УДК 004.738.52

ИЗВЛЕЧЕНИЕ ОНТОЛОГИЙ ИЗ WIKI-СИСТЕМ

В.К. Шестаков

Рассматривается подход к извлечению онтологий из Wiki-систем, а также его использование в разработке и сопровождении Wiki-систем, заполняемых содержимым на основе онтологий, и другие возможные применения. Описывается реализация данного подхода в виде клиентского приложения.

Ключевые слова: Wiki, онтологии, извлечение данных, Pywikipediabot, Semantic MediaWiki.

Введение

Для удовлетворения все возрастающих информационных потребностей пользователей разработаны разнообразные средства построения информационных систем. Одним из таких удобных и простых в использовании средств сбора и хранения информации являются Wiki-системы [1]. Они позволяют работать не только с текстовым, но и с мультимедийным контентом, имеют удобный и интуитивно понятный интерфейс, просты в освоении. Однако их большим недостатком является то, что они позволяют отслеживать в создаваемых информационных системах только структурную целостность ссылок, не обеспечивая при этом логической непротиворечивости и семантической согласованности (semantic consistency) [2] используемых в них понятий.

Общая идея предлагаемого подхода состоит в создании инструментария, который бы обеспечивал заполнение Wiki-систем содержимым с согласованной системой понятий (семантически согласованных Wiki). Wiki с такими свойствами можно получить, если строить ее на основе логически непротиворечивой онтологии, описывающей предметную область будущей системы. В этот инструментарий должны также входить средства контроля и отслеживания изменений в онтологии.

Первая составляющая данного подхода, заключающаяся в заполнении Wiki-систем содержимым на основе онтологий, была описана в предыдущей работе автора [3]. В настоящей работе рассматривается его вторая составляющая, обратная первой, посвященная извлечению онтологий из уже существующих Wiki-систем. Как уже говорилось, она необходима для контроля, трансформации, сопровождения и развития Wiki, которые уже наполнены содержимым. Кроме того, она может иметь и другие применения, например, построение предварительного, чернового варианта онтологии предметной области по уже существующей Wiki-системе или объединение нескольких Wiki по близким предметным областям.

Существует много подходов к извлечению знаний из Wiki-систем в виде онтологии. Чаще всего используется Википедия. На ее основе осуществляют извлечение структурированной информации и предоставление доступа к ней [4], построение онтологии верхнего уровня (general-purpose ontology) [5] и крупномасштабной онтологии людей [6], ее используют в качестве среды для разработки онтологий [7] и как источник корпуса текстов для построения онтологии конкретной предметной области [8], а также для автоматического построения крупномасштабной мультимодальной онтологии для классификации веб-изображений [9]. Все упомянутые подходы и системы используют Википедию только в качестве источника информации, а извлеченную информацию они затем используют в своих целях, например, для построения своей собственной онтологии, базы данных или классификации чего-либо.

В настоящей работе предлагается несколько другой подход. Он заключается в том, что на основе информации, извлекаемой из выбранной Wiki-системы, строится не онтология вообще (например, какой-то предметной области), а онтология именно данной конкретной Wiki. Это дает возможность получить для дальнейшего анализа и использования ее структуру и содержание.

Метод извлечения онтологий

В основе любой Wiki-системы лежит так называемый Wiki-движок – комплекс программных средств для преобразования Wiki-разметки в код, предназначенный для отображения в браузере. Одним из самых распространенных движков является MediaWiki [10] (на его основе работает широко известная Википедия). Для него существуют специальные дополнения, называемые расширениями, позволяющие получить определенную функциональность. Так, расширение Semantic MediaWiki [11, 12] дает возможность добавлять семантическую информацию за счет расширения разметки, а также предлагает средства для работы с этой информацией.

Для проведения работ технического характера в Wiki-системах используют боты – специальные программы для выполнения заданного набора операций. Они являются клиентскими приложениями, поэтому не требуют внесения изменений на стороне сервера (т.е. для их работы не нужно модифицировать движок или ставить какое-либо расширение). Например, в Википедии боты используются для таких задач, как переименование категорий и статей, расстановка интервики-ссылок (ссылок на родственные проекты), исправление ссылок, удаление спама и т.п. Для реализации ботов используются различные языки программирования, а также существуют разные библиотеки для облегчения их написания. Одной из наиболее развитых библиотек является Python WikipediaBot Framework [13]. Она использует MediaWiki API (специальный интерфейс прикладного программирования) для взаимодействия с MediaWiki-системой – авторизации, получения данных и внесения изменений.

Пояснив основные понятия, можно перейти к рассмотрению общей схемы работы инструментальной системы, которая представлена на рисунке.

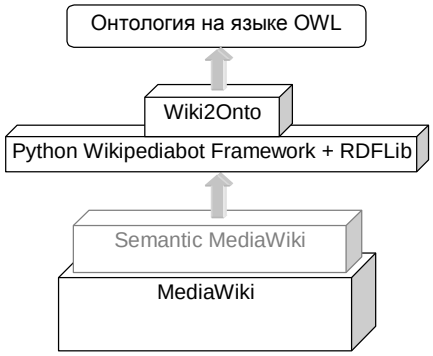

Рисунок. Общая схема работы инструментальной системы

Сначала программный модуль Wiki2Onto, разработанный в рамках данного проекта, при помощи Python WikipediaBot Framework извлекает онтологию из Wiki-системы, работающей на базе MediaWiki, возможно, с расширением Semantic MediaWiki, а затем с использованием библиотеки RDFLib [14] сохраняет в файл на языке OWL [15]. В табл. 1 представлено соответствие конструкций Semantic MediaWiki и языка OWL.

Semantic MediaWiki	**Конструкция в онтологии**
Категория	owl:Class
Подкатегория	rdfs:subClassOf
Страница	owl:NamedIndividual
Обычная ссылка	owl:ObjectProperty «Ссылается на»
Семантическая ссылка	*Зависит от типа*

Таблица 1. Соответствие конструкций Semantic MediaWiki и онтологии

Рассмотрим этот процесс более подробно. Онтология из Wiki-системы извлекается в следующем порядке. Сначала извлекаются все классы, при этом каждому классу соответствует одна категория Wiki, а структура вложенности категорий Wiki определяет иерархию классов. Затем извлекаются все страницы как экземпляры соответствующих классов. Для пустых страниц, на которые в Wiki имеются ссылки, заводится специальный служебный класс «Несуществующие страницы». После этого просматриваются все ссылки на каждой странице. Для начала определяется, является ли ссылка обычной или семантической. Если ссылка обычная, то для соответствующего экземпляра класса в OWL-онтологии заводится объектное свойство «Ссылается на» (так как ссылка обычная, а не семантическая, то у нее нет своего собственного имени, и данное имя выбрано для всех таких ссылок) со значением в виде экземпляра, имя которого совпадает с именем страницы, на которую указывает ссылка. Если ссылка семантическая, то она имеет структуру <*название свойства, значение свойства*>, и для нее сначала определяется тип ее свойства. Если свойство имеет тип «*Страница*» или его тип не указан, то в OWL-онтологии заводится объектное свойство с соответствующими именем (*название свойства*) и значением (*значение свойства*). (Заметим, что по умолчанию свойство ссылки имеет тип «*Страница*».) Если же свойство имеет какой-то другой стандартный тип, то тип свойства данных в OWL-онтологии определяется согласно табл. 2 (в качестве значения «owl:Annotation property» [16] используется соответствующий URI: «tel:» для телефонного номера [17], «mailto:» для адреса электронной почты, «http:» для URL и URI аннотации). Для пользовательских типов свойств создается собственный тип свойства данных.

Так как реализация модуля Wiki2Onto еще не доведена до финальной стадии, то пока поддерживаются не все стандартные типы свойств Semantic MediaWiki, а только те, что представлены в табл. 2. На данный момент этот модуль работает только с Wiki-системами на базе MediaWiki.

Извлекаемая онтология сохраняется в формате OWL, а не ограничивается RDFS по нескольким причинам. Во-первых, используется разделение свойств на два класса – объектные (owl:ObjectProperty) и типов данных (owl:DatatypeProperty). Во-вторых, в Semantic MediaWiki существуют ограничения на значения свойств с помощью специального свойства «Allows value» [18] и это нужно учитывать. В-третьих, один из способов проверки получаемой онтологии заключается в добавлении в нее аксиом.

Следует отметить, что Wiki-система, из которой извлекается онтология, не обязательно должна функционировать с расширением Semantic MediaWiki. Однако в случае использования Wiki без этого расширения извлекаемая онтология будет гораздо беднее, так как в ней не будет присутствовать специальная семантическая информация. В частности, нельзя будет извлечь атрибуты, разнообразие отношений также будет невелико. Правда, путем индивидуальной настройки на конкретную Wiki-систему объем извлекаемой из нее информации можно увеличить (например, если некоторые данные в ней приведены в

однотипном формате или для их представления используются шаблоны [19], то для их извлечения можно реализовать специальную функцию).

Тип свойства	Тип в онтологии
Строка	string
Число	double
Булево	boolean
Дата	dateTime
Текст	string
Код	string
Телефонный номер	owl:Annotation property
URL	owl:Annotation property
Почта	owl:Annotation property
URI аннотации	owl:Annotation property

Таблица 2. Соответствие при извлечении стандартных типов

В Semantic MediaWiki существует свой собственный встроенный инструмент для экспорта в RDF, позволяющий извлекать размеченную семантическую информацию из списка страниц. Кроме того, для извлечения сразу всех семантических данных из Wiki, оборудованной данным расширением, существует специальный внешний инструмент [20]. Но, как можно заметить, таим способом можно извлечь только явно размеченную семантическую информацию, а вся остальная останется неохваченной. Подход, описанный в настоящей работе, позволяет извлекать дополнительно часть этой информации, причем методы ее извлечения планируется развивать (в том числе за счет лингвистических средств и возможности индивидуальной подстройки под конкретную систему).

Также существует возможность оборудовать Semantic MediaWiki хранилищем RDF и хранить всю семантическую информацию в нем [21]. Это упрощает ее экспорт, но не избавляет от вышеописанного недостатка. Кроме того, этот способ сужает область применения за счет дополнительного требования на наличие хранилища. Если же на Wiki не установлено расширение Semantic MediaWiki, то в ней нет подобных встроенных средств экспорта информации в RDF.

В то же время предлагаемый подход, как указывалось выше, не требует наличия этого расширения в обязательном порядке. В качестве способа реализации было выбрано именно написание собственного модуля, работающего по принципу MediaWiki-бота, а не разработка расширения для MediaWiki для увеличения области применения. Ведь не всегда к Wiki-системе, из которой нужно извлечь онтологию, есть доступ для того, чтобы установить на ней свое расширение.

Итак, существуют подходы для извлечения таксономии из MediaWiki-систем без семантических расширений (например, в проектах YAGO и DBpedia, упомянутых во введении), есть стандартные средства для экспорта семантической информации в расширении Semantic MediaWiki, но нет систем, объединяющих в себе и то, и другое. И именно в этом заключается новизна данной работы.

Варианты применения метода извлечения онтологий

В первую очередь, метод позволяет проследить, как изменилась онтология после того, как эксперты поработали над содержимым Wiki-сайта, созданного на базе первоначальной онтологии. Это может понадобиться не только ради обычного любопытства, но и для вполне серьезных целей – например, отслеживание развития проекта, верификация получаемой онтологии (ontology verification) и ее реинжиниринг (ontology reengineering) [22], проверка качества и сбалансированности получаемой структуры данных [23], координация в развитии отдельных частей предметной области. К полученной онтологии также можно применить одну из существующих машин вывода для получения неявных знаний.

Кроме того, извлекать онтологию можно не только из той Wiki-системы, которая была ранее заполнена на основании некоторой онтологии, но и из уже существующей Wiki, заполненной обычным образом. Например, хотим построить онтологию некоторой предметной области и обнаружили Wiki, содержащую нужные нам сведения из этой области. Вместо того чтобы вручную строить нужную онтологию «с нуля», можно извлечь из этой системы ее предварительный, черновой вариант, а затем уже дорабатывать его, что гораздо проще.

Еще одно применение метода извлечения онтологии заключается в объединении нескольких Wiki-систем по близким предметным областям. Непосредственное объединение систем может быть довольно сложным и потребовать много ручной работы, так как крайне затруднительно отследить все связи и пересечения между двумя системами. Используя предлагаемый подход, можно поступить гораздо проще: извлечь онтологию из каждой системы и провести их слияние (ontology merging) [22], а после этого по объединенной онтологии заполнить содержимым требуемую Wiki-систему.

Заключение

В работе рассмотрен подход и предложен метод извлечения онтологий из Wiki-систем, разработан прототип инструментальной системы, реализующий данный метод в указанном объеме.

Этот метод позволяет не только сопровождать, контролировать и развивать Wiki-системы, наполнение информацией которых производилось на основе онтологии, но и имеет некоторые другие применения, такие как построение предварительного, чернового варианта, онтологии предметной области по уже существующей Wiki или объединение нескольких Wiki по близким предметным областям.

В дальнейшем планируется завершить реализацию модуля Wiki2Onto, а также расширить его возможности, в частности, применить методы компьютерной лингвистики для извлечения информации со страниц Wiki-систем.

Литература

1. Leuf B., Cunningham W. The Wiki Way: Quick Collaboration on the Web. – Addison-Wesley Professional, 2001. – 464 p.
2. Baader F., Nutt W. Basic Description Logics: The Description Logic Handbook. – Cambridge University Press. – 2002. – P. 47–100.
3. Шестаков В.К. Инструменты построения информационных систем на основе Wiki-технологии и онтологий предметных областей // Сборник трудов конференции «Управление знаниями и технологии семантического веба – 2010». – СПб: СПбГУ ИТМО, 2010. – С. 150–159.
4. Bizer C., Lehmann J., Kobilarov G., Auer S., Becker C., Cyganiak R., Hellmann S. DBpedia – A Crystallization Point for the Web of Data // Journal of Web Semantics: Science, Services and Agents on the World Wide Web. – 2009. – Is. 7. – P. 154–165.
5. Suchanek F.M., Kasneci G., Weikum G. YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia // Proceedings of the 16th International Conference on World Wide Web (Banff, Alberta, Canada, May 8–12, 2007). WWW '07. – NY: ACM Press, 2007. – P. 697–706.
6. Shibaki Y., Nagata M., Yamamoto K. Constructing Large-Scale Person Ontology from Wikipedia // Proceedings of the 2nd Workshop on «Collaboratively Constructed Semantic Resources». – Coling, 2010. – P. 1–9.
7. Hepp M., Bachlechner D., Siorpaes K. Harvesting Wiki Consensus – Using Wikipedia Entries as Ontology Elements // Proceedings of the First Workshop on Semantic Wikis – From Wiki to Semantics, Annual European Semantic Web Conference (ESWC 2006). – 2006. – P. 124–138.
8. Cui G.Y., Lu Q., Li W.J., Chen Y.R. Corpus Exploitation from Wikipedia for Ontology Construction // Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008). – Marrakech, 2008. – P. 2125–2132.
9. Wang H., Jiang X., Chia L.-T., Tan A.-H. Wikipedia2Onto --- Adding Wikipedia Semantics to Web Image Retrieval // Proceedings of the WebSci'09: Society On-Line, 18-20 March 2009. – Greece: Athens, 2009. – P. 297–306.
10. Сайт проекта MediaWiki [Электронный ресурс]. – Режим доступа: http://mediawiki.org, св. Яз. англ. (дата обращения 30.11.2011).
11. Сайт проекта Semantic MediaWiki [Электронный ресурс]. – Режим доступа: http://semantic-mediawiki.org, св. Яз. англ. (дата обращения 30.11.2011).
12. Krötzsch M., Vrandecic D., Völkel M., Haller H., Studer R. Semantic Wikipedia // Journal of Web Semantics. – Elsevier, 2007. – № 5. – P. 251–261.
13. Сайт проекта Python WikipediaBot Framework [Электронный ресурс]. – Режим доступа: http://pywikipediabot.sourceforge.net, свободный. Яз. англ. (дата обращения 30.11.2011).
14. Сайт проекта RDFLib [Электронный ресурс]. – Режим доступа: http://rdflib.net, св. Яз. англ. (дата обращения 30.11.2011).
15. Motik B., Patel-Schneider P.F., Parsia B., eds. OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. W3C Recommendation, 27 October, 2009 [Электронный ресурс]. – Режим доступа: http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/, св. Яз. англ. (дата обращения 30.11.2011).
16. OWL Web Ontology Language Reference [Электронный ресурс]. – Режим доступа: http://www.w3.org/TR/owl-ref/, св. Яз. англ. (дата обращения 30.11.2011).
17. The tel URI for Telephone Numbers [Электронный ресурс]. – Режим доступа: http://www.ietf.org/rfc/rfc3966.txt, св. Яз. англ. (дата обращения 30.11.2011).
18. Semantic MediaWiki Property:Allows value [Электронный ресурс]. – Режим доступа: http://semantic-mediawiki.org/wiki/Property: Allows_value, св. Яз. англ. (дата обращения 30.11.2011).
19. Википедия: Механизм шаблонов [Электронный ресурс]. – Режим доступа: http://ru.wikipedia.org/wiki/Википедия: Механизм_шаблонов, св. Яз. англ. (дата обращения 30.11.2011).
20. Semantic MediaWiki: RDF export [Электронный ресурс]. – Режим доступа: http://semantic-mediawiki.org/wiki/Help:RDF_export, св. Яз. англ. (дата обращения 30.11.2011).

21. Semantic MediaWiki: Using SPARQL and RDF stores [Электронный ресурс]. – Режим доступа: http://semantic-mediawiki.org/wiki/Help:Using_SPARQL_and_RDF_stores, св. Яз. англ. (дата обращения 30.11.2011).
22. Suarez-Figueroa M.C., Gomez-Perez A. Towards a Glossary of Activities in the Ontology Engineering Field // Proceedings of 6th International Conference on Language Resources and Evaluation (LREC'08). – Marrakech, 2008. – P. 870–873.
23. Гаврилова Т.А., Горовой В.А., Болотникова Е.С., Горелов В.В. Субъективные метрики оценки онтологий // Материалы Всероссийской конф. с межд. участием «Знания-Онтологии-Теории» (ЗОНТ-2009). Т. 1. – Новосибирск: Институт математики СО РАН, 2009. – С. 178–186.

Шестаков Владимир Константинович – Новосибирский государственный университет, аспирант, zfc@ngs.ru