



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

МЕТОДЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ МОДЕЛИРОВАНИЯ

Коршиков Д.Н., Лахин О.И., Носкова А.И., Юрыгина Ю.С.

*ООО «НПК «Разумные решения»,
г. Самара, Россия*

korshikov@smartsolutions-123.ru, lakhin@smartsolutions-123.ru,

noskova@smartsolutions-123.ru, yurygina@smartsolutions-123.ru

В работе приводятся методы представления знаний в виде предметных онтологий с использованием базиса языка OWL и его расширением до мета-онтологии Аристотеля, дается краткий обзор инструмента создания онтологий, а также описываются функции конструктора онтологий и решаемые задачи.

Ключевые слова: онтология; мета-онтология Аристотеля; семантическая сеть; RDF; OWL; мультиагентные системы; моделирование.

Введение

Требования к интеллектуальным системам управления ресурсами из года в год становятся все жестче, что обусловлено усложнением процессов современного рынка, неопределенностью и динамикой изменения спроса и предложения, возрастающей конкуренцией.

В условиях высокой неопределенности резко возросла ценность знаний специалистов в различных областях деятельности, а, следовательно, возникла потребность в их формализации, хранении и повторном использовании. Основной проблемой является то, что не все знания можно легко описать и запрограммировать, т.к. они обычно являются разнородными, многосвязными, часто как декларативными, так и процедурными, связанными не только иерархическими, но и сетевыми структурами. [Матюшин, 2014].

Поскольку сокращение трудоемкости разработки и сопровождения интеллектуальных программных систем является важной проблемой, то использование онтологий для управления знаниями является целесообразным [Шалфеева, 2011].

В статье рассматриваются методы представления знаний в виде онтологий, описываются особенности выбранного подхода, приводятся примеры решаемых задач.

1. Почему онтологии?

1.1. От классической системы к интеллектуальной

На самом начальном этапе создания системы всегда возникают вопросы: как именно она будет строиться, какой методологией руководствоваться, какие подходы использовать.

Классический подход, с одной стороны, позволяет использовать стандартные методологии, архитектуры и наборы готовых компонентов. Вся информация жестко структурирована и хранится в базах данных. Результат работы такой системы предсказуем и стабилен. С другой стороны, использование жестко закодированных данных и логики делает такие системы сложно расширяемыми и негибкими.

В свою очередь, интеллектуальные системы, в частности мультиагентные системы, использующие онтологии в качестве баз знаний, легко поддаются модификации при наличии такой необходимости. Онтологии также позволяют сильно расширить круг задач, решаемых интеллектуальной системой, в различных актуальных направлениях.

1.2. Спектр решаемых задач

Актуальность использования онтологий в качестве баз знаний легко оценить. Перечень задач, решаемых интеллектуальными системами на основе онтологий, достаточно велик. Рассмотрим основные из них:

- создание глобальной базы знаний путем интеграции различных онтологий, за счет широкого использования полисинонимии и ретранслируемости;
- решение задач, связанных с информационным поиском;
- решение задач, связанных с распознаванием образов;
- решение задач дедуктивного и индуктивного выводов и др.

Дополнительные преимущества онтологии предоставляют в вычислительном (сокращается время вычислений) и экономическом (сокращаются затраты на разработку) плане по сравнению с уже существующими решениями, основанными на классических подходах.

2. Методы и средства представления знаний

Онтология – это средство формализации знаний о предметной области в форме, упрощающей взаимодействие между пользователем и системой.

Термин «онтология» начал использоваться применительно к информационным технологиям сравнительно недавно, в начале 90х годов прошлого века. Но несмотря на это, уже существует достаточно большое количество различных форм и методов описания знаний в виде онтологий со своими достоинствами и недостатками. Также были выделены различные базисы, на которых специалисты могли бы основываться, описывая свои предметные области, так как одной из основных особенностей онтологического подхода является единство терминологии.

Различные формы представления знаний используются в системах, решающих разнообразные задачи, но ни одна из существующих форм не способна полностью покрыть потребности, возникающие при разработке интеллектуальных систем. В связи с этим возникла необходимость доработать существующие методы и определить свой базис для описания и хранения онтологий.

Для создания универсального метода было принято решение взять за основу общепринятые стандарты представления знаний. Согласно этим стандартам, онтологии можно классифицировать по целям создания на онтологии представления, онтологии верхнего уровня, онтологии предметных областей и прикладные онтологии. В настоящей работе более подробно рассматриваются первые два вида онтологий. Пользователь работает с онтологией верхнего уровня, которая автоматически программными средствами преобразуется в онтологию представления.

2.1. Онтологический базис

Для решения поставленных задач онтология представления описывается с помощью языка OWL. Это дает возможность использовать готовые инструменты, работающие с данными описанными

с помощью OWL, а также упрощает интеграцию со сторонними системами. Формат представления данных - RDF (т.е. технически модель данных, с которой работает система, представляет RDF граф).

RDF - это модель для описания объектов. Идея RDF состоит в том, чтобы одним простым способом можно было описать различные утверждения в таком структурированном виде, чтобы его могли обрабатывать компьютерные программы.

RDF – это формат представления информации, организованной в виде графов или сетей. RDF определяет управляемые графы связей, представленные тройками (триплетами), имеющими общий вид «субъект – предикат – объект». Множество RDF-утверждений образует ориентированный граф, в котором вершинами являются субъекты и объекты, а рёбра помечены предикатами.

OWL является расширением RDF-понятий и представляет собой вычислительный логико-ориентированный язык описания онтологий для семантической паутины. В основе языка – представление действительности в модели данных «объект – свойство». Структура OWL-онтологии включает: классы, таксономические отношения между классами, свойства (бинарные отношения), экземпляры классов и отношений.

Для расширения возможностей языка OWL при создании онтологий верхнего уровня в качестве онтологического базиса вводится мета-онтология Аристотеля, основными концептами которой являются «объект», «свойство», «процесс», «отношение» и «атрибут». Согласно этой модели:

- существуют объекты, которые обладают свойствами и характеризуются состояниями;
- свойства выражают способность объектов вступать в процессы взаимодействия на основе законов мира;
- отношения между объектами могут отражать структурные, функциональные, временные или любые другие виды связей;
- чтобы выполнить действие над объектом, необходимо соблюдение определенных условий, которые задаются свойствами и отношениями;
- действия (процессы) изменяют состояния объектов, их свойства и отношения;
- свойства, отношения и действия характеризуются значениями атрибутов;
- атрибуты объекта/отношения являются качественной или количественной характеристикой понятия;
- правила являются обобщенными понятиями для формализованных условий вида «если-то» (предикатов) и высказываний (утверждений, аксиом, фактов) [Скобелев, 2012].

Преимуществом использования такого онтологического базиса является возможность описывать процессы и действия, а значит, помимо физических объектов создавать и процедурные

онтологии, что актуально при решении задач моделирования различных бизнес-процессов.

2.2. Конструктор онтологий

Описываемый базис до сих пор не использовался ни в одном из существующих редакторов онтологий в качестве основного, поэтому параллельно с разработкой и описанием методов создается новый инструмент для работы с онтологиями – Конструктор Онтологий, который учитывает все указанные выше особенности. Его основными функциональными возможностями являются [Вакурина, 2014]:

- создание и редактирование онтологий;
- отображение онтологий в различных видах;
- хранение и использование онтологий.

2.2.1. Создание и редактирование онтологий

Пользователь может оценить преимущества выбранного подхода уже на этапе создания онтологии. Концепция OWL позволяет не только создавать онтологию с нуля, но и описывать новые знания, основываясь на уже существующих онтологиях, таким образом постоянно наращивая объемы информации, не дублируя ее. Реализация многопользовательского режима еще более упрощает эту работу. С этой точки зрения, задача создания единой глобальной базы знаний, о которой говорилось ранее, уже не кажется неразрешимой.

Следующим преимуществом работы в создаваемом Конструкторе является то, что пользователь может при необходимости совершать манипуляции с онтологиями, используя сравнительно простой язык взаимодействия DSL (Domain Specific Language). С его помощью можно вводить различные условия, правила, аксиомы и т.д.

Знания в онтологии могут создаваться не только пользователем, но и системой на основании рассуждений. За это отвечает модуль логического вывода. На данном этапе происходит тесное переплетение онтологического подхода и мультиагентных технологий, так как модуль логического вывода для решения задач классификации использует агентную модель.

Ниже рассматриваются основные принципы работы агентной модели для решения задач логического вывода:

- каждой сущности (экземпляру) и каждому классу онтологии назначается программный агент;
- задача агента - обладать всеми знаниями об этой сущности: каким классам принадлежит, в каких процессах участвует и т.д.;
- знания разделяются на факты и логические выводы.

Для получения знаний агент применяет свои аксиомы, а также запрашивает знания у других агентов. Набор аксиом агента расширяем, что позволяет наделять агента новыми знаниями.

Описание поведения агента:

- каждый агент хранит факты о себе;

- каждый агент делает логические выводы о себе и хранит их;
- каждый агент уведомляет других агентов о своих изменениях, на основе которых они делают новые выводы о себе;
- каждый агент по запросу предоставляет информацию о себе для других агентов;
- сообщения с фактами и уведомления об изменениях содержит информацию, о добавляемых и удаляемых утверждениях;
- при удалении факта, все логические выводы, которые основываются на нем, становятся недействительными.

Преимущества такого подхода:

- при появлении новых фактов будут проверены только те аксиомы, которые напрямую или опосредовано, связаны с этим фактом («локальные возмущения» в сцене);
- знания можно хранить децентрализованно, что делает систему легко расширяемой;
- архитектура системы позволяет распараллелить решение задачи.

Однако есть и недостаток: при расширении онтологии и увеличении количества аксиом переговоры разрастаются. Но так как архитектура системы позволяет распараллелить решение задачи, то данная проблема решается расширением аппаратных мощностей.

Как упоминалось ранее, очевидной особенностью предлагаемого подхода является возможность описания процедурной онтологии. Принцип ее создания схож с принципом создания онтологии физического мира, но вместо классов объектов вводятся классы процессов и действий. В рамках описания процесса необходимо определить:

- набор участников процесса и условия их участия;
- тело процесса, которое описывает изменения, вносимые в сцену данным процессом при моделировании.

При описании участника процесса указывается его принадлежность к описанному ранее классу объекта, а также описываются некоторые правила, которые связывают его с другими участниками процесса.

Для описания изменений, вносимых в сцену данным процессом, используется язык DSL. Ниже приведены примеры команд языка DSL, которые могут использоваться (вначале имя команды, в скобках тип передаваемых параметров):

- создать объект;
- удалить объект;
- задать атрибут;
- создать отношение;
- удалить отношение;
- запустить процесс;
- остановить процесс.

Процессы могут состоять из других процессов, в этом случае в теле процесса указывается набор процессов, из которых он состоит.

Действия и процессы описываются практически одинаково, за исключением того, что в действии из участников должен быть выделен субъект, который может выполнить это действие.

На основании описанных в онтологии классов процессов и действий можно строить цепочки – бизнес-процессы, которые в свою очередь могут использоваться для описания алгоритмов действий, описывающих некоторые регламенты и правила.

2.2.2. Отображение онтологий

Для того, чтобы наглядно отображать созданные онтологии в различных разрезах, пользователю предоставляется возможность просмотра созданных онтологий в нескольких вариантах: табличная форма, словарь понятий и карточка концепта, семантическая сеть, алгоритм действий (в случае бизнес-процесса). Каждое из представлений обладает своими особенностями и может иметь разные варианты использования в зависимости от потребностей пользователя.

2.2.3. Использование онтологий для моделирования и решения прикладных задач

В рамках решения конкретных прикладных задач вначале создается онтология, описывающая статическую часть предметной области: классы объектов, атрибуты, отношения. Далее описывается динамическая часть: классы процессов, действий, бизнес-процессы, события. На основании онтологии создается набор моделей, которые описывают непосредственно объекты решаемой задачи, на данном этапе определяются значения атрибутов объектов, отношения между ними. После того как все необходимые модели готовы, на основании них создается сцена, которая используется для решения прикладных задач. Агенты, отслеживают изменения в сцене и выполняют заданное им поведение. Ниже приведены основные задачи агентов в рамках моделирования:

- решают задачи классификации того или иного объекта в сцене на основании описания классов и правил принадлежности к ним;
- находят участников процессов, запускают процессы на выполнение, в случае если все участники найдены, и условия взаимного участия удовлетворены;
- останавливают процессы, если прекращаются выполняться условия и ограничения, накладываемые на участников процесса.

В процессе работы со сценой средствами DSL языка можно управлять сценой, добавляя новые объекты, отношения между ними и выполнять другие команды. Для управления DSL-командами используется специализированный инструмент «коммандер», который предоставляет удобный пользовательский интерфейс для ввода команд. Контекстная подсказка помогает ориентироваться пользователю и оперативно вводить необходимые

изменения, меняя при необходимости не только сцену, но и непосредственно саму онтологию. Таким образом, знания и логика работы системы может меняться на лету, что является большим преимуществом рассматриваемого подхода, и может использоваться как непосредственно во время эксплуатации системы, так и при предварительной подготовке знаний системы, используемых при решении прикладных задач.

Заключение

В данной статье рассмотрены основные методы и принципы построения онтологий, рассмотрены методы представления знаний для решения задач моделирования. Выполнен краткий обзор функциональности Конструктора онтологий, который является средой разработки онтологий и предоставляет средства моделирования. Рассмотренные методы могут использоваться для описания любой предметной области и решения различных прикладных задач.

Библиографический список

- [Матюшин, 2014] Матюшин, М.М. Методы и средства построения онтологий для визуализации связанных информационных объектов произвольной природы в сложных информационно-аналитических системах / М.М. Матюшин [и др.] // Информационно-управляющие системы. – 2014. – № 2(69). – С.9-17.
- [Шалфеева, 2011] Шалфеева, Е.А. Возможности использования онтологий при разработке и сопровождении программных систем / Е.А. Шалфеева // Владивосток : ИАПУ ДВО РАН, 2011. – с.16. – 80 экз.
- [Скобелев, 2012] Скобелев, П.О. Онтологии деятельности для ситуационного управления предприятием в реальном времени / П.О. Скобелев // Онтология проектирования. – 2012 - №1(3). – С.26-48.
- [Вакурина, 2014] Вакурина, Т.Г. Онтология российского сегмента Международной космической станции и ее практическое использование в интеллектуальных аэрокосмических приложениях / В.В. Котеля, О.И. Лахин, М.М. Матюшин, П.О. Скобелев // Материалы IV Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» Open Semantic Technologies for Intelligent Systems (OSTIS-2014). – Минск: БГУИР, 20 – 22 февраля 2014 г. – С. 221-226.

METHODS OF KNOWLEDGE REPRESENTATION TECHNIQUES FOR USE IN MODELING

Korshikov D, Lakhin O, Noskova A,
Yurygina Yu

Software Engineering Company Smart Solutions,
Samara, Russian Federation
korshikov@smartsolutions-123.ru,
lakhin@smartsolutions-123.ru,
noskova@smartsolutions-123.ru,
yurygina@smartsolutions-123.ru

Knowledge representation techniques as a top-level ontology using OWL language and meta-ontology of Aristotle are listed, a brief overview on developing tools for creating ontologies are made, and some problems solved by using the proposed methods are discussed in this article.