



УДК 004.8

ФОРМАЛЬНЫЕ ОНТОЛОГИИ

Плесневич Г.С.

Московский энергетический институт (Национальный исследовательский университет)
г. Москва, Россия
salve777@mail.ru

Излагаются основные понятия, касающиеся онтологий. Даются уточненные определения формальной онтологии и связанных с ней понятий. Также дается краткое описание системы «Бинарная Модель Знаний», предназначенной для построения онтологий.

Ключевые слова: искусственный интеллект, представление знаний, онтологии, Семантический Веб.

ВВЕДЕНИЕ

С первых работ по искусственному интеллекту проблемы представления знаний рассматривались как центральные в этой области. В 80-х годах прошлого века в отдельных работах, связанных с представлением знаний, стал применяться термин «онтология» как вычислительная модель предметной области или как компонента системы, основанной на знаниях. Но широкое использование этого термина наблюдалось с начала 90-х годов. В настоящее время онтологии стали важнейшим инструментом в различных областях информатики (а не только в искусственном интеллекте).

В настоящей статье мы даем некоторые уточнения понятия онтологии и связанных с ней понятий. Определение онтологии основано на формальной модели понятия в экстенциональном аспекте.

1. Что такое онтология?

Решение задачи в любой области (математика, физика, экономика, инженерное дело, информатика и т.д.) включает рассмотрение объектов и манипуляцию ими. Объектами могут быть как физические (конкретные) сущности, так и ментальные (абстрактные). Таким образом, с задачей ассоциируется множество объектов, называемое *проблемной*, или *предметной областью*. Предметная область, как правило, состоит из объектов, обладающих структурой, и имеются также структуры, связывающие эти объекты друг с другом. Понимание предметной области достигается с помощью понятий, которые классифицируют ее объекты и связи между ними. *Концептуализация* предметной области – это результат фиксации этих понятий.

Концептуализация представляет знание о

предметной области, а не о конкретном состоянии дел в этой области. Иными словами, концептуализация неподвержена изменениям или меняется очень редко. Состояние дел, напротив, часто изменяется. Конкретное состояние дел в предметной области представляется множеством фактов.

Если мы хотим автоматизировать решение задач (что, в частности, служит основной целью искусственного интеллекта и, вообще, информатики), то нужно иметь точные, формальные спецификации концептуализаций.

Каждый *агент* (человек или программная система) имеет свою базу знаний, и только то, что выразимо с помощью онтологии, может быть запомнено и использовано в этой базе знаний. Когда агент желает общаться с другим агентом, он использует конструкторы из некоторой онтологии. Для того чтобы общение было понятным агентам, понимание онтологии должно *разделяться* этими агентами. Таким образом, можно дать следующее определение понятия «онтология» (принадлежащее Т. Груберу [Gruber, 1993]). *Онтология* – это явная формальная спецификация концептуализации, разделяемая некоторым сообществом агентов.

Замечание. Термин «онтология» взят из философии, где он понимается как ветвь метафизики, изучающая наиболее фундаментальные категории бытия, или существования. В информатике «существовать» – значит «быть представленным в компьютере». Поэтому термин «онтология» в информатике имеет значительно более конкретный смысл.

Онтология представляет знание о предметной области, выраженное в общих ее связях и закономерностях. Конкретное знание выражается при помощи фактов, структурированных по этой онтологии. Совокупность этих фактов, называемую *базой фактов*, можно рассматривать как *состояние* онтологии. Онтология, взятая вместе с базой

фактов, составляет *базу знаний* о предметной области. Таким образом, если O – онтология и FB – база фактов (структурированных по онтологии O), то база знаний есть $KB = O \cup FB$.

В базе знаний KB онтология O является стабильной компонентой, а база фактов FB , как правило, переменная. Можно ввести параметр γ , значение которого фиксирует базу фактов и, следовательно, базу знаний: $KB_\gamma = O \cup FB_\gamma$. Этим параметром может быть время, контекст, положение в пространстве, ситуация и т.п. Параметр γ будем называть *точкой соотнесения*.

Онтология состоит из предложений некоторого языка, который должен быть формальным. Факты также являются предложениями (существенно более простого) формального языка. Другими словами, базы знаний (онтологии и базы фактов) строятся как описания, сделанные в (формальных) языках представления знаний.

Концептуализация предметной области осуществляется с помощью понятий. Онтологии поэтому, как формальные описания, должны использовать формальные представления понятий. В разделе 3 мы рассмотрим общую формальную модель понятия.

2. О применении онтологий

С самого начала исследований по искусственному интеллекту центральными были (и остаются) проблемы представления знаний. Понятие онтологии, в сущности, относится к представлению знаний. Но в последние годы работы по онтологиям вышли за пределы теоретических исследований. Разработка онтологий стала занятием инженеров – экспертов по предметным областям.

Онтология дает общий словарь терминов для экспертов и пользователей, которым необходимо совместно использовать информацию в данной предметной области.

Практическое значение онтологий резко возросло в связи с распространением Интернета. В Вебе (WWW) онтологии используются в Google для классификации веб-сайтов. Компания Amazon, осуществляющая продажу через Интернет, разработала онтологию товаров и услуг с их характеристиками. Другой пример – онтология UNSPSC (United Nations Standard Products and Services Code – система ООН стандартных кодов для товаров и услуг).

В настоящее время разрабатываются стандартные онтологии, которые могут использоваться совместно экспертами по предметным областям. Имеются различные факторы, определяющие потребность в построении онтологий. Главные из них следующие:

- необходимость общего понимания экспертами и пользователями (или программными агентами) знания о предметной области для совместного использования этого знания;
- возможность повторного использования знания о предметной области;
- запись в точной явной форме неявных

допущений о предметной области;

- анализ знания о предметной области.

Совместное использование знаний – одна из главных целей построения онтологий. Пусть, например, несколько веб-сайтов содержат информацию в некоторой области. Если эти веб-сайты совместно имеют одну и ту же онтологию терминов, то становится возможным применение программных агентов, которые могут извлекать информацию из этих сайтов, создавая соответствующую базу знаний. Программные агенты могут использовать эту базу знаний для ответов на запросы пользователей или использовать эти знания как входную информацию для других приложений.

Возможность повторного использования знания – важная черта процесса построения онтологий. Если одни эксперты построили онтологию некоторых понятий, то другие могут просто повторно использовать ее при построении онтологий для своих предметных областей. Если эксперту нужно построить большую онтологию, то он может интегрировать несколько известных онтологий, описывающих части большой предметной области.

Формулирование явных допущений о моделируемой предметной области позволяет в дальнейшем менять эти допущения, когда изменяется наше знание о предметной области. Это формулирование выполняется в декларативной форме. Заметим, если использовать процедурное представление допущений (в обычных языках программирования), то эти допущения не только было бы сложно понять, но также сложно изменять.

Декларативное представление знаний в онтологиях позволяет использовать логические методы анализа онтологий. Важный вопрос анализа – является ли данная онтология непротиворечивой или противоречивой. Этот вопрос возникает, например, при интеграции онтологий. Другой вопрос – являются ли данные предложения онтологии избыточными.

В настоящее время онтологии рассматриваются как весьма полезный формальный инструмент для представления семантики для многих приложений. Например, онтологии применяются в следующих областях:

- Программная инженерия;
- Семантический Веб;
- веб-сервисы;
- бизнес-информатика;
- биоинформатика;
- многоагентная технология;
- электронное обучение;
- машинная обработка естественного языка.

3. Формальные понятия и формальные онтологии

Американский толковый словарь Webster следующим образом определяет термин «понятие»: An abstract or general idea inferred or derived from specific instances (Абстрактная или общая идея,

выведенная или произведенная из конкретных примеров.)

Понятие есть ментальная сущность (объект), а ее примерами (экземплярами) могут быть как физические, так и ментальные сущности. Для того чтобы моделировать понятия в компьютере, необходимо символическое, формальное представление понятий и их примеров. Такое представление основано на именах, которые присваиваются (реально или потенциально) моделируемым объектам. Между именами и моделируемыми объектами имеется отношение референции *ref*, которое назначает объекту его имя. Вообще говоря, данный объект может иметь не одно имя, а несколько. Мы скажем, что имена *a* и *b* *корелферентны* и запишем $a \sim b$, если они обозначают один и тот же объект. Таким образом, $a \sim b$ тогда и только тогда, когда существует объект *o* такой, что $o \text{ ref } a$ и $o \text{ ref } b$. Ясно, что корелференция является отношением эквивалентности.

Понятие имеет два аспекта – экстенционал и интенционал. *Экстенционал* (или объем) понятия – это множество всех его примеров. *Интенционал* – это совокупность условий, характеризующих примеры понятия.

Формальное понятие строится из имен. В экстенциональном аспекте *формальное понятие* имеет следующие компоненты:

- имя понятия *C*;
- *универсум* понятия U^C – множество всех имен, которые могут обозначать примеры понятия *C*;
- *множество точек соотнесения* Γ ;
- подмножество $E_\gamma^C \subseteq U^C$ для каждой точки соотнесения $\gamma \in \Gamma$ – *множество всех примеров* понятия *C* в этой точке (точнее, имен примеров);
- отношение эквивалентности \sim_γ , заданное на множестве E_γ^C для каждой точки соотнесения $\gamma \in \Gamma$ – *корелференция* в этой точке;
- пара $\text{Ext}_\gamma^C = (E_\gamma^C, \sim_\gamma)$ для каждой точки соотнесения $\gamma \in \Gamma$ – *экстенционал* понятия *C* в этой точке;
- семейство $\text{Ext}^C = \{ \text{Ext}_\gamma^C \mid \gamma \in \Gamma \}$ – *полный экстенционал* понятия *C*.

Объекты предметной области обладают, как правило, свойствами, или *атрибутами*. Атрибут *A* формального понятия *C* можно рассматривать как функцию, заданную на множестве $E^C = \bigcup \{ E_\gamma^C \mid \gamma \in \Gamma \}$. Значением этой функции служит имя примера другого понятия (или этого же понятия) или элемент некоторого типа данных.

Пример 1. Рассмотрим задачу построения системы, хранящей сведения об учебном процессе. Тогда можно ввести такие понятия, как «студент», «преподаватель», «предмет» и т.п. Предположим, что понятие «студент» имеет атрибуты ФИО, ГодРожд (год рождения), Руков (руководитель). Формальное понятие «студент» определим как имеющее следующие компоненты.

- Имя понятия: Студент;
- Универсум понятия:
 $U^{\text{Студент}} = \text{iName} \cup \text{Surr}$

{[ФИО:х, ГодРожд:у, Руков:z] |
 $x \in \text{String}, y \in \text{Integer}, z \in \text{Surr}$ }

Здесь *iName* – тип данных, элементы которого используются для имен индивидуальных объектов, а *Surr* – тип данных, элементами которого служат так называемые *суррогаты*, т.е. стандартные системные имена, идентифицирующие объекты – примеры понятий. Элементами типа данных *Surr* будем считать #1, #2, #3, ...

- Множество точек соотнесения:

$\Gamma = \{ \text{сем1}, \text{сем2}, \dots, \text{сем10} \}$.

Эти точки представляют семестры. Они выбраны потому, что информация об учебном процессе может меняться каждый семестр.

- Экстенционалы $\text{Ext}_{\text{сем}j}^{\text{Студент}}$ ($j = 1, 2, \dots, 10$) могут быть представлены таблицами. Например, для $\text{Ext}_{\text{сем2}}^{\text{Студент}}$ можно указать следующую таблицу.

Студент

Surr	Coref	ФИО	ГодРожд	Руков
#60	['староста группы A13']	'Андреев В.М.'	1993	#11
#61	[]	'Борисов И.В'	1992	#11
:				
#97	[]	'Яковлев Л.Я'	1992	#12

Здесь *Surr* – атрибут, значениями которого служат суррогаты, а *Coref* – атрибут, значениями которого служат списки корелферентных имен. Первая строка таблицы представляет три корелферентных имени:

#61 ~ 'староста группы A13' ~ [ФИО:'Андреев В.М.', ГодРожд:1993, Руков:#11].

Таким образом, имеем

$E_{\text{сем2}}^{\text{Студент}} = \{ \#60, \#61, \#97, \text{'староста группы A13'}, [\text{ФИО:'Борисов И.В.', ГодРожд:1993, Руков:#11}], \dots, [\text{ФИО:'Яковлев Л.Я.', ГодРожд:1993, Руков:#11}] \}$.

- Отношение корелференции $\sim_{\text{Студент}_{\text{сем}j}}$ определяется из строк таблицы для точки соотнесения *семj* так, как это было показано выше.

- Экстенционал понятия Студент в точке соотнесения *семj* есть пара $\text{Ext}_{\text{сем}j}^{\text{Студент}} = (E_{\text{сем}j}^{\text{Студент}}, \sim_{\text{Студент}_{\text{сем}j}})$.

- Полный экстенционал понятия Студент есть семейство

$\text{Ext}^{\text{Студент}} = \{ \text{Ext}_{\text{сем1}}^{\text{Студент}}, \text{Ext}_{\text{сем2}}^{\text{Студент}}, \dots, \text{Ext}_{\text{сем10}}^{\text{Студент}} \}$.

Мы предполагаем, что руководителем студента является некоторый преподаватель. В формальном понятии Студент этому соответствует атрибут Руков, значениями которого служат суррогаты, обозначающие примеры понятия Препод. Экстенционал этого понятия также может быть представлен таблицей, например, таблицей с атрибутами ФИО, Должность и Кафедра (кроме

стандартных атрибутов Surf и Coref). Предположим, что эта таблица содержит следующую строку.

#11	[]	‘Вагин В.Н.’	Профессор	ПМ
-----	----	--------------	-----------	----

Применение атрибута A к имени e , обозначающего пример понятия, записывается как $e.A$. В данном случае, например, имеем
 #60.ФИО = ‘Андреев В.М.’,
 #60.ГодРожд = 1993,
 ‘староста группы А13’.ФИО = ‘Андреев В.М.’,
 #60.Руков = #11, #60.Руков.ФИО = #11.ФИО = ‘Вагин В.Н.’.

Как мы видели, экстенционал понятия C в каждой точке соотнесения $\gamma \in \Gamma$ есть пара $(E_\gamma^C, \sim_\gamma)$, состоящая из множества имен E_γ^C , обозначающих примеры понятия, и отношения кореференции \sim_γ , заданного на этом множестве имен E_γ^C . Если взять фактор-множество E_γ^C / \sim_γ множества E_γ^C по отношению эквивалентности \sim_γ , то это множество будет состоять из абстрактных объектов, которые взаимно однозначно соответствуют объектам предметной области, рассматриваемых в точке соотнесения γ .

Заметим, что в примере 1 экстенционалы в каждой точке соотнесения конечны. Но в общем случае экстенционалы формальных понятий могут быть счетными.

Данное выше определение формального понятия относится к экстенциональному аспекту термина «понятие». Но имеется также интенциональный аспект этого термина, связанный со спецификацией компонентов формального понятия. Мы используем термин «спецификация», а не термин «определение», имея в виду, что результат определения однозначен, но результат спецификации может быть неоднозначен.

В примере 1 два понятия Студент и Преподаватель связаны друг с другом с помощью атрибута Руководитель. Кроме того, эти понятия имеют одни и те же точки соотнесения.

Вообще, пусть E – конечное множество формальных понятий, имеющих одни и те же точки соотнесения. Тогда мы скажем, что E – *система понятий*. Некоторые из понятий системы могут быть связаны друг с другом. Но каждую связь можно также определить как формальное понятие. Поэтому можно считать, что система понятий E включает понятия, определяющие связи между понятиями.

Формальной онтологией для системы понятий E назовем конечное множество предложений O некоторого формального языка, специфицирующих формальные понятия из E . (В дальнейшем, говоря о формальных понятиях, будем опускать слово «формальные».)

Предположим, что все понятия $C \in E$ специфицированы с помощью предложений некоторого формального языка. Предложения этого языка в соответствие с тем, какому из компонент понятия они специфицируют, классифицируются

следующим образом.

- *Структуральные предложения.* Специфицируют универсумы понятий.
- *Логические предложения.* Специфицируют экстенционалы понятий равномерно по точкам соотнесения.

- *Транзитные предложения.* Специфицируют изменения экстенционалов при переходе от одной точки соотнесения к другой. Эти переходы определяются некоторым бинарным отношением, заданным на точках соотнесения.

Замечание. Фактически спецификация универсума является определением, т.е. результат спецификации однозначен (как это видно в примере 1).

Пример 2. Для спецификации указанного в примере 1 универсума $U^{\text{Студент}}$ понятия Студент можно взять структуральное предложение $\text{Студент}[\text{ФИО:String, ГодРожд:Integer, Руков:Преподаватель}]$.

Логические предложения мы понимаем очень широко. Это предложения, интерпретируемые как произвольные ограничения на экстенционалы для любой точки соотнесения. Возьмем, например, ограничение на экстенционалы понятия Студент, состоящее в том, что студенты одной и той же группы должны иметь только таких руководителей, которые работают на одной и той же кафедре. Это ограничение можно представить следующим логическим предложением:

FORALL Студент X, Y (IF X.Группа = Y.Группа THEN X.Руководитель.Кафедра = Y.Руководитель.Кафедра).

Транзитные предложения представляют ограничения, выражающие изменения экстенционалов при переходах в соответствие с бинарным отношением, заданным на точках соотнесения. В данном случае естественно взять отношение Позже, определяемое следующим образом: $\text{сем}_j \text{ Позже } \text{сем}_k \Leftrightarrow_{\text{df}} j > k$. Возьмем ограничение, состоящее в том, что для каждого студента допускается только одна смена руководителя. Тогда это ограничение можно представить следующим предложением:

FORALL Студент X; Рог Y
 (IF EXIST Рог Z; Z Позже Y; (X.Руководитель < Y > =/= X.Руководитель < Z >)
 THEN FORALL Рог V; V Позже Z; (X.Руководитель < V > = X.Руководитель < Z >).

Здесь Рог обозначает тип данных «точка соотнесения» (point of reference). Выражение X.Руководитель < Y > обозначает значение X.Руководитель в точке соотнесения Y.

Замечание. В примере 1 были приведены предложения из языков системы «Бинарная Модель Знаний» (см. раздел 4) [Plesniewicz, 2004], [Плесневич, 2005].

Пусть E – система понятий, отвечающая данной концептуализации, а O – формальная онтология, специфицирующая E . Тогда E служит моделью онтологии O . В соответствующей интерпретации все предложения из O будут истинными. Эту интерпретацию назовем *главной*. Но, поскольку результат спецификации, как правило, неоднозначен, существуют другие интерпретации онтологии O (соответствующие другим системам

понятий).

На основе этих интерпретаций можно определить отношение \models логического следствия между онтологией \mathcal{O} и логическим или транзитным предложением σ языка логической спецификации:

$\mathcal{O} \models \sigma \Leftrightarrow_{df}$ не существует интерпретации, в которой все логические и транзитные предложения из \mathcal{O} истинны, а предложение σ ложно.

Понятие логического следствия используется при анализе онтологии. Это анализ включает, в частности, решение следующих задач:

- выяснить, является ли онтология \mathcal{O} противоречивой;
- для данного логического или транзитного предложения σ выяснить, верно ли $\mathcal{O} \models \sigma$;
- для данного логического или транзитного предложения σ выяснить, является ли это предложение избыточным, т.е. верно ли, что $\mathcal{O} \setminus \{\sigma\} \models \sigma$.

База фактов FB для данной онтологии \mathcal{O} состоит из конечного множества примеров и контрпримеров понятий, входящих в эту онтологию. Точнее, пусть Δ – конечное подмножество множества Γ . Для каждой точки соотнесения $\gamma \in \Delta$ и каждого понятия $C \in \mathcal{O}$ возьмем конечное подмножество $+F_\gamma^C$ множества E_γ^C и конечное подмножество $-F_\gamma^C$ множества $U^C \setminus E_\gamma^C$. Для любых элемента $e \in U$, понятия C и точки соотнесения γ выражения $+(e, C, \gamma)$ и $-(e, C, \gamma)$ назовем фактами. Первое выражение интерпретируется как утверждение « e – пример понятия C в точке соотнесения γ », а второе – как утверждение « e – контрпример понятия C в точке соотнесения γ ». База фактов – это множество

$$FB = \{+(e, C, \gamma) \mid e \in +F_\gamma^C, \gamma \in \Gamma\} \cup \{-(e, C, \gamma) \mid e \in -F_\gamma^C, \gamma \in \Gamma\}.$$

База знаний – это множество $KB = \mathcal{O} \cup FB$.

Определение ответов на запросы к базе знаний можно определить на основе отношения логического следствия. Запрос можно рассматривать как предложение σ , составленное из принадлежащих онтологии \mathcal{O} имен и содержащее свободные индивидуальные переменные x_1, x_2, \dots, x_n . Ответ на запрос σ – это множество

$$\{(x_1, x_2, \dots, x_n) \mid KB \models \sigma\}.$$

В Семантическом Вебе онтологии играют центральную роль (см., например, [Antonioni & Harmelen, 2008]). Различные языки для онтологий в Семантическом Вебе были разработаны в последнее десятилетие. Организация W3C (World Wide Web Consortium) предложила стандарты для основных языков. Стандартным синтаксисом для этих языков является синтаксис XML. Предложения языков XML, XML Schema и RDF являются структуральными. Язык RDF Schema содержит простейшие логические предложения. Языки семейства OWL содержат логические предложения, эквивалентные языкам дескриптивной логики (description logic) [Baader et al., 2003].

Разработанный М. Кифером язык F-logic (фреймовая логика) также используется для построения онтологий. Он содержит логические

предложения, равные по выразительной силе логике первого порядка [Kifer et al., 1995]. М. Кифер также предложил так называемую логику транзакций (transaction logic), содержащую предложения, которые можно классифицировать как транзитные [Bonner & Kifer, 1994].

Замечание. В логических языках, являющимися вариантами логики первого порядка формальные понятия представляются одноместными предикатами, а их атрибуты – одноместными функциями. В языках модальной пропозициональной логики формальные понятия представляются пропозициональными переменными, причем точками соотнесения служат миры.

4. Бинарная Модель Знаний

Бинарная Модель Знаний (БМЗ) – это система концептуальных (понятийно-ориентированных) языков, предназначенная для построения онтологий.

В БМЗ имеются два типа понятий: *классы* и *бинарные связи*. (Бинарная связь представляет бинарное отношение между элементами классов).

В БМЗ используются как простые, так и составные имена для объектов, классов и бинарных связей. Простое имя класса или имя бинарной связи представляется слитной строчкой символов, начинающееся с большой буквы, и без знаков препинания и специальных символов, а также произвольной строчкой символов, взятых в одиночные кавычки. Составные имена классов и бинарных связей получаются применением соответствующих операций.

• $C(*)$ обозначает класс, примерами которого служат конечные множества примеров понятия C , т.е.

$$U^{C(*)} = \{x \mid x - \text{конечное подмножество } U^C\}, \\ E^{C(*)} = \{x \mid x - \text{конечное подмножество } E^C\}.$$

• $C(p, q)$, где p и q – натуральные числа и $p \leq q$, обозначает понятие, примерами которого служат конечные множества примеров понятия C , причем размеры этих множеств заключены в пределах чисел p и q .

• $L((p, q)(r, s))$, где $p \leq q$ и $r \leq s$, обозначает бинарную связь, примерами которой служат конечные множества примеров связи L . При этом, если множество X из $L((p, q)(r, s))$ рассматривать как двудольный граф, то степень любой левой (правой) его вершины должна быть заключена между числами p и q (r и s).

• $C_1 \mid C_2 \mid \dots \mid C_n$ обозначает понятие, примерами которого служат примеры всех понятий C_j .

В БМЗ входят следующие языки:

• язык *структурной спецификации* (ЯСС). Предложения ЯСС определяют универсумы понятий;

• язык *спецификации типов данных* (ЯТД). Предложения ЯТД задают абстрактные типы данных, которые используются как домены атрибутов;

• язык *атрибутивных условий* (ЯАУ). Предложения ЯАУ определяют выраженные в

терминах атрибутов ограничения на примеры понятий;

- язык запросов (ЯЗ). Предложениями ЯЗ служат запросы к базам фактов;
- язык логической спецификации (ЯЛС). ЯЛС содержит несколько подязыков.

Онтология в БМЗ – это конечное множество O предложений из ЯСС U ЯТД U ЯЛС. Обозначим через $O/ЯСС$, $O/ЯТД$ и $O/ЯЛС$ части онтологии O , состоящие соответственно из предложений языков ЯСС, ЯТД и ЯЛС. В синтаксически корректной онтологии O предложения из этих частей должны быть согласованы в следующем смысле:

- структура понятий, участвующих в предложении из $O/ЯЛС$, определяется предложениями из $O/ЯСС$;
- атрибуты, содержащиеся в предложении из $O/ЯСС$, имеют домены, которые являются предложениями из $O/ЯТД$.

Предложения ЯСС состояются из примитивных предложений, имеющих следующие формы:

$C[D]$, $C[A:D]$, $C[A:T]$, (CLD) , $(CLD)[E]$,
 $(CLD)[A:E]$, $(CLD)[A:T]$,

где C , D , E – имена понятий, L – имя бинарной связи, а T – спецификация типа данных.

Имя D в предложении $C[D]$ является компонентой класса C . Также E в предложении $(CLD)[E]$ является компонентой бинарной связи L . Имя C в первых трех предложениях или выражение (CLD) в остальных четырех предложениях называются *головой* этих предложений. Предложения с одинаковыми головами могут быть соединены в одно предложение. Например, из примитивных предложений $C[D]$, $C[A:E]$ и $C[A:T]$ получается предложение $C[D,A:E,A:T]$.

Приведем пример онтологии в языке ЯСС.

Пример 3.

<‘Учебный процесс’>

1. Студент[ФИО:String, Группа:String].
2. Препо[ФИО:String, Должность:String
Работает_на:Кафедра, Стаж:Integer].
3. Сотрудник[Препо | НаучСотр | Иженер |
Лаборант | Секретарь].
4. Предмет[Назв:String, Зависит_от:Предмет,
КоличЧасов:Integer].
5. Кафедра[Назв:String ЗавКафедрой:Препо,
Состав:Сотрудник(*), Факультет:String].
6. (Студент СдалЭкз Предмет)[Дата:Date,
Оценка: {3,4,5}, Кому:Препо].

</ ‘Учебный процесс’>

ЗАКЛЮЧЕНИЕ

В настоящей статье даны некоторые уточнения понятия онтологии и связанных с ней понятий. Определение онтологии основано на формальной модели понятия в экстенциональном аспекте. Дано краткое описание системы «Бинарная Модель Знаний», предназначенной для построения онтологий.

Работа выполнена при финансовой поддержке Российского Фонда Фундаментальных Исследований по проекту №11-01-00538.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Antoniou & Hamerlen, 2003] Antoniou G., van Hamerlen F. A Semantic Web Primer. – MIT Press, 2003.
- [Baader et al. 2003] Baader F., Calvanese D., McGuinness D., Nardi D., Patel-Schneider P. (eds.) Description Logic Handbook. – Cambridge University Press, 2003.
- [Bonner & Kifer, 1994] Bonner A.J., Kifer M. An overview of transaction logic // Theoretical Computer Science, 193 (2), 1994.
- [Gruber, 1993] Gruber T.R. A translation approach to portable ontology specifications // Knowledge Acquisition, 5 (2), 1993.
- [Kifer et al., 1995] Kifer M., Lausen G., Wu J. Logical foundation of object-oriented and frame-based languages // Journal of ACM, 42 (4), 1995.
- [Plesniewicz, 2004] Plesniewicz G.S. Binary Data and Knowledge Model // In: Stefanuk, Kaijiri (eds.) Knowledge-based software engineering. – IOS, 2004.
- [Плесневич, 2005] Плесневич Г.С. Бинарные модели знаний // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов III-го Международного научно-практического семинара (Коломна, 15-17 мая 2005 г.). – М.: Физматлит, 2005.

FORMAL ONTOLOGIES

Plesniewicz G.S.

Moscow Power Engineering University, Russia

salve777@mail.ru

Main concepts related to ontologies are presented. Some refined definition of formal ontology is given.

INTRODUCTION

In the paper we give a definition of formal ontologies. The definition is based on the formal model of concept which is extensional in essence. Using this definition we define the notion of formal ontology.

MAIN PART

A formal concept has the following components: (a) the name C ; (b) the universe U^C ; (c) the set of points of reference Γ ; (d) for every $\gamma \in \Gamma$ a subset $E_\gamma^C \subseteq U^C$ – the set of all instances at the point of reference γ ; (e) for every $\gamma \in \Gamma$ an equivalence \sim_γ on E_γ^C – the coreference relation at the point of reference γ ; (f) for every $\gamma \in \Gamma$ the pair $Ext_\gamma^C = (E_\gamma^C, \sim_\gamma)$ – the extension of C at the point of reference γ ; (g) the family $Ext^C = \{Ext_\gamma^C | \gamma \in \Gamma\}$ – the full extension of the concept C .

A concept system is a finite set of formal concepts with the same points of reference. A formal ontology is a specification (by some formal languages) of a given concept system.

Using these definitions we also define main notions related to formal ontologies (interpretations, logical consequence, consistence et al.)

CONCLUSION

In the paper, the refined notion of formal ontologies and related notions are defined.