритм такой редукции описан в [5]
- Второй этап:
  - отображение полученного запроса в FOL в среду реляционной системы БД (переписывание в SQL, препроцессинг)
  - обеспечено получение только запросов вида CQ или UCQ (Union + Select-Project-Join)
- Третий этап:
  - обработка запроса в ABox, представленного как реляционная система БД.

# Вычислительная сложность обработки запросов

---

- Авторы логики DL-Lite<sub>A,id</sub> получили следующие оценки вычислительной сложности для OBDA, основанных на этой логике:
  - для рассуждений на TBox - полиномиальная сложность (относительно размера TBox)
  - для обработки запросов на ABox (сложность по данным) - LogSpace (относительно объема БД)
- Оценка для запросов на ABox следует из того, что они сводятся в OBDA-системах, имеющих рассмотренную архитектуру, к обработке SQL-запросов в реляционной системе БД
- Эти оценки имеют силу и для систем, использующих профиль QWL2 QL.

# Отличия OBDA-систем от систем DDB

---

- Различия функциональности OBDA-систем и систем DDB:
  - используется иной класс логик
  - в отличие от DDB предусматриваются рассуждения не только над интенционалом, но и запросы над экстенционалом
  - хранение данных в DDB – не «черный ящик»
  - как следствие отображение интенционала в даталогическую среду явным образом специфицируется
  - более низкий уровень вычислительной сложности рассуждений и обработки запросов
  - большой объем экстенционала небольшой, соответствующий реальным приложениям
  - используются стандартизованные языки описания интенционала и экстенционала.

# Реализации OBDA-систем-1

---

## Система QuOnto (QUerying ONTOlogies)

- Использует для описания онтологий логику семейства DL-Lite
- Имеет средства рассуждений над онтологиями и обработки запросов данных
- Механизм рассуждений хорошо оптимизирован
- Может оперировать внутренними данными или данными в системе БД
- Располлагает драйверами для ряда СУБД: Oracle, DB2, SQL server, MySQL ...
- Реализована на языке Java
- Имеет API для некоторых специальных проектов
- Имеются свободно доступные версии с адаптерами (wrapped versions): ROWLkit, QToolKit, Адаптер DIG Server с плагином OBDA Protégé.

## ROWLkit (первая реализация OWL2 QL Profile)

- Система с графическим пользовательским интерфейсом для рассуждений над OWL2 QL онтологиями, их верификации и обработки запросов данных
- Использует сервисы QuOnto, дополненные средствами для того, чтобы иметь дело с онтологиями OWL2 QL
- Он воспринимает в качестве входных данных онтологии OWL2 QL через OWL API
- Реализован в Java и использует in-Memory СУБД H2 Java для хранения Abox
- Может оперировать также с данными во внешней памяти.

# Реализации OBDA-систем-2

---

## QToolKit

- Графический интерфейс для QuOnto, позволяющий представлять DL-Lite- онтологий и осуществлять рассуждения на них средствами ее резонера
- Позволяет использовать все возможности рассуждений QuOnto
- Abox хранится во внутренней репозитории данных

## DIG Server wrapper + OBDA Protege Plug-in

- Графический интерфейс для рассуждений средствами OuOnto над DL-Lite- онтологиями
- QuOnto предоставляет для DIG Server'а интерфейс, позволяющий специфицировать отображение DL-Lite онтологии в среду систем баз данных Oracle, DB2 и др.
- Plug-in с открытым кодом для Protégé, расширяющий возможности редактирования онтологий, который может использоваться как клиент для интерфейса Onto DIG для спецификации и выполнения запросов на DL-Lite онтологиях с отображениями.

## MASTRO

- The Mastro – OBDA-система – расширение резонера QuOnto
- Базируется на DL-Lite<sub>A</sub>-логике
- Позволяет специфицировать онтологии, выполнять запросы и метазапросы, проверять выполнимость онтологии (взаимную непротиворечивость данных).

# Пример практического применения

---

- MASTRO использовалась, в частности, в банковской системе
- Детальный анализ опыта опубликован в статье [21]
- Онтология включает: 79 концептов, 33 роли, 37 атрибутов концептов, около 600 аксиом, около 50 ограничений
- Большая работа – создание онтологии
- Сложности описания отображения
- Необходимо основательное знание источника и онтологии
- Высокая производительность системы
- Проверено качество источника данных – обнаружены неполнота и противоречивость данных.

# Выводы

---

- Проведены массированные исследования и отработана теоретическая основа OBDA-систем, обеспечивающих практически приемлемую сложность при достаточно высокой выразительной силе языка описания онтологий
- Онтологические языки могут использоваться для концептуального моделирования в качестве основы семантического пользовательского интерфейса в таких системах
- Основные исследовательские результаты в данной области получены в университетах Sapienza Universita di Roma и Libera Universita di Bolzano
- Помимо теоретических исследований, создан ряд прототипов, которые успешно используются в реальных приложениях (QuOnto, MASTRO)
- Разработан профиль стандарта OWL2 языка описания онтологий - OWL2 QL, базирующийся на семействе дескриптивных логик (его прототип – логика DL-Lite<sub>A,id</sub>)
- Можно считать, что технологии OBDA-систем достигли уровня проработки, оправдывающего промышленную реализацию, и тиражируемые программные средства для их разработок, несомненно, появятся в близкое время.

# Литература-1

---

- 1. S. Abiteboul. 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: 26-29
- 2. S. Abiteboul. Towards a deductive object-oriented database language. Data & Knowledge Engineering Volume 5, Issue 4, October 1990, p. 263-287
- 3. A. Acciarri, D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, M. Palmieri, R. Rosati. QUONTO: QUerying ONTOlogies. Proc. of the 20th national conf. on Artificial intelligence - Volume 4. Pittsburgh, Pennsylvania, 2005, pp. 1670-1671.
- 4. D. Calvanese. 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>
- 5. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, and R. Rosati. Ontologies and Databases: The DL-Lite Approach. Semantic Technologies for Informations Systems - 5th Int. Reasoning Web Summer School (RW 2009). LNCS, Vol. 5689, 2009.
- 6. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. JAR, 39(3):385–429, 2007.
- 7. D. Calvanese, D. Lembo. Ontology-Based Data Access. Tutorial. 6th Int. Semantic Web Conference (ISWC 2007), Busan, South Korea. - Nov. 12th, 2007
- 8. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, and R. Rosati. Ontology-Based Database Access. Proc. of the 15th Italian Conf. on Database Systems (SEBD 2007), 2007.



# Литература-2

---

- 9. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati. DL-Lite: Tractable Description Logics for Ontologies. In Proc. of AAAI 2005, pp. 602–607, 2005.
- 10. E. Franconi. Ontologies and Databases: myths and challenges. PVLDB '08, August 23-28, 2008, Auckland, New Zealand. VLDB Endowment, ACM, 2008.
- 11. H. Gallaire and J. Minker, editors. Logic and Databases. Plenum Press, New York, Apr. 1978.
- 12. M. Kifer, G. Lausen. F-Logic: A Higher-Order Language for reasoning about Objects, Inheritance, and Scheme. June 3, 1997
- 13. J. Minker. Logic and Database: a 20 Year Retrospective. Department of Computer Science. Institute for Advanced Computer Studies. University of Maryland. College Park, MD, 20742, USA. In: Workshop on Logic in Databases, San Miniato, Italy, 1996.
- 14. Mengchi Liu, Gillian Dobbie, and Tok Wang Ling. A Logical Foundation for Deductive Object-Oriented Databases. National University of Singapore. ACM Transactions on Database Systems, Vol. 27, No. 1, March 2002, pp. 117–151
- 15. J. Peckham and F. Maryanski. Semantic Data Models. ACM Computing Surveys, Vol. 20, No. 3, September 1988.
- 16. H. Perez-Urbina, B. Motik, and I. Horrocks. Rewriting conjunctive queries under description logic constraints. In Proc. of the Workshop on Logic in Databases (LID 2008), 2008.
- 17. A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati. Linking Data to Ontologies. JODS-2008
- 18. A. Poggi and M. Ruzzi. Ontology-based data access with MASTRO. [http://www.webont.org/owled/2007/PapersPDF/submission\\_33.pdf](http://www.webont.org/owled/2007/PapersPDF/submission_33.pdf)

# Литература-3

---

- 19. R. Ramakrishnan and J. Ullman. A survey of research on deductive database systems, 1993. <http://www.iai.uni-bonn.de/III/lehre/vorlesungen/DeduktiveDatenbanken/SS06/Downloads/RU93.pdf>
- 20. M. Rodriguez-Muro, L. Lubyte, D. Calvanese. Realizing Ontology Based Data Access: A Plug-in for Protégé. <http://www.inf.unibz.it/~lubyte/pub/iimas08.pdf>
- 21. D.F. Savo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, V. Romagnoli, M. Ruzzi, G. Stella. MASTRO at Work: Experiences on Ontology-based Data Access. Proc. 23rd Int. Workshop on Description Logics (DL2010), CEUR-WS 573, Waterloo, Canada, 2010.
- 22. B. Smith. Ontology and Information Systems. [http://ontology.buffalo.edu/ontology\(PIC\).pdf](http://ontology.buffalo.edu/ontology(PIC).pdf)
- 23. J. Ullman. Principles of Database and Knowledge\_Base Systems. Volume I. Principles of Computer Science Series. Computer Science Press, Incorporated. Rockville, Maryland, 1988.
- 24. J. Ullman. Principles of Database and Knowledge\_Base Systems. Volume II. The New Technologies. Principles of Computer Science Series. Computer Science Press, Incorporated, Rockville, Maryland, 1989.
- 25. OWL 2 Web Ontology Language Document Overview. W3C Recommendation 27 October 2009. <http://www.w3.org/TR/2009/REC-owl2-overview-20091027/>
- 26. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C Recommendation 27 October 2009. <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/>
- 27. OWL 2 Web Ontology Language Primer. W3C Recommendation 27 October 2009. <http://www.w3.org/TR/2009/REC-owl2-primer-20091027/>
- 28. OWL 2 Web Ontology Language Profiles. W3C Recommendation 27 October 2009. <http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/>

# Конец

---

Спасибо за внимание