

УДК 004.023

В.В. Любченко, канд. техн. наук, доц.,
В.С. Кавицька, магістр,
Одес. нац. политехн. ун-т

МЕТОДИКА ОПРЕДЕЛЕНИЯ ЭКВИВАЛЕНТНОСТИ КЛАССОВ ОНТОЛОГИЙ

В.В. Любченко, В.С. Кавицька. Методика визначення еквівалентності класів онтологій. Розглянуто проблему інтеграції онтологій моделей предметних областей. Проведено аналіз етапів інтеграції онтологій моделей предметних областей. Розроблено методіку визначення ступеня еквівалентності між класами двох онтологій. Розглянуто операції над класами на етапі інтеграції онтологій.

Ключові слова: онтології моделей предметних областей, інтеграція онтологій, еквівалентність класів, операції над класами.

В.В. Любченко, В.С. Кавицька. Методика определения эквивалентности классов онтологий. Рассмотрена проблема интеграции онтологий моделей предметных областей. Проведен анализ этапов интеграции онтологий моделей предметных областей. Разработана методика определения степени эквивалентности между классами двух онтологий. Рассмотрены операции над классами на этапе интеграции онтологий.

Ключевые слова: онтологии моделей предметных областей, интеграция онтологий, эквивалентность классов, операции над классами.

V.V. Lyubchenko, V.S. Kavitskaya. Method of determining the equivalence classes of ontologies. The problem of ontology integration domain models is considered. The stages of an integration of ontology domain models are analyzed. The method of determining the degree of equivalence between the classes of two ontologies are developed. The operations on classes at the stage of integrating ontologies are considered.

Keywords: the ontology domain models, the integration of ontologies, the equivalence classes, the operations on classes.

Введение. Онтологические технологии широко используются в информационных системах для хранения, представления, формализации, систематизации и обработки информации. По своей сути онтология предметной области представляет собой формальную модель понятийной структуры предметной области. В формулировке Грубера онтология определена как формальная спецификация концептуализации, которая имеет место в некотором контексте предметной области [1]. Под концептуализацией понимается представление предметной области через описание множества понятий (концептов) предметной области и связей (отношений) между ними. Путем создания онтологий формируется согласованное между специалистами формализованное представление структуры предметной области.

Из известных операций над онтологиями интеграция онтологий является одной из наиболее важных и часто используемых [2]. Суть указанной операции состоит в построении результирующей онтологии из двух исходных с сохранением первичных онтологических знаний в таком виде, чтобы результирующая онтология была максимально связанной (содержала все отношения между классами) и не содержала дублирующих данных (например, эквивалентных классов).

Постановка проблемы. Задача интеграции онтологий моделей предметных областей обсуждается довольно широко — уже сформулировано множество условий, определений и методов, представлены различные уровни и этапы интеграции онтологий. Но даже сейчас нет четкого соглашения о том, что включает в себя интеграция онтологий. Анализ ситуации в области интеграции онтологий позволяет сделать вывод о недостаточной изученности интеграции онтологий в целом [2, 3]. Также мало уделено внимания одному из основных этапов, который существенно влияет на результат интеграции онтологий в целом, а именно, определению степени эквивалентности между классами двух онтологий.

Цель работы — разработка методики определения степени эквивалентности классов двух онтологий как одного из основных этапов интеграции онтологий моделей предметных областей.

Основной материал. Формально онтология представляется тройкой $\langle X, R, F \rangle$, где X — конечное множество концептов (понятий, терминов) предметной области, которую представляет онтология O ; R — конечное множество отношений между концептами (понятиями, терминами) заданной предметной области; F — конечное множество функций интерпретации, заданных на концептах и/или отношениях онтологии O [4]. Данная модель носит общий характер и на практике используется ее реализация в таких системах, как например, Protege [5]. Система Protege — одна из наиболее популярных систем работы с онтологиями, созданная в Стэнфордском университете (США). По версии разработчиков системы Protege все понятия предметной области делятся на классы, подклассы, экземпляры. Экземпляры могут быть как у класса, так и подкласса и описываются фреймом. Далее рассматривается модель системы Protege в качестве основной.

В общем случае методику интеграции онтологий моделей предметных областей можно разделить на три относительно независимых этапа, на каждом из которых учитывается результат, полученный на предыдущем этапе [6]:

1. Определение степени эквивалентности между классами двух онтологий. На данном этапе производится сравнение всех классов одной онтологии со всеми классами другой. Качество процедуры сравнения существенно влияет на качество интеграции онтологий.

2. Поэлементное объединение множеств классов и множеств слотов исходных онтологий. Основная часть, которая и является интеграцией, основывается на степени эквивалентности, вычисленной на предыдущем этапе для каждой пары классов. Этот этап, как правило, выполняется полностью автоматически, так как не требует принятия решений относительно семантики классов и полностью формализован. Выбор алгоритма работы данного этапа интеграции онтологий влияет на степень интеллектуальности всей процедуры. Под интеллектуальностью процедуры интеграции онтологий понимается возможная степень сложности результирующих графов — родовидовых иерархий (классы, соединенные отношениями “класс-подкласс”) и семантических сетей (классы, соединенные семантическими отношениями), которые могут быть построены при интеграции, возможная степень их отличия от соответствующих графов исходных онтологий.

3. Проверка результата интеграции на корректность. Под корректностью понимается непротиворечивость результирующей онтологии и степень ее соответствия ожиданиям пользователя. Степень же соответствия может быть полностью реализована только самим пользователем, тогда как критерий противоречивости может быть определен формально. Возможные типы противоречий определяются моделью описания онтологии. В простом варианте — это противоречия, связанные с уникальностью имен классов и слотов. В общем случае, противоречия определяются ложностью аксиом и функций интерпретации. Поскольку большинство редакторов онтологий имеют ограниченные возможности задания логических зависимостей между классами и их отношениями, противоречия таких типов не проверяются в связи с отсутствием в модели функций интерпретации в явном виде (модель Protege, например, сводит задание функций интерпретации к заданию фасетов).

Таким образом, первый этап интеграции онтологий моделей предметных областей является наиболее важным и подлежит подробному рассмотрению.

Разработаем методику определения эквивалентности классов двух онтологий. Пусть заданы исходные онтологии O_1 и O_2 и пара сравниваемых классов $C_1 \in O_1$ и $C_2 \in O_2$. Необходимо определить эквивалентен ли класс C_1 классу C_2 , при этом результатом указанной операции является логическое значение либо “истина” либо “ложь”.

Эквивалентность классов может быть определена наиболее простым способом путем сравнения классов по имени. Данный подход является основным критерием большинства существующих модулей интеграции онтологий. В наиболее развитых из них (таких как PROMPT [7]) существует возможность нечеткого сравнения имен классов и сравнения с учетом синонимов, однако в случае сложных синонимических и омонимических конструкций сравнение классов

по данному критерию дает неверные результаты. Причиной таких ошибок является предположение о роли класса в онтологии по его имени, тогда как имя не определяет класс (как термин не определяет понятия), а служит удобной для человека меткой класса.

Более сложным вариантом сравнения классов является сравнение по содержанию, однако во многих случаях отождествление классов (понятий) на основе поэлементного равенства всех их экземпляров является ошибочным. Несмотря на это, сравнение объемов классов активно используется в современных системах как вспомогательная информация для установления подобия между ними.

Понятие полностью определяется своим содержанием, что следует из самого определения содержания понятия [8], а поскольку класс в модели Protege аналогичен понятию, то это же должно быть справедливо и для классов.

Исходя из этого, можно заключить, что при сравнении классов наиболее важной составляющей в структуре является их содержание, т.е. множество слотов, доменом которых является данный класс. Однако, отношения между множествами (в данном случае, множествами слотов) не ограничиваются отношениями эквивалентности и различия — последнее может представлять включение одного множества в другое, включение второго множества в первое, непустое пересечение двух множеств и отсутствие общих элементов. Указанные отношения множеств слотов данных $\{S_1\}$ и $\{S_2\}$ двух классов C_1 и C_2 соответственно определим следующим образом:

— Совпадение всех слотов класса C_1 со всеми слотами класса C_2 (поэлементное равенство множеств $\{S_1\}$ и $\{S_2\}$) означает равенство содержания двух классов, что свидетельствует об эквивалентности классов.

— Включение множества $\{S_1\}$ слотов класса C_1 во множество $\{S_2\}$ слотов класса C_2 ($\{S_1\} \subset \{S_2\}$) свидетельствует об обобщении содержания класса C_2 до содержания класса C_1 . Это означает, что сам класс C_1 является более общим по отношению к классу C_2 . Исходя из принципа обратной зависимости содержания и объема понятий (увеличение содержания влечет уменьшение объема и наоборот), можно заключить, что класс C_1 является одним из надклассов класса C_2 .

— Включение множества $\{S_2\}$ слотов класса C_2 во множество $\{S_1\}$ слотов класса C_1 ($\{S_1\} \supset \{S_2\}$) является случаем, обратным случаю (2), и означает, что класс C_1 является подклассом C_2 , т.е. понятие, представляемое классом C_1 , является уточнением понятия, представляемого классом C_2 .

— Пересечение множеств слотов классов C_1 и C_2 ($\{S_1\} \cap \{S_2\} \neq \emptyset$) свидетельствует о наличии общих слотов. Это означает, что существует некоторый класс C , являющийся надклассом для классов C_1 и C_2 , а сами классы принадлежат одному уровню иерархии, что соответствует частичной эквивалентности.

— Пустое пересечение множеств слотов классов C_1 и C_2 ($\{S_1\} \cap \{S_2\} = \emptyset$) свидетельствует о том, что у первого общего для C_1 и C_2 некоторого надкласса C отсутствуют слоты. В этом случае эти классы можно считать полностью различными.

Отметим только, что такой механизм не применим к классам с пустым множеством слотов — в этом случае (согласно описанному механизму) такие классы должны быть эквивалентны, что, однако, в большинстве случаев неверно. Пустое множество слотов класса говорит о том, что понятие, представляемое классом, не требует описания его содержания в явном виде, либо что его содержание неизвестно. Последний случай следует отнести к некорректности входных данных, так как операция ориентирована на интеграцию полных онтологий. Случай же с необязательностью указания содержания некоторого понятия возникает при уверенности при создании онтологии, что оно (понятие) известно интерпретатору онтологии (системе обработки онтологий или другому специалисту по знаниям). В таком случае понятие, а следовательно и представляющий его класс, должны однозначно идентифицироваться своим именем, поэтому предлагается классы без слотов сравнивать по именам.

Далее, основываясь на результатах поэлементного сравнения классов, можно перейти к непосредственной интеграции онтологий, которая, как отмечалось ранее, выполняется в основном автоматически, так как не требует принятия решений относительно семантики классов. Отметим только, что перечисленные варианты эквивалентности классов приводят к такому же количеству различных операций над классами на этапе интеграции онтологий:

— если классы эквивалентны, то они представляют одно и то же понятие в онтологии, следовательно, должны быть “склеены” в один;

— если класс одной онтологии является обобщением соответствующего класса другой онтологии, такие классы должны представляться как класс и подкласс, соответственно, причем совпадающие слоты должны быть удалены из подкласса, так как они будут унаследованы от надкласса;

— если класс одной онтологии является уточнением соответствующего класса другой онтологии, такие классы должны представляться как подкласс и класс, соответственно, причем совпадающие слоты должны быть удалены из подкласса, так как они будут унаследованы от надкласса;

— если классы двух онтологий частично эквивалентны, то они представляют собой схожие понятия, то есть должны иметь общий надкласс, являющийся их обобщением, при этом совпадающие слоты должны быть удалены из подкласса, так как они будут унаследованы от обобщающего класса;

— если классы различны, они должны быть скопированы в результирующую онтологию.

Выводы. Описанная методика определения эквивалентности классов является одним из основных этапов интеграции онтологии моделей предметных областей. Качество выполнения этого этапа существенно влияет на качество интеграции онтологий моделей предметных областей в целом.

Литература

1. Gruber, T.R. A translation approach to portable ontology specifications / T.R. Gruber // Knowledge systems laboratory, Computer Science Department. — Stanford, California, 1993. — Vol. 6, № 3. — P. 71 — 92.
2. Овдей, О.М. Обзор инструментов инженерии онтологий [Электронный ресурс] / О.М. Овдей, Г.Ю. Проскудина // Тр. VI Всерос. науч. конф. “Электрон. библиотеки: перспективн. методы и технологии, электронные коллекции”. — Москва, 2004. — Т. 7, вып. 4. — <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4/op> — 20.12.2004.
3. Гаврилова, Т.А. Онтологический инжиниринг: от истории к практическому формированию прикладных онтологий / Т.А. Гаврилова // Когнитивные исследования: сб. науч. тр. — М.: Ин-т психологии РАН, 2006. — Вып. 2. — С. 293 — 307.
4. Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. — СПб.: Питер, 2000. — 384 с.
5. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications / H. Knublauch, R.W. Ferguson, N.F. Noy, M.A. Musen // Third Intern. Semantic Web Conf. — ISWC 2004, Hiroshima, Japan. — 2004. — P. 229 — 243.
6. К интеграции онтологий предметных областей / А.В. Палагин, А.В. Михайлюк, В.Ю. Величко, И.Г. Петренко // Inform. Models of Knowledge, ITHEA. — Kiev, Ukraine — Sofia, Bulgaria, 2010. — С. 69 — 85.
7. Noy, N.F. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment / N.F. Noy, M.A. Musen // Seventeenth Nat. Conf. on Artificial Intelligence (AAAI-2000). — Austin, TX, 2000. — P. 429 — 439.
8. Войшвилло, Е.К. Понятие как форма мышления: логико-гносеологический анализ / Е.К. Войшвилло. — М.: Изд-во МГУ, 1989. — 239 с.

References

1. Gruber, T.R. A translation approach to portable ontology specifications / T.R. Gruber // Knowledge systems laboratory, Computer Science Department. — Stanford, California, 1993. — Vol. 6, № 3. — pp. 71 — 92.
2. Ovdey, O.M. Obzor instrumentov inzhenerii ontologii [Overview of ontology engineering tools] [Electronic resource] / O.M. Ovdey, G.Yu. Proskudina // Tr. VI Vseros. nauch. konf. “Elektron. biblioteki:

- perspektivn. metody i tekhnologii, elektronnye kolleksii” [Proc. of VI All-Rus. Sci. conf. “Electron. Libraries: Prospective Methods and Technologies, Electronic Collections”]. — Moscow, 2004. — Vol. 7, iss. 4. — Available at: <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4/op> — 20.12.2004.
3. Gavrilova, T.A. Ontologicheskii inzhiniring: ot istorii k prakticheskomu formirovaniyu prikladnykh ontologiy [Ontological engineering: from history to practical formation of applied ontologies] / T.A. Gavrilova // Kognitivnye issledovaniya: sb. nauch. tr. — Moscow, 2006. — Iss. 2. — pp. 293 — 307.
 4. Gavrilova, T.A. Bazy znaniy intellektual'nykh sistem [Knowledge base of intelligent systems] / T.A. Gavrilova, V.F. Khoroshevskiy. — St. Petersburg, 2000. — 384 p.
 5. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications / H. Knublauch, R.W. Ferguson, N.F. Noy, M.A. Musen // Third Intern. Semantic Web Conf. — ISWC 2004, Hiroshima, Japan. — 2004. — pp. 229 — 243.
 6. K integratsii ontologiy predmetnykh oblastey [To the integration of domain ontologies] / A.V. Palagin, A.V. Mikhaylyuk, V.Yu. Velichko, I.G. Petrenko // Inform. Models of Knowledge, ITHEA. — Kiev, Ukraine — Sofia, Bulgaria, 2010. — pp. 69 — 85.
 7. Noy, N.F. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment / N.F. Noy, M.A. Musen // Seventeenth Nat. Conf. on Artificial Intelligence (AAAI-2000). — Austin, TX, 2000. — pp. 429 — 439.
 8. Voyshvillo, E.K. Ponyatie kak forma myshleniya: logiko–gnoseologicheskii analiz [Concept as a form of thinking: logical and epistemological analysis] / E.K. Voyshvillo. — Moscow, 1989. — 239 p.

Рецензент д-р техн. наук, проф. Одес. нац. политехн. ун-та Крисиллов В.А.

Поступила в редакцию 10 июня 2013 г.