

ОНТОЛОГИЧЕСКИЙ ИНЖИНИРИНГ

С.М. Авдошин,

к.т.н., профессор, заведующий кафедрой управления разработкой программного обеспечения, руководитель отделения программной инженерии факультета бизнес-информатики ГУ-ВШЭ

М.П. Шатилов, студент второго курса отделения программной инженерии факультета бизнес-информатики ГУ ВШЭ

Рассмотрены основные понятия онтологического инжиниринга, языки описания онтологий и занимаемая ими ниша в контексте семантической паутины. Проанализированы инструменты онтологического инжиниринга и наиболее конкурентоспособные инструменты для построения, отображения и объединения онтологий.

Введение

«О тбросьте свои изношенные, ветхие идеи о лидерстве. Самая успешная корпорация 1990-х гг. получит название обучающейся организации». Эти слова красовались на страницах одного из лидеров деловой прессы журнала Fortune, освещающего тенденции развития мирового бизнеса. В сложном и динамично развивающемся мире конкурентное преимущество компании возможно только благодаря использованию идей, влияющих на современные тенденции развития бизнеса. Использование организацией передовых технологий своего времени — реальный пропуск в лидеры. Информационные порталы, электронная коммерция, системы сбалансированных показателей, реинжиниринг бизнес-процессов, управление знаниями — это лишь часть идей, могущих позволить компании одолеть своих соперников. По утверждению многих специалистов, технология управления знаниями — единственный источник надежного и стойкого конкурентного преимущества компании. Информация, накопленная её сотрудниками, должна превращаться в ценные и осмысленные знания, которые в дальнейшем станут руководством к действию. Таким образом, конкурентное преимущество компании предлагает эффективный сбор, анализ и представление данных для своих сотрудников, которые на основании полученных знаний смогут принимать верные решения.

Многие компании обладают огромными массивами информации, рассредоточенной в базах и хранилищах данных, на бумаге, в отчетах о проделанной работе и в деловой переписке. Но самое главное — они имеют опыт, накопленный в течение многих лет и хранящийся в головах сотрудников. Проблема заключается в извлечении и концентрировании этих знаний в одном месте в удобной и доступной форме. Одно из важнейших и перспективных направлений в области формализации знаний — *онтологический инжиниринг* — процесс проектирования и разработки *онтологий*. При отсутствии общепринятой методологии и технологии этот процесс не тривиальная задача, требующий от разработчиков профессионального владения технологиями инженерии знаний — от методов их извлечения до структурирования и формализации [1].

Онтология — *спецификация на концептуальном уровне* [2]. Данный термин произошел от названия раздела философии, изучающего проблемы бытия. Онтология в широком смысле — это наука о бытии. В контексте информационных технологий этот термин приобрел несколько другой смысл. Здесь онтология — попытка всеобъемлющей и детальной формализации определённой области знаний с помощью концептуальной схемы; включает в себя словарь указателей на термины предметной области и логические выражения, описывающие соотношение между ними. То есть онтология — сеть, содержащая термины и понятия и показывающая взаимосвязь между ними. Онтологии:

- ✧ применяются в искусственном интеллекте и семантической паутине как форма представления знаний о реальном мире в цифровом формате;
- ✧ позволяют представить понятия в таком виде, что они становятся пригодными для компьютерной обработки;
- ✧ остаются понятными не только машине, но и человеку;
- ✧ по сути являются интерфейсом общения между людьми и компьютерами.

Классификация онтологий

Обычно используют три классификации онтологий [3]:

- ✧ по степени формальности;
- ✧ по наполнению содержимым;
- ✧ по цели создания.

Рассмотрим соответствующие классификации более подробно.

Спектр онтологий — классификация онтологий по степени формальности

На рис. 1 представлена классификация (спектр) онтологий в зависимости от деталей их реализации [3]. Косой чертой на рисунке отделены «машинопонятные» и «человеческопонятные» описания.

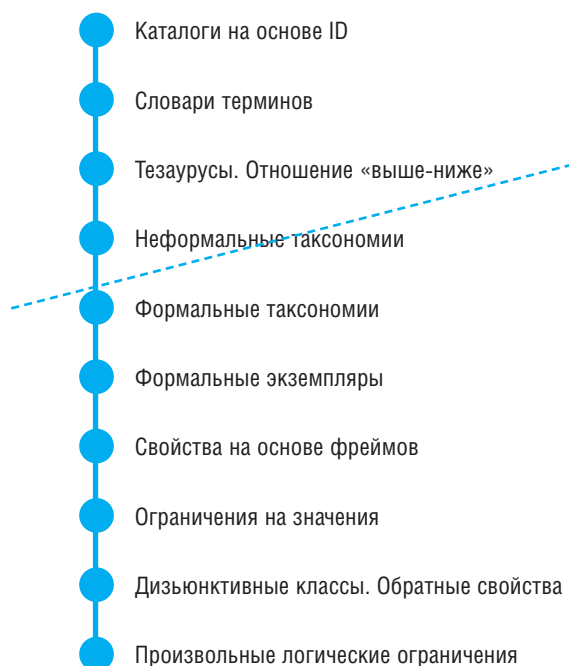


Рис. 1. Спектр онтологий

Каталоги на основе ID — конечный список терминов — представляют точную интерпретацию терминов. Например, говоря слово «онтология» мы будем использовать одно и то же значение, соответствующее некоторому ID в каталоге.

Словари терминов (гlossарии), дающие грамматическую и стилистическую характеристики терминов, примеры употребления и другие сведения, непригодны для автоматической обработки программными агентами.

Тезаурусы — нормативные словари ключевых слов и дескрипторов (словарных единиц в виде слов, словосочетаний или кодов, называющих класс условной эквивалентности, в который включены эквивалентные и близкие по смыслу ключевые слова) [4].

Формальные таксономии (греч. расположение по порядку, закон) — это онтологии, включающие точное определение отношения isA (класс-подкласс). Здесь соблюдается отношение транзитивности [3]: если B — подкласс класса A, то каждый подкласс класса B тоже является подклассом класса A. В *неформальных таксономиях* это отношение выполняется не всегда.

Формальные экземпляры. Не все онтологии содержат только имена классов; многие содержат на нижнем уровне экземпляры (индивиды) классов. Такого рода онтологии относятся к данной категории.

Онтологии, в структурных элементах которой содержатся *фреймы*, будут рассмотрены ниже.

При необходимости выразить больше информации выразительные средства онтологии (и её структура) усложняется. Например, может потребоваться заполнять значение какого-либо свойства экземпляра, используя математическое выражение, основанное на значениях других свойств и даже других экземплярах. Многие онтологии позволяют объявлять два и более классов *дизъюнктивными* (непересекающимися). Это означает, что у данных классов нет общих экземпляров.

Классификация онтологий по цели создания

В данной классификации выделяют следующие типы онтологий:

- ✧ *онтологии представления*. Цель создания таких онтологий — описать семантику языков, которые будут в дальнейшем использоваться для описания мета-онтологий, онтологий предметной области и прикладных онтологий. Например, средствами RDF/RDFS [5] описаны понятия языка OWL [6];

- ✧ *мета-онтологии* не зависят от выбранной предметной области. Ими характеризуются общие понятия, на основании которых строятся онтологии более низкого уровня. Такого рода онтологии повторно используются в предметных областях;
- ✧ *онтологии предметной области*; с их помощью формально описываются предметные области, в которых уточняются понятия, определенные в мета-онлогиях. Эти онтологии могут повторно использоваться в рамках той предметной области, для которой они созданы;
- ✧ *прикладные онтологии* описывают концептуальную модель конкретной задачи или приложения. Возможность их повторного использования отсутствует.

Классификация онтологий по содержанию

Отличие этой классификации от предыдущей в том, что здесь рассматривается реальное содержание онтологии, а не абстрактная цель ее создания. Различают *общие онтологии*, *онтологии конкретных задач* и *предметные онтологии*.

Элементы онтологии

Все современные онтологии (независимо от спецификации языка и его программной реализации) строятся примерно одинаково. Основные компоненты онтологий:

- ✧ *концепты* (классы, понятия, сущности, категории);
- ✧ *свойства* концептов (слоты, атрибуты, роли);
- ✧ *отношения* между концептами;
- ✧ некоторые *ограничения* (фацеты ограничения ролей).

Экземпляр концепта служит для представления элемента предметной области, группу элементов которого описывает концепт (например, человек — это концепт, а Агафья Петровна — экземпляр концепта). Сама онтология и множество экземпляров составляют *базу знаний*.

Концепт — это шаблон, содержащий множество правил, определяющих форму экземпляра, т.е. то каким образом может быть построен экземпляр. Возвращаясь к примеру с Агафьей Петровной, человек является концептом потому что это некоторая обобщенная сущность, не имеющая четких очертаний, но обладающая правилами, описывающая, каким образом эти очертания должны быть заданы.

Концепты могут иметь атрибуты — имена или структуры полей записи и характеризуют размер или тип информации, содержащейся в поле; используются для хранения информации об экземпляре концепта. Значение атрибута может быть как сложным, так и простым типом данных. К примеру, возраст Агафьи Петровны — это атрибут, а полное количество лет, присвоенное атрибуту возраст, значение атрибута.

Отношениями между концептами называются зависимости между экземплярами онтологий. Обычно отношением является атрибут, ссылающийся на другой экземпляр. К примеру, друзья Сидор и Борислав любят поиграть во дворе Агафьи Петровны, при этом Сидор приходится ей внуком. В таком случае значением атрибута «Внук» у Сидора будет равно «Агафья Петровна».

Читатели, знакомые с идеями объектно-ориентированного программирования (ООП), могли заметить некоторую схожесть описания классов ООП с описанием классов онтологий. ООП сосредотачивается главным образом на методах классов — программист принимает проектные решения, основанные на операторных свойствах класса; разработчик онтологии принимает эти решения, основываясь на структурных свойствах класса [7]. Поэтому структура класса и отношения между классами в онтологии в значительной степени отличаются от структуры подобной предметной области, описанной в терминах ООП.

Формальная модель онтологии

Под формальной моделью онтологии понимают

$$O = \langle T, R, F \rangle,$$

где T — конечное множество терминов предметной области, которую описывает онтология O ;
 R — конечное множество отношений между терминами описываемой предметной области;
 F — конечное множество функций интерпретации, которые задаются на терминах и отношениях онтологии [8].

Здесь появляется еще одна классификация — классификация онтологий по их структуре [1], в которой модели онтологий разделяются на три группы: *простые* (в которых множества R и F могут отсутствовать), *на основе фреймов* и *на основе логик*.

Понятие «фрейм», предложенное М. Минским, одним из основателей теории искусственного

интеллекта, обозначает структуру данных для представления стереотипной ситуации (для обозначения структуры знаний для восприятия пространственных сцен [1]). По сути, фрейм — некоторая логическая запись, где каждому слоту ставится в соответствие значения, присоединенные процедуры или другие фреймы. Они служат для описания ситуаций, объектов и других понятий, а также для описания взаимозависимостей между ними. Фрейм — это попытка сохранить и впоследствии представить знания таким образом, чтобы максимально приблизить к тому, как это делает человеческий мозг. Это напоминает хранение наших знаний в довольно крупных размерах. При этом они очень тесно переплетаются между собой.

Примером является все та же Агафья Петровна. Её описание на основе сети фреймов приведено на рис. 2.

Другой тип организации структуры онтологии — описание ее структуры на основании логических моделей. В представлении знаний выделяют формальные логические модели, основанные на классическом исчислении предикатов первого порядка (first-order logic) и дескриптивной логике (description logic). В первом случае предметная область описывается конечным набором аксиом. Они являются общими для любой области объектов исследования с заданными на этих объектах предикатами (т.е. свойствами и отношениями). Такая логика предъявляет очень высокие требования и ограничения к предметной области, поэтому используется мало. Наибольшую популярность в настоящее время получил класс дескриптивных логик.

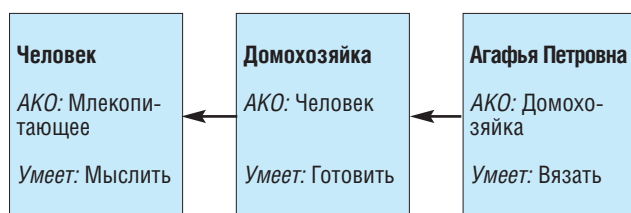


Рис. 2. Сеть фреймов (АКО — A Kind OF)

Дескриптивные логики — семейство формализмов для представления знаний — основаны на таких понятиях, как концепты и роли и характеризуются конструкциями, позволяющими строить сложные концепты и роли из более простых. Основная выгода от этого семейства формализмов в том, что с их помощью могут быть описаны устойчивые и полные алгоритмы для решения проблем категоризации и выполнимости. Механизм рассуждений в дескриптивных логиках решает проблемы эквивалентности, выполнимости и категоризации [8].

Применение онтологий

Исследования в области онтологий и онтологических систем довольно перспективны. С одной стороны, это развитие семантического веба, а с другой — искусственного интеллекта.

Один из самых крупных проектов в области семантического веба — дублинское ядро (*Dublin Core*), служащее стандартом метаданных для описания большого диапазона сетевых ресурсов [9]. Этот стандарт состоит из двух уровней: простого (*simple*) и квалифицированного (*qualified*). Если простой уровень содержит в себе 15 элементов, то квалифицированный включает в себя еще три дополнительных элемента и группу квалификаторов, уточняющих семантику элементов. Все это качественно повышает уровень поиска необходимых данных. Например, системы помощи GNOME [10] и KDE [11] основаны на проекте «дублинское ядро».

Проект *Enterprise project* ставит перед собой целью улучшение существующих методов моделирования работы организации [12]. Ведется разработка инструментальных средств для полного описания работы предприятия. Однако здесь возникают семантические конфликты, при которых под одним и тем же термином понимают разные вещи. Эта проблема решается построением онтологии. В ней заданы основные термины предметной области, в которой ведется разработка.

Онтологии используют и в системе образования. Например, в проекте по разработке государственных образовательных стандартов [13].

Помимо вышеописанных, существуют как более мелкие, так и более крупные проекты (например, категоризация продаваемых товаров и их характеристики на сайте Amazon.com [14]). Все это свидетельствует, что онтологии востребованы во многих областях человеческой деятельности.

Семантическая паутина

Объем информации, содержащейся в Интернете, растет гигантскими темпами. По данным поисковой машины Yandex в феврале 2004 г. насчитывалось 428,571,000 веб-документов в русской части Интернета, в апреле 2005 г. — 1,555,937,000, а в декабре этого же года — 2,538,893,000. Такой лавинообразный рост — свидетельство того, насколько быстро растут терабайты информации в русскоязычном секторе Интернета. Следовательно, алгоритмы поиска и анализа данных все более усложняются. Поисковая машина выдает на запрос тысячи веб-страниц, в которых найти нужные — большая проблема. Должен произойти какой-то качественный

скачок в развитии Интернета. *Семантическая паутина* — результат такого скачка. Если раньше компьютер мог читать документы, то теперь он может их понимать. Следовательно качество поиска возрастает. Идеолог и создатель этой идеи Тим Бернс-Ли утверждает, что «**семантический веб разрабатывает языки для выражения информации в форме, доступной для машинной обработки**» [15]. По сути, Интернет останется снаружи почти таким же, каким был раньше для людей, но кардинально изменится для машин, которые смогут разбираться в «бездонной помойке», делая её более полезной для людей.

В настоящее время семантический веб всё быстрее набирает обороты, но развит пока довольно слабо, поскольку активная работа над ним началась совсем недавно. Тем не менее, есть уже первые результаты — *Extensible Markup Language* (XML) или расширяемый язык разметки [16]. Данный язык — универсальный синтаксический фундамент, на котором могут строиться решения проблем представления данных, а также отношений между ними. Помимо XML существуют и более молодые, как следствие менее распространенные технологии: XHTML [17], XSLT [18], RDF, OWL и другие. Использование разработок в области XHTML и XSLT выходит за рамки данной статьи, а языки описания онтологий RDF и OWL рассмотрены ниже.

Языки описания онтологий

Для реализации широкого диапазона онтологий требуются мощные и гибкие языки представления. **Один из ключевых моментов в создании онтологической модели — выбор конкретного языка (*ontology specification language*). Таких языков довольно много и разобраться в том, какое место они занимают в разработке онтологий, проблематично. Язык должен:**

- ✧ быть достаточно выразительным и удобным;
- ✧ помочь пользователю абстрагироваться от низкоуровневых проблем, позволяя тем самым сосредоточиться на первостепенной задаче — проектировании онтологии.

Языки описания онтологий классифицируют следующим образом:

- ✧ *языки, традиционно используемые сообществом создателей онтологий* (*ontology community*). Это DOGMA (Developing Ontology-Grounded Methods and Applications) [19], KIF (Knowledge Interchange Format) [20], OCML

(Operational Conceptual Modelling Language) [21], LOOM [22], CycL [23], F-Logic [24] и OKBC (Open Knowledge Base Connectivity) [25];

- ✧ *языки, созданные в контексте Интернет-среды и рекомендованные консорциумом W3C: XML, RDF, RDFS, DAML [26], OIL [27], OWL. Последний наиболее поздний из всех созданных языков.*

Рассмотрим некоторые языки из представленных классов подробнее.

Язык **LOOM** — используется для создания экспертных систем и других интеллектуальных приложений — основан на дескриптивной логике и относится к KL-ONE-подобной группе языков [28]. Он значительно выразительнее и мощнее своих предшественников — языков, основанных на фреймах — OCML, OKBC и F-Logic. Его важное преимущество — возможность задавать множество ограничений на созданные концепты.

В рамках проекта семантической интерпретации информационных ресурсов Интернета (семантический веб) **предложен стандарт описания метаданных о документе RDF (Resource Description Framework), использующий XML-синтаксис.** RDF использует базовую модель данных «объект — атрибут — значение» и способен сыграть роль универсального языка описания семантики ресурсов и взаимосвязей между ними. Важная особенность стандарта — расширяемость: он позволяет задать структуру описания источника, используя существующие и создавая новые классы, свойства, типы, коллекции.

RDF Schema (RDFS) — стандарт, предложенный по инициативе W3C для представления онтологических знаний — относится к онтологии представления и специфицирует множество допустимых схем данных. RDFS предоставляет хорошие базовые возможности для описания словарей типов предметных областей.

DAML+OIL — семантический язык разметки web-ресурсов, расширяющий стандарты RDF и RDF Schema. Последняя версия DAML+OIL обеспечивает богатый набор конструкций для создания онтологии и разметки информации таким образом, чтобы их могла читать и понимать машина. Данный стандарт спецификации возник на базе DAML (DARPA Agent Markup Language) и European Commission OIL (Ontology Inference Layer), которые разработаны для поддержки процесса обмена знаниями и интеграции знаний. Онтология DAML+OIL состоит из: заголовков (*headers*), элементов классов (*class elements*), элементов свойств (*property elements*), экземпляров (*instances*).

OWL (web ontology language) — язык представления веб-онтологий— средство описания веб-документов и веб-приложений; расширяет возможности XML, RDF, RDF Schema и DAML+OIL. Этот проект предусматривает создание мощного механизма семантического анализа.

Управление неструктурированной информацией в Интернете невозможно без поддержки мощных инструментов. Чтобы структурировать эти данные, компьютерным агентам требуются машиночитаемые описания содержимого документов и характеристик доступных веб-ресурсов. Такого рода мета данные должны присутствовать в любых источниках данных. Язык OWL и является языком, представляющим возможности такого описания. **Онтологии OWL — последовательности аксиом и фактов, а также ссылок на другие онтологии — содержат компоненту для записи авторства и другой подробной информации.**

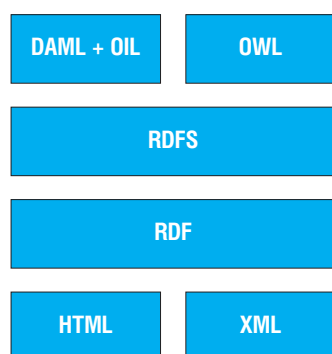


Рис. 3. Иерархия языков для web-среды

Сами по себе языки представления онтологий не были бы так сильно востребованы, **если бы не возникало необходимости автоматически обрабатывать онтологии, наполнять их содержимым и выполнять запросы к хранилищам онтологий.** Наиболее популярными среди языков запросов к RDF-хранилищам являются языки RDQL [29], SPARQL [30]. В настоящий момент в Stanford Knowledge Systems Laboratory разрабатывается язык запросов к OWL — OWL-QL [31].

Инструментальные средства обработки онтологий

В течение последних семи лет появилось много инструментов для разработки и использования онтологий. Инструментальная поддержка одновременно важна как для процесса разработки (построение онтологии, её комментирование, объединение

с другими онтологиями и т.д.), так и для использования онтологических моделей в различных областях, таких как электронная коммерция, управление знаниями, семантический веб и др.

При создании онтологий целесообразно пользоваться подходящими инструментами. Какой тип инструментов наиболее удобен для построения онтологий? В каком формате хранятся онтологии в выбранном инструменте? В какие форматы можно экспортировать и из каких форматов можно импортировать онтологии? При выборе инструментов для работы с онтологиями необходимо ответить на эти и многие другие вопросы.

Общие характеристики редакторов

Модель знания

Под формализмом понимается теоретическая часть, лежащая в основе способа представления онтологических знаний. Примерами формализмов могут быть вышеописанные логика предикатов первого порядка, дескриптивная логика и фреймы. Помимо этого, существуют другие способы представления онтологий, например, концептуальные графы. Формализм влияет на внутреннюю структуру данных, но также может существенно воздействовать на формат представления и даже на пользовательский интерфейс [3].

Формат представления онтологии задает вид хранения и способ передачи онтологических описаний. Формат — это языки представления онтологий: RDFS, OWL, DAML+OIL и проч.

Архитектура программного обеспечения

Различают трехуровневую, многоуровневую архитектуру и архитектуру типа клиент-сервер.

Трехуровневая архитектура предполагает наличие трех компонентов приложения: клиентское приложение, подключенное к серверу приложений, использующий сервер базы данных [32].

Многоуровневая архитектура приложения разделяет пользовательские сервисы, прикладные сервисы и сервисы данных.

В **архитектуре клиент-сервер** прикладные и пользовательские сервисы реализованы на клиентской рабочей станции, а данные централизованно хранятся на сервере. В этой модели клиенты подключаются непосредственно к серверу, на все время работы приложений [33].

Функциональность

Важная характеристика программного средства — его функциональность, т.е. множество сценариев его использования. В базовый набор функций входят:

- ✧ сохранение проекта в нужном формализме и формате (экспорт);
- ✧ возможность чтения различных форматов и формализмов (импорт);
- ✧ редактирование метаданных проекта (к примеру, настройка формы редактирования и представления данных);
- ✧ создание и редактирование онтологий.

В дополнительный набор функций могут входить, например, возможность поддержки какого-либо языка запросов, анализ целостности, возможность представления онтологии в виде графа с его последующим анализом и редактированием, извлечением информации и проч.

Инструменты

Инструменты работы с онтологиями сгруппированы в табл. 1. Наиболее мощные из них:

- ✧ редакторы общего назначения Protégé и Ontolingua [34];
- ✧ редактор, ставящий своей целью развитие семантического веба Altova SemanticWorks 2008 [35];
- ✧ плагин PROMPT к Protégé, позволяющий объединять и группировать онтологии [45];
- ✧ инструмент SmartTools [36], позволяющий конструировать, управлять, делить и критиковать модели знаний, представленных в форме концептуальных карт (concept maps).

Наибольший интерес представляет редактор Protégé. Это свободно распространяемая Java-программа, созданная лабораторией медицинской информатики школы медицины Стенфордского университета. Первоначально Protégé предназначался для моделирования в области медицины, но теперь используется и в других предметных областях. Поскольку этот редактор — продукт с открытым исходным кодом, для него написано множество плагинов. Также для него имеется довольно обширная база знаний, а объединение людей вокруг этого продукта на официальном сайте Protégé [37] насчитывает более 80 тысяч человек. В настоящее время данный продукт — наиболее перспективным редактором онтологий из всех существующих.

Protégé предназначен для построения (создания, редактирования и просмотра) онтологий прикладных областей. Инструментальные средства, входящие в состав продукта, включают редактор онтологий, позволяющий проектировать онтологии, создавая их иерархическую структуру абстрактных и конкретных классов. На основе сформированной

онтологии Protege позволяет генерировать формы получения знаний для введения экземпляров классов и подклассов. [38].

В рамках проекта на основе Protégé создана линейка из двух продуктов: Protégé-Frames и Protégé-OWL. Protégé-Frames использует фреймовую модель представления знаний OKBC (*Open Knowledge Base Connectivity*). Это позволило адаптировать его и для редактирования моделей предметной области, представленных не в OWL, а в других форматах (UML, XML, SHOE [39], DAML+OIL, RDF и RDFS и т. п.). В основе Protege-OWL — дескриптивная логика, что позволяет использовать его в контексте семантического веба.

Редактор **Protégé-OWL** является расширением Protégé, позволяющим пользователю:

- ✧ открывать и сохранять OWL и RDF онтологии;
- ✧ редактировать и, в последствии, визуализировать классы, свойства и правила SWRL (*Semantic Web Rule Language*) [40];
- ✧ определять логические характеристики класса как выражения OWL;
- ✧ выстраивать механизм рассуждений таким образом, как это сделано в рамках классификаторов дескриптивной логики.

Гибкая архитектура **Protégé-OWL** позволяет легко конфигурировать и расширять редактор при помощи плагинов. Protégé-OWL тесно интегрирован с каркасом Java приложений Jena. Protégé-OWL имеет открытый исходный код API, написанный на Java, с помощью которого можно разрабатывать всевозможные компоненты или создавать различные сервисы и ресурсы семантического веба [41].

Редактор **Protégé-Frames** предоставляет полноценный пользовательский интерфейс и базу знаний (*knowledge server*) для разработки и хранения онтологий. В редакторе существует возможность настройки форм для ввода данных об объектах класса. Возможности редактора:

- ✧ разнообразие элементов пользовательского интерфейса можно видоизменять так, чтобы пользователь посредством запросов к серверу мог как получать, так и отправлять данные. При этом информация, которую вводит пользователь, понятна как человеку, так и компьютеру;
- ✧ архитектура продукта построена таким образом, чтобы в программу можно было встраивать дополнительные компоненты. Примерами таких компонент являются графические компоненты, включающие графики и таблицы, медиа-компоненты, включающие звуки, изображения и видео, различные форматы хранения данных (RDF, XML, HTML и т.д.);

Название программы	Разработчик	Версия	Дата последнего релиза	Доступность	Расширяемость	Форматы импорта/экспорта онтологий	Представление в виде графа	Извлечение информации	Ссылки
Редакторы									
Protege	Stanford Medical Informatics	3.3.1	Август 2007	Open source	+	RDF, RDFS, DAML+OIL, XML, OWL, Clips, UML, FLogic, Java, XMI	Да	Нет	www.protege.stanford.edu
OntoEdit	Ontoprize	2.6.6	Март 2004	Свободная лицензия	+	OXML, RDF(S), DAML+OIL, FLogic	Да	Нет	www.ontoprize.de/products/ontoedit
Ontolingua	KSL, Stanford University	1.0.650	Октябрь 2002	Свободный доступ	-	Ontolingua, KIF, CML, DL, LO-OM, CLIPS, Epi-Kit, Prolog, IDL	Нет	Нет	www.ksl.stanford.edu/software/ontolingua
Apollo	Knowledge Media Institute of Open University, UK	1.0	Август 2004	Open source	+	RDF, XML, Meta, OCML		Нет	www.apollo.open.ac.uk m.koss@open.ac.uk
Semantic Works	Altova	2008	2008	Лицензия	-	RDF, XML, OWL	Да	Нет	www.altova.com/products/semanticworks/semantic_web_rdf_owl_editor.html
Link Factory	Language and Computing nv	3.3 19-Nov-	2003	Лицензия	-	XML; RDF(S); DAML+OIL/OWL	Нет, но возможно посмотреть графическое представление с помощью Java Beans.	Да	www.landc.be info@landc.be
Объединение и отображение онтологий									
PROMPT Плагин к Protege	Noy and Musen	2.4.8	Июнь 2005	Open source	-	см. Protege	см. Protege	см. Protege	www.protege.stanford.edu/plugins/prompt
Onto Merge	Yale University	0.1 alpha	Апрель 2002	-		RDF, DAML+OIL, OWL, WSDL, XML	Нет	Нет	www.cs.yale.edu/homes/dvm/daml/ontology-translation.html
Visio 2007	Microsoft Corp.	12.0.4518	Декабрь 2006	Лицензия	+	XML	Да	Нет	www.microsoft.com
Chimaera	KSL, Stanford University	0.1.45	Август 2003	Свободный доступ	-				www.ksl.stanford.edu/software/chimaera/
Извлечение информации									
Сmap Tools	Institute for Human and Machine Cognition, University of West Florida	4.11	Июль 2007	Свободный доступ	-	Отсутствуют	Да	Да	www.ihmc.us
OntoX	Technion - Israel Institute of Technology	1.0	Май 2004	Свободный доступ	-	XML с поддержкой BizTalk; (RDF & OWL planned)	Только просмотр	Извлечение онтологий из веб-ресурсов с использованием элементов form и ссылок.	www.iew3.technion.ac.il/OntoBuilder/
VisualText Editor-Grammar KB	International Inc Text	1.7	Октябрь 2003	Свободный доступ	-	HTML, XML. Возможность загрузки в БД.	Да	Да	www.textanalysis.com
K-Infinity Knowledge Builder	Intelligent Views GmbH	2.0	Сентябрь 2003	Свободный доступ	+	RDFS и XM	Да	Да	www.i-views.de

✧ API, написанный на Java, предоставляет доступ к использованию и отображение онтологий, созданных с помощью редактора, другим приложениям и встроенным в Protégé-Frames плагинам [42].

Altova SemanticWorks 2008 — революционный продукт в области создания и обработки RDF/OWL онтологий [16] — единственный представитель в классе визуальных редакторов онтологий для семантического веба. Поскольку код генерируется автоматически согласно графическому виду создаваемой онтологии, пользователь может сфокусироваться на логической и семантической составляющей проектирования. По мере работы с программой пользователь может переключаться между графическим и текстовым видами редактора с целью просмотра и редактирования сгенерированного кода. Программный продукт имеет следующие возможности:

- ✧ графические RDF/RDFS/OWL редакторы;
- ✧ поддержка OWL Lite, OWL Full и OWL DL;
- ✧ специальные средства для просмотра больших онтологий;
- ✧ проверка документа на семантическую правильность.

Для использования данного продукта требуется лицензия.

Ontolingua. Кроме собственно редактора онтологии, система Ontolingua содержит сетевой компонент **Webster** для определения концептов; сервер, обеспечивающий доступ к онтологиям Ontolingua по протоколу OKBC; **Chimaera** — инструмент для анализа и объединения онтологий [38]. К сожалению, в настоящее время система больше не развивается. Большую часть продуктов для работы с онтологиями постигла та же участь. В 2004 г. программных средств, поддерживаемых и развиваемых, было около сотни, а к настоящему времени их осталось не более двадцати [44].

PROMPT — дополнение к системе Protégé, реализованное в виде плагина. PROMPT позволяет:

- ✧ сравнивать две версии похожих онтологий и на основании этого дополнять одну онтологию другой;
- ✧ объединять две онтологии в одну;
- ✧ извлекать некоторую часть онтологии с целью дальнейшего использования.

По двум онтологиям, которые надо объединить, PROMPT строит список операций (например, объединение терминов или их копирование в новую онтологию) и передаёт его пользователю, кото-

рый может выполнить одну из предлагаемых операций. Затем список операций модифицируется, создаётся список конфликтов и их возможных решений. Это повторяется до тех пор, пока не будет готова новая онтология.

Инструмент **ИНМС CmapTools**, разработанный ИНМС (Institute for Human and Machine Cognition), позволяет конструировать, управлять, делить и критиковать модели знаний, представленных в форме концептуальных карт (concept maps, c-map). Данная программа позволяет пользователям, помимо множества других возможностей, создавать собственные концептуальные карты на ПК и затем отсылать их на сервер (Cmap Server), соединять их с другими концептуальными картами на сервере, автоматически создавать веб-страницы из них, а также разрабатывать концептуальные карты синхронно с другими пользователями в сети интернет.

Заключение

В табл. 1 приведены сведения об основных и наиболее важных параметрах, могущих повлиять на решение о выборе инструмента. В настоящее время количество инструментов онтологического инжиниринга превышает сотню, но развивающихся и поддерживаемых продуктов очень мало. Поэтому в данной таблице рассмотрены только инструменты, которые поддерживаются и развиваются в настоящее время, а также наиболее мощные средства, интересные с научной точки зрения, поддержка которых, к сожалению, не ведётся.

Проблемы выбора инструмента для работы с онтологиями как таковой не существует, поскольку в каждой из групп есть явные фавориты. Например, наиболее удобные редакторы онтологий — Protégé и Altova SemanticWorks 2008, а лидер в области объединения и отображения онтологий — плагин PROMT к Protégé. ■

Список сокращений

ООП — объектно-ориентированное программирование.

АКО — A Kind OF — ссылка на фрейм, являющийся

API — Application Programming Interface — интерфейс прикладного программирования набор стандартных программных прерываний, вызовов процедур (функций, методов) и форматов данных, которые могут использовать прикладные программы для запроса и получения от операционной системы или программного интерфейса связанного с ними обслуживания.

Data Mining — технология анализа хранилищ данных, базирующаяся на методах искусственного интеллекта и инструментах поддержки принятия решений.

DAML — DARPA Agent Markup Language — язык разметки семантического веба.

DARPA — The Defense Advanced Research Projects Agency — агентство перспективных исследований МО США. В 1973 г. агентство создало сеть ARPAnet, которая впоследствии дала начало Интернету.

DOGMA — Developing Ontology-Grounded Methods and Applications.

GNOME — GNU Network Object Model Environment — свободная среда рабочего стола для UNIX-подобных операционных систем.

HTML — Hyper Text Markup Language, язык гипертекстовой разметки.

ID — Identifier — идентификатор.

KDE — K Desktop Environment — графический пользовательский интерфейс фирмы Corel.

KIF — Knowledge Interchange Format.

LOOM — язык и среда для создания интеллектуальных приложений.

OCML — Operational Conceptual Modelling Language.

OKBC — Open Knowledge Base Connectivity — прикладной интерфейс программирования для доступа к базам знаний систем представления знаний.

OMG — Object Management Group — группа управления объектами, некоммерческое объединение, разрабатывающее стандарты для создания интероперабельных (платформно-независимых) приложений уровня предприятий.

RDF — Resource Description Framework — технология RDF разработана W3C для описания содержи-

мого сайтов, более глобально — для построения семантической сети Интернета.

RDFS — RDF Schema — стандарт, предложенный по инициативе W3C для представления онтологических знаний.

RDQL — RDF query language — язык запросов к RDF.

OWL — Web Ontology Language — язык онтологии для Интернета.

OWL-QL — OWL query language — язык запросов к OWL.

SHOE — Simple HTML Ontology Extensions — XML-совместимый язык представления знаний для Web.

SPARQL — SPARQL Protocol and RDF Query Language — разрабатываемый стандарт семантической паутины, в настоящее время проходящий стандартизацию консорциума W3C. К настоящему времени уже существует несколько различных реализаций

SWRL — Semantic Web Rule Language.

UML — Unified Modeling Language — унифицированный язык моделирования.

W3C — World Wide Web Consortium — организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины.

XHTML — Extensible HTML. Язык XHTML, разработан W3C и предназначен для поддержки XML в Web-страницах.

XML — eXtensible Markup Language, расширенный язык разметки

XSL — Extensible Stylesheet Language — расширяемый язык таблиц стилей.

XSLT — Extensible Stylesheet Language Transformations часть спецификации XSL, задающая язык преобразований XML-документов. ■

Список литературы

1. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2001.
2. Tom Gruber. What is an Ontology? // Available at: www.ksl.stanford.edu/kst/what-is-an-ontology.html.
3. Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д. Онтологии и Тезаурусы. Казань, 2006.
4. Мильчин А. Э. Издательский словарь-справочник. М.: ОЛМА-Пресс, 2006.
5. Resource Description Framework (RDF) // Available at: www.w3.org/RDF/.
6. Web Ontology Language (OWL) // Available at: www.w3.org/2004/OWL/.
7. Natalya F. Noy, Deborah L. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001.
8. Gao Shu, Omer F. Rana, Nick J. Avis, Chen Dingfang. Ontology-based semantic matchmaking approach // Advances in Engineering Software 38 (2007). P. 59–67.
9. The Dublin Core Metadata Initiative // dublincore.org/.
10. GNOME // www.gnome.org/.
11. KDE // www.kde.org/.
12. The Enterprise Ontology // www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html.
13. Никитин В.В. Информационно-методическое обеспечение. Москва: МАКС Пресс, 2006.

14. Amazon.com // amazon.com/.
15. Transcript of Tim Berners-Lee's talk to the LCS 35th Anniversary celebrations, Cambridge Massachusetts, 1999/April/14 // Available at: www.w3.org/1999/04/13-tbl.html.
16. Extensible Markup Language (XML) // www.w3.org/XML.
17. XHTML2 Working Group Home Page // www.w3.org/MarkUp/.
18. The Extensible Stylesheet Language Family (XSL) // www.w3.org/Style/XSL/.
19. DOGMA // en.wikipedia.org/wiki/DOGMA.
20. Knowledge Interchange Format (KIF) // www.ksl.stanford.edu/knowledge-sharing/kif/.
21. OCML Operational Conceptual Modelling Language // kmi.open.ac.uk/projects/ocml/.
22. MacGregor R. Inside the LOOM classifier // SIGART bulletin. 1991. № 2. P. 70–76.
23. CycL // en.wikipedia.org/wiki/CycL.
24. F-Logic // en.wikipedia.org/wiki/F-Logic.
25. Open Knowledge Base Connectivity // www.ai.sri.com/~okbc/.
26. DAML // www.daml.org/about.html.
27. OIL // www.ontoknowledge.org/oil/.
28. Wikipedia // en.wikipedia.org/wiki/KL-ONE.
29. RDF Query Survey // www.w3.org/2001/11/13-RDF-Query-Rules/.
30. SPARQL Query Language for RDF // www.w3.org/TR/rdf-sparql-query/.
31. OWL-QL Project for the Stanford Knowledge Systems Laboratory // www.ksl.stanford.edu/projects/owl-ql/.
32. Википедия. Свободная энциклопедия // www.wikipedia.org.
33. Словарь по естественным наукам. Глоссарий.py // www.glossary.ru.
34. Ontolingua // www.ksl.stanford.edu/software/ontolingua/.
35. Altova SemanticWorks 2008 // www.altova.com/products/semanticworks/semantic_web_rdf_owl_editor.html.
36. IHMC CmapTools // cmap.ihmc.us/download/.
37. The Protégé Ontology Editor and Knowledge Acquisition System // protege.stanford.edu/.
38. Гладун А.Я., Рогушина Ю.В. Онтологии и в корпоративных системах // Корпоративные системы. 2006. №1.
39. SHOE // www.cs.umd.edu/projects/plus/SHOE/.
40. SWRL: A Semantic Web Rule Language Combining OWL and RuleML // www.w3.org/Submission/SWRL/.
41. Protege OWL // Available at: protege.stanford.edu/overview/protege-owl.html.
42. Protege Frames // Available at: protege.stanford.edu/overview/protege-frames.html.
43. Altova SemanticWorks 2008 // www.altova.com/products/semanticworks/semantic_web_rdf_owl_editor.html.
44. Michael Denny. Ontology Building: A Survey of Editing Tools // Available at: www.xml.com/pub/a/2002/11/06/ontologies.html.
45. Prompt // Available at: protege.cim3.net/cgi-bin/wiki.pl?Prompt.



ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ – ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

представляет свои периодические издания

ВОПРОСЫ ОБРАЗОВАНИЯ
ЕЖЕКВАРТАЛЬНЫЙ НАУЧНО-ОБРАЗОВАТЕЛЬНЫЙ
ЖУРНАЛ

Издается с 2004 г.

Главный редактор –
Ярослав Иванович Кузьминов

Издание освещает теоретические и прикладные проблемы российского образования. Содержит статьи ведущих российских и зарубежных ученых и экспертов. В каждом номере – дискуссии, рецензии, обзоры публикаций и законодательства в области образования.

Каталог Агентства «Роспечать» – индекс 82950 Объединенный
каталог «Пресса России» – индекс 15163

Координаты редакции:
101990 Москва, ул. Мясницкая, 20, офис 308
E-mail: edu.journal@hse.ru
Тел: (495) 628-5102, 621-8523

Modern research results and methodological principles of knowledge representation

H. Abdulrab,

Professor of computer science, LITIS laboratory, INSA (Rouen, France).

E. Babkin,

Associate Professor, Faculty of Business Informatics and Applied Mathematics, State University – Higher School of Economics (Nizhny Novgorod). LITIS laboratory, INSA (Rouen, France).

This article concerns the modern methods of knowledge representation which can be effectively applied in modern intellectual information system. The major attention is given to very promising concept of ontology and corresponding formal and software engineering methods for ontology manipulation.

Современные результаты и методологические принципы в области представления знаний

Х. Абдулраб,

профессор информатики, сотрудник лаборатории LITIS, Институт прикладных наук (Руан, Франция).

Э.А. Бабкин,

к.т.н., доцент кафедры информационных систем и технологий, факультет бизнес-информатики и прикладной математики НФ ГУ ВШЭ. Сотрудник лаборатории LITIS, Институт прикладных наук (Руан, Франция).

В статье рассматриваются современные методы представления знаний для автоматизированных интеллектуальных систем. Эта область теоретической информатики стала чрезвычайно актуальной в связи с задачами поиска информации в Интернете и проблемами интеграции сложных программных систем. Основное внимание в статье уделяется концепции онтологии, определяющей терминологический словарь предметной области и логическую микро-теорию. Для представления и обработки онтологий в составе программных систем в статье анализируются формальные и инженерные методы на основе аппарата формальной логики (дескриптивные логики) и парадигмы удовлетворения ограничений (система Alloy). В качестве формальной основы для представления и преобразования онтологий также рассматривается перспективная теория информационных потоков Дж. Барвиза и Д. Селигмана.