УДК 002.53:004.89

РАБОТА С ОНТОЛОГИЧЕСКОЙ МОДЕЛЬЮ ОРГАНИЗАЦИИ НА ОСНОВЕ ДЕСКРИПТИВНОЙ ЛОГИКИ

А.Ф. Тузовский

Институт «Кибернетический центр» ТПУ
Томский научный центр СО РАН
E-mail: TuzovskyAF@kms.cctpu.edu.ru

Рассмотрена последовательность преобразований онтологической модели в модель дескриптивной логики с применением системы логического вывода. В качестве системы поддержки работы с моделью дескриптивной логики используется система логического вывода RACER. Приводится способ проверки правильности онтологии на основе доказательства запроса на выполнимость модели дескриптивной логики.

Введение

В процессе разработки систем управления знаниями (СУЗ) разрабатываются онтологические модели знаний организации, которые включают множество основных понятий и отношений между ними. Для использования в СУЗ такие модели должны быть описаны на некотором формальном языке, который удобен для программной реализации. Запись на формальном языке позволяет выполнять однозначное описание онтологий, а также позволяет доказывать их согласованность и обрабатывать различные запросы к ним. Такие доказательства не только гарантируют проверку одной онтологии, но и позволяют выполнять доказательство над набором онтологий, например, проверить логическую согласованность двух или более онтологий. В настоящее время существуют разные подходы к работе с онтологиями. В данной статье предлагается выполнять работу с онтологиями с использованием теории дескриптивных логик [1] и системы логического вывода.

Дескриптивные логики

Дескриптивные логики (ДЛ) являются общим классом логик специально спроектированных для моделирования словарей [2]. Они определяют формальный язык для *понятий* и отношений (называемых *ролями*) вместе с теорией доказательства. ДЛ является языком выражения утверждений о фактах (т. е., утверждений о том, что является *истинным*) и запросов к ним, включая выполнимость (*satisfiability*) и включение (*subsumption*). Элементами этого языка являются: *понятие*, которое представляет класс, категорию или сущность; *роль,* которая является бинарным отношением между понятиями. Кроме этого в ДЛ входят *конструкторы (операции)* для понятийных выражений, включающие конъюнкцию, дизъюнкцию и определение отношений.

Системы, построенные с помощью дескриптивной логики, используют для создания *Баз Знаний* (БЗ), которые представляются в виде пары *Tbox* и *Abox*: *Tbox* (*terminological knowledge*) это набор утверждений, описывающих набор классов, их свойства и отношения между ними (интенсиональные знания); *Abox* (*assertional knowledge*) представляет собой реализацию схемы классов в виде набора экземпляров, содержащих утверждения об экземплярах понятий (экстенсиональные знания). По существу *Tbox* является моделью того, что *может* быть истинным, а *Abox* является моделью того, что в *настоящее время* является истинным.

ДЛ имеет формальную семантику, которая может использоваться для выполнения автоматического доказательства на основе БЗ. Автоматическое доказательство позволяет отвечать на такие запросы, как: выполнимость понятий (*satisfiability*) – может ли существовать некоторое понятие C; включение (*subsumption*) – является ли некоторое понятие C вариантом понятия D; согласованность (*consistency*) – является ли вся БЗ согласованной; и проверка экземпляра (*instance checking*) – является ли некоторое утверждение истинным.

Смысл ДЛ, кратко, может быть определен модельно-теоретической семантикой. ДЛ определяется интерпретацией $I=(\Delta^I, \bullet^I)$, где Δ^I является доменом рассуждений, а не пустое множество $\bullet^I$ является функцией интерпретации, которая сопоставляет каждое понятие с подмножеством Δ^I. Понятию C интерпретация сопоставляет множество C^I, а каждой роли R подмножество $(\Delta^I \times \Delta^I)$.

Понятия могут строиться из выражений. В состав операторов включается базовый набор операций (дополнение, объединение и пересечение множеств) и количественные ролевые ограничения. В таблице приведены операторы (конструкторы) и их семантика в ДЛ.

Аксиомы определяют отношения включения (*subsumption relations*) между понятиями ($A \sqsubseteq B$) (т. е. ($A \Rightarrow B$)) и ролями ($R \sqsubseteq S$). Первое утверждение определяет иерархию классов, а второе определяет иерархию отношений. Отметим, что аксиомы могут быть определены для выражений, например, $((B \wedge C \wedge \neg D) \sqsubseteq A)$.

В ДЛ любой запрос сводится к определению выполнимости (*satisfiability*) БЗ (т. е. ее логической согласованности). Таким образом, для ответа на запрос Q, требуется доказать, что «not Q» является не выполнимым (т. е. текущая БЗ не может предполагать not Q), что в этом случае приводит к заключе-

Таблица. *Конструкторы семантики для простой дескриптивной логики*

Элементы логики	Название	Синтаксис	Семантика	Символ
Понятия	Понятие	A	$A^I \subseteq \Delta^I$	
	Вершина (*Top*)	⊤	Δ^I	AL
	Основание (*Bottom*)	⊥	$\varnothing$	AL
	Пересечение	C⊓D	$C^I \cap D^I$	AL
	Объединение	C⊔D	$C^I \cup D^I$	U
	Отрицание	¬C	$\Delta^\wedge \backslash C^I$	C
	Ограничение на значение	∀R.C	$\{a \in \Delta^I \mid \forall b.(a,b) \in R^I \rightarrow b \in C^I\}$	AL
	Квантификатор существования	∃R.C	$\{a \in \Delta^I \mid \exists b.(a,b) \in R^I \wedge b \in C^I\}$	ε
	Неуточненное (безусловное) числовое ограничение	$\geq n$ R $\leq n$ R $= n$ R	$\{a \in \Delta^I \mid \lvert\{b \in \Delta^I \mid (a,b) \in R^I\}\rvert \geq n\}$ $\{a \in \Delta^I \mid \lvert\{b \in \Delta^I \mid (a,b) \in R^I\}\rvert \leq n\}$ $\{a \in \Delta^I \mid \lvert\{b \in \Delta^I \mid (a,b) \in R^I\}\rvert = n\}$	N
	Уточненное (условное) числовое ограничение	$\geq n$ R.C $\leq n$ R.C $= n$ R.C	$\{a \in \Delta^I \mid \lvert\{b \in \Delta^I \mid (a,b) \in R^I \wedge b \in C^I\}\rvert \geq n\}$ $\{a \in \Delta^I \mid \lvert\{b \in \Delta^I \mid (a,b) \in R^I \wedge b \in C^I\}\rvert \leq n\}$ $\{a \in \Delta^I \mid \lvert\{b \in \Delta^I \mid (a,b) \in R^I \wedge b \in C^I\}\rvert = n\}$	Q
Роли	Имя роли	R	$R^I \subseteq \Delta^I \times \Delta^I$	AL
Аксиомы	Включение понятий	B⊆A	$B^I \subseteq A^I$	AL
	Иерархия ролей	R⊆S	$R^I \subseteq S^I$	I
	Инверсия ролей	R^-	$\{(y,x) \mid (x,y) \in R^I\}$	$^{-1}$

нию, что *Q* должно быть истинным. Понятно, что этот метод может быть менее эффективен, чем более простые модели, например реляционные. Преимуществом использования этого универсального метода вывода является то, что он может использоваться даже в тех случаях, когда БЗ является не полностью определенной.

С помощью автоматического доказательства можно формировать ответы на запросы к *Tbox* (схеме) и *Abox* (экземплярам). Запросы к *Tbox* определяют классификацию и отношения между понятиями, а запросы к *Abox* определяют текущее состояние известных фактов. Выполнимость понятия является доказательством того, что понятие, или понятийное выражение является логически согласованным с БЗ.

Представление и управление онтологиями с использованием дескриптивной логики

Дескриптивная логика может использоваться для управления онтологиями путем отображения понятий и отношений онтологии в утверждения логики. В этом случае, при выполнении доказательств будут использоваться экземпляры понятий, представленных в онтологии. ДЛ используется для создания баз знаний, доказательства логической согласованности баз знаний и ответа на запросы к ним. Доказательство выполнимости БЗ (т. е. то, что она не является логически противоречивой) является обоснованием правильности онтологии.

Сформированные с использованием ДЛ базы знаний также могут использоваться и для обработки запросов. Ответы на запросы получаются в результате кодирования запроса в виде понятий ДЛ и доказательства того, что эти понятия являются выполнимыми. Если запрос является выполнимым, то понятие, описанное в запросе, является логически согласованным с онтологией. Кроме этого операция включения ДЛ может использоваться для обнаружения отношений класс-подкласс, которые заданы в онтологии неявно.

В предлагаемом подходе онтологии предметных областей отображаются на ДЛ посредством набора преобразований, которые позволяют неявные предположения из модели перенести в онтологию. На рисунке показанф, последовательность работы с онтологией с использованием ДЛ.

В соответствии с рассматриваемым подходом, первоначально выполняется неформальное построение модели знаний предметной области (1) путем определения набора основных понятий и отношений между ними. Выявленные в результате этого понятия и отношения используют для описания неформальной онтологии (2). Затем полученная онтология должна быть описана на некотором формальном языке (3), т. е. языке, который имеет строгое соответствие с формальной логикой. После этого описание онтологии считывается программой, которая выполняет грамматический разбор закодированной онтологии и создает структуры данных в виде множества экземпляров классов (4). Полученные структуры данных передаются в систему логического вывода, которая интерпретирует их для формирования утверждений дескриптивной логики (5). Сформированная таким образом модель ДЛ может использоваться для выполнения запросов к ней. В результате всех этих действий доказательства, выполняемые на логической модели, будут использовать факты из онтологии и исходной модели (6).

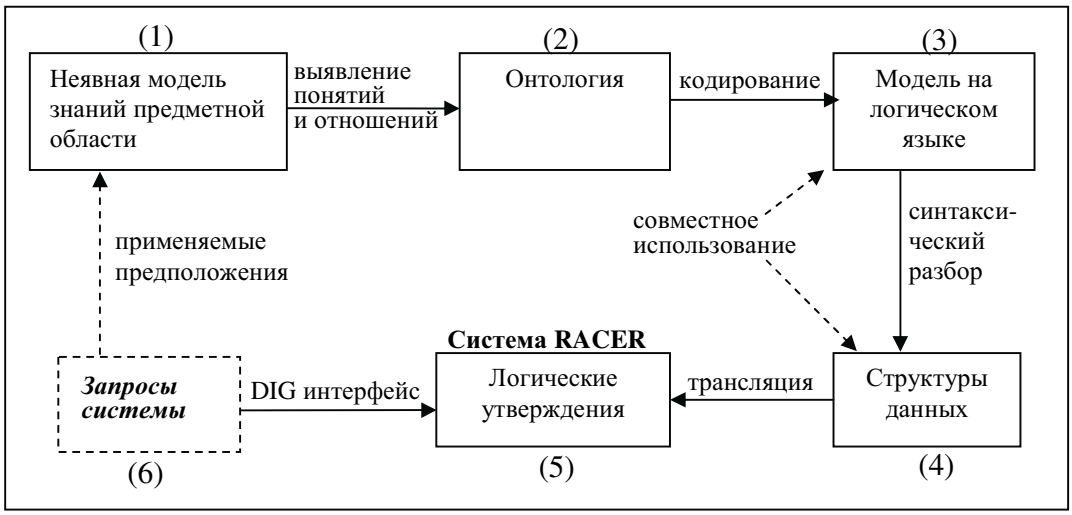

Рисунок. *Последовательность работы с онтологией с использованием ДЛ*

Для записи онтологии на формальном языке (2) могут использоваться стандарты Semantic Web, которые определяют XML-язык для кодирования онтологий, называемый Web Ontology Language (OWL) [3]. Язык OWL-DL является подмножеством OWL языка, хорошо согласующегося с ДЛ. Онтология, написанная на языке Web Ontology Language (OWL), может интерпретироваться в набор утверждений Базы Знаний [2]. Если удастся доказать, что база знаний логически согласована, то это будет означать, что онтология также является согласованной, т. е. правильной. Формальная семантика означает, что запросы к базе знаний отражают факты об онтологии. В отличие от стандарта XML, документ OWL-DL в действительности является логической программой.

Эта технология предоставляет формальный язык для кодирования онтологий, вместе с универсальным XML форматом для ее совместного использования. Вместе они составляют необходимый фундамент для использования онтологий в системах управления знаний, работающих с онтологической моделью.

Для формирования логических структур и обработки запросов могут использоваться системы логического вывода, которые выполняют автоматическое доказательство теорем на моделях ДЛ. В настоящее время доступно несколько таких систем логического вывода. Например, система CORBA FaCT Server реализует БЗ и автоматическое доказательство для ДЛ SHIQ(D) [4] (здесь S обозначает логику ALC_{R+}, а (D) – говорит о том, что используются стандартные типы данных). Система логического вывода RACER реализуют аналогичный набор сервисов для ДЛ $ALCNH_{R+}$, которая очень схожа с SHIQ [5].

Взаимодействие с системой логического вывода может выполняться с использованием стандартного высокоуровневого протокола DIG (*Description logic Implementation Group protocol*) [6]. Многие системы логического вывода поддерживают этот интерфейс, что позволяет выбирать в СУЗ наиболее подходящую.

Проверка правильности онтологий

Когда онтология представлена в виде БЗ, как было описано выше, то можно выполнять доказательства на БЗ, что соответствует выполнению выводов на используемой для ее построения онтологии.

Доказательство выполнимости понятия является доказательством того, что данное понятие логически допустимо относительно фактов БЗ. Понятие является не выполнимым, если определение понятия ведет к противоречию. Когда понятие выполнимо относительно ДЛ, то тогда понятие онтологии является правильным (*valid*), т. е. логически согласованным со всеми определениями онтологии.

Доказательство включения понятия является доказательством того, что одно понятие предполагает другое, в соответствии с БЗ. Когда *subsumes*(*A, B*) в соответствии с БЗ (т. е., *B* предполагает *A*), то тогда *B* является подклассом *A* в онтологии. ДЛ может определять отношения включения (*subsumption*), которые подразумеваются, но не заявлены в онтологии, например, когда доказывается, что два понятия являются логически эквивалентными, то это может предполагать отношения наследования классов (*subsumption*), которые не были явно заявлены утверждениями онтологии.

Вся БЗ является выполнимой, если все понятия в БЗ являются выполнимыми. Это реализуется в виде последовательности тестов *выполнимости понятий*, по одному для каждого понятия БЗ. Когда БЗ является выполнимой, то тогда онтология является *правильной* (*valid*). Если база знаний содержит противоречие, то тогда онтология тоже содержит противоречие и не является правильной.

На основе этого можно использовать следующий метод определения правильности онтологий, описанный в [7]. Данный метод включает следующие шаги:

1. Онтология представляется в виде размеченного графа.
2. Граф последовательно анализируется для формирования набора предложений дескриптивной логики.
3. Каждое предложение добавляется к базе знаний.
4. База знаний проверяется для доказательства ее выполнимости. Если база знаний является выполнимой, то онтология является *правильной*.

В дальнейшем используется следующее упрощенное определение онтологии: **Онтология** это направленный граф, $O=(C, R)$, в котором C – набор вершин, R – набор ребер. Также следует определить понятие правильной онтологии. **Правильная онтология** (*valid ontology*) это онтология, в которой все понятия ($c \in C$) являются выполнимыми (*satisfiable*).

Пусть онтология O_A представляется размеченным графом $O_A=(C_a, R_a)$. Создается база знаний $БЗ_A$ для представления онтологии. Утверждениями базы знаний $БЗ_A$ являются предложения дескриптивной логики, которые определяют понятие для каждого $c_a \in C_a$, и роль или отношение включения для каждого $r_a \in R_a$. По определению *satisfiable*($БЗ_A$): **истинна** если $\forall c_a \in C_a$: *satisfiable*(ca)=*истинна,* иначе **ложна.** В связи с этим определением: *satisfiable*($БЗ_A$)$\Leftrightarrow$*valid*(O_A).

Можно отметить, что это определение требует, чтобы существовало стандартное кодирование онтологии, которое приводит к правильному размеченному графу, и что этот граф будет интерпретироваться для создания правильных утверждений ДЛ. Такое преобразование становится возможным в результате определения формальной семантики онтологического языка, используя язык OWL-DL, который имеет формальную семантику, отражаемую в дескриптивной логике.

Доказательство полагается на содержание базы знаний, которая может рассматриваться, как объединение утверждений о предметной области. Дескриптивная логика является монотонной [2], что означает, что один и тот же набор утверждений будет создавать одну и ту же базу знаний, не зависимо от порядка этих утверждений. При использовании одного и того же алгоритма, интерпретация графа будет всегда создавать одни и те же утверждения.

Если процесс, описанный выше, реализован правильно, то тогда две эквивалентные онтологии должны быть либо обе верны, либо обе неверны, в соответствии с этим тестом. Например, рассмотрим две онтологии: онтологию $O_A=(C_a, R_a)$, и онтологию $O_B=(C_b, R_b)$. Предположим, что $O_A \equiv O_B$, т. е. $(C_a \equiv C_b) \wedge (R_a \equiv R_b)$.

Для онтологии O_A строится база знаний $БЗ_A$. Утверждения $БЗ_A$ являются предложениями дескриптивной логики, которые определяют понятие для каждого $c_a \in C_a$ и роль для каждого $r_a \in R_a$. Аналогично для онтологии O_B строится база знаний $БЗ_B$. Утверждения $БЗ_B$ являются предложениями дескриптивной логики, которая определяет понятие для каждого $c_b \in C_b$ и роль для каждой $r_b \in R_b$.

Так как $(C_a \equiv C_b) \wedge (R_a \equiv R_b)$, то $БЗ_A = БЗ_B$, возможно только за исключением порядка утверждений. Предположим, что O_A является не правильной. По определению это означает, что утверждение *satisfiable*($БЗ_A$) является *ложным*, т. е., $c_{inval} \in C_a$: *satisfiable*(c_{inval})= *false.*

Так как $(C_a \equiv C_b)$, то должно также быть невыполнимое понятие в O_B, $\exists c_{b\text{-}inval} \in C_b$: $c_{b\text{-}inval}=c_{inval}$. Таким образом утверждение *satisfiable*($c_{b\text{-}inval}$) является *ложным*, и поэтому *satisfiable*($БЗ_B$) тоже является *ложным***.** Следовательно, $\neg$*satisfiable*($БЗ_A$)$\Rightarrow\neg$*satisfiable*($БЗ_B$) и *satisfiable*($БЗ_A$)$\Rightarrow$*satisfiable*($БЗ_B$). Аналогичным образом, $\neg$*satisfiable*($БЗ_B$)$\Rightarrow\neg$*satisfiable*($БЗ_A$) и *satisfiable*($БЗ_B$)$\Rightarrow$*satisfiable*($БЗ_A$).

В связи с этим, $(O_A \equiv O_B) \Rightarrow$(*satisfiable*($БЗ_A$)$\Leftrightarrow$(*satisfiable*($БЗ_B$)), и из определения правильной онтологии следует, $(O_A \equiv O_B) \Rightarrow$(*valid*($A$)$\Leftrightarrow$*valid*($B$)).

Заключение

Использование дескриптивной логики позволяет программно реализовать работу с формально описанными онтологиями областей знаний. В частности, появляется возможность выполнять доказательства логической согласованности и других свойств онтологии. Формальные языки описания онтологий являются необходимым средством для работы с ними, но они не предоставляют возможность выполнять требуемые запросы к используемым онтологиям. Для этой цели необходимо использовать систему логического вывода.

СПИСОК ЛИТЕРАТУРЫ

1. Тузовский А.Ф., Чириков С.В., Ямпольский В.З. Системы управления знаниями (методы и технологии). – Томск: Изд-во НТЛ, 2005. – 260 с.
2. The Description Logic handbook: theory, implementation, applications / Ed. F. Baader – Cambridge: Cambridge University Press, 2003. – 564 p.
3. W3C, Web Ontology Language (OWL) Guide Version 1.0. [Электронный ресурс]. – 2002. – Режим доступа: http://www.w3.org/TR/2002/WD-owl-guide-20021104/
4. Patel-Schneider P.F., Horrocks I. DLP and FaCT // Lecture Notes in Artificial Intelligence – Berlin, Springer Verlag, 1999. – P. 19–23.
5. Haarslev V., Moller R. RACER: A core inference engine for the Semantic Web // Proc. of the 2nd Intern. workshop on evaluation of ontology-based tools (EON-2003), Florida, USA, – 2003. – P. 27–36.
6. The DIG Description Logic Interface: DIG 1.1 [Электронный ресурс]. – 2003. – Режим доступа: http://dl-web.man.ac.uk/dig/2003/02/interface.pdf
7. McGrath R.E. Semantic infrastructure for a ubiqutious computing environment [Электронный ресурс]. – 2005. – Режим доступа: http://www.cs.uiuc.edu/research/techreports.php?report=UIUCDCS-R-2005-2587&downioad=pdf