

Krassimir Markov, Vitalii Velychko, Oleksy Voloshin  
(editors)

**Information Models  
of  
Knowledge**

**ITHEA<sup>®</sup>  
KIEV – SOFIA  
2010**

**Krassimir Markov, Vitalii Velychko, Oleksy Voloshin (ed.)**

**Information Models of Knowledge**

ITHEA®

Kiev, Ukraine – Sofia, Bulgaria, 2010

ISBN 978-954-16-0048-1

First edition

Recommended for publication by The Scientific Council of the Institute of Information Theories and Applications FOI ITHEA  
ITHEA IBS ISC: 19.

This book maintains articles on actual problems of research and application of information technologies, especially the new approaches, models, algorithms and methods for information modeling of knowledge in: Intelligence metasynthesis and knowledge processing in intelligent systems; Formalisms and methods of knowledge representation; Connectionism and neural nets; System analysis and synthesis; Modelling of the complex artificial systems; Image Processing and Computer Vision; Computer virtual reality; Virtual laboratories for computer-aided design; Decision support systems; Information models of knowledge of and for education; Open social info-educational platforms; Web-based educational information systems; Semantic Web Technologies; Mathematical foundations for information modeling of knowledge; Discrete mathematics; Mathematical methods for research of complex systems.

It is represented that book articles will be interesting for experts in the field of information technologies as well as for practical users.

General Sponsor: Consortium FOI Bulgaria ([www.foibg.com](http://www.foibg.com)).

Printed in Ukraine

**Copyright © 2010 All rights reserved**

© 2010 ITHEA® – Publisher; Sofia, 1000, P.O.B. 775, Bulgaria. [www.ithea.org](http://www.ithea.org) ; e-mail: [info@foibg.com](mailto:info@foibg.com)

© 2010 Krassimir Markov, Vitalii Velychko, Oleksy Voloshin – Editors

© 2010 Ina Markova – Technical editor

© 2010 For all authors in the book.

© ITHEA is a registered trade mark of FOI-COMMERCE Co., Bulgaria

**ISBN 978-954-16-0048-1**

C/o Jusautor, Sofia, 2010

## КРИТЕРИИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ ВЫБОРЕ РЕДАКТОРА ОНТОЛОГИЙ

Дмитрий Климань, Вера Любченко

**Abstract:** *The known editors of ontologies are analyzed, and their main characteristics are identified. Three decision-making criteria to choose ontology editor are formulated.*

**Keywords:** *ontology, ontology editor*

**ACM Classification Keywords:** *I.2.4 Knowledge Representation Formalisms and Methods*

---

### Введение

Представление данных в форме онтологий обеспечивает возможность их автоматической обработки, а также позволяет наглядно и четко сформулировать знания по выбранной тематике. Обучение, как процесс целенаправленной передачи знаний, является естественной областью для использования онтологий.

В [Любченко, 2009] построена онтологическая модель предметной области учебного курса, в частности построена таксономия понятий, описывающая 5 базовых понятий предметной области, и сформулировано множество аксиом, которые формально ограничивают семантику таксономии понятий на основе объяснений. Построенная модель позволяет стандартизировать интегральное представление контента учебного курса и выполнять агрегирования образовательных объектов на основе их логической структуры и разделения содержания и представления.

Очевидно, что онтологическая модель предметной области полезна только в том случае, когда она может быть повторно использована в процессе разработки новых и анализа существующих учебных курсов. Для облегчения преподавателям работы с моделью следует обеспечить возможность автоматического наполнения шаблонных структур модели содержанием. Для этих целей обычно используют редактор онтологий – приложение, поддерживающее создание онтологий или манипулирование ими, предоставляющее возможности работы с одним или несколькими языками представления онтологий, импорт/экспорт в разные форматы, доступ к библиотекам онтологий, визуализацию, машины вывода, языки запросов, поддержку онтологий верхнего уровня и другое. Актуальной является задача выбора редактора онтологий для автоматизации формирования онтологической модели предметной области учебного курса и определения критериев, влияющих на этот выбор.

Постановка задачи. Необходимо определить основные критерии принятия решений при выборе редактора онтологий в условиях определенной базовой онтологической модели.

---

### Популярные редакторы онтологий

Редакторы онтологий – сравнительно молодая группа программных приложений. Самым старым из них является *Java Ontology Editor – JOE* (<http://www.cse.sc.edu/research/cit/demos/java/joe/joeBeta-jar.html>), последняя версия которого вышла 19 мая 1998 года. Этот редактор был создан в виде Java-апплета и поддерживал минимальные онтологические конструкции, такие как сущности, отношения и атрибуты.

На сегодняшний день редакторы онтологий значительно расширили свои возможности. Наиболее удобные среди них, как правило, являются коммерческими. К таким можно отнести *Be Informed Suite* (<http://www.beinformed.nl>) – программу, разработанную голландской компанией Be Informed, которая содержит удобный графический интерфейс, алгоритмы предположения и экспорт в стандартные форматы. Она была создана в первую очередь для разработки больших приложений на основе онтологий.

Ещё один довольно крупный коммерческий редактор онтологий *LinKFactory* (<http://www.nuance.com/healthcare/products/nlp.asp>) был создан компанией Language and Computing на базе Java-технологий. Редактор имел графический интерфейс, поддержку многопользовательской работы, поддержку нескольких миллионов объектов знаний и минимальное время обработки. После того, как

Language and Computing стала составной частью Nuance Communication Inc., редактор был переименован в Natural Language Processing (NLP) и используется только в сфере медицины.

К узкоспециализированным редакторам онтологии можно отнести редакторы *myWeb* (<http://www.ontologyonline.org/main.html>), *OBO-Edit* (<http://geneontology.sourceforge.net/>) и *OntoBuilder* (<http://iew3.technion.ac.il/OntoBuilder/>). Первые два редактора предназначены только для сферы биологии. Редактор *myWeb* позволяет создавать онтологии в сфере молекулярной биологии, описывающие молекулярные функции и биологические процессы. Он имеет плагин для Firefox – *jOWL* и поддерживает выполнение запросов. *OBO-Edit* был создан в рамках Gene Ontology Consortium, он решает сходные с *myWeb* задачи, но в отличие от *myWeb* распространяется бесплатно с открытым кодом. *OntoBuilder* был разработан для применения в сфере медицины с целью создания словарей данных и реализован в виде веб-сервиса.

Среди свободно распространяемых редакторов онтологий можно выделить онлайн-редакторы, редакторы в виде расширений для Firefox и редакторы в виде отдельных приложений. Бесплатные редакторы, как правило, создаются различными университетами для использования в образовательных целях. Университет Флориды создал *MapTools Ontology Editor* (COE) (<http://coe.ihmc.us/groups/coe/>), написанный на Java и содержащий пакет программных инструментов для построения, совместного использования и просмотра онтологий, использующих Ontology Web Language (OWL). Для концептуальных карт разработан интерфейс, благодаря которому в понятной форме отображаются структура, содержимое и масштаб онтологии.

Университет города Осака, Enegate Co Ltd. и Department of Knowledge Systems (Mizoguchi Laboratory) разработали редактор *Hozo* (<http://www.hozo.jp/>), предназначенный специально для создания тяжеловесных и хорошо продуманных онтологий. Новая версия редактора поддерживает импорт OWL. С каждой новой версией редактора публикуются Hozo API ядра Hozo, что позволяет сторонним разработчикам использовать ядро при создании своих собственных Java приложений.

Похожие редакторы были созданы университетами Манчестера и Мэриленда – *OilEd* (<http://www.cs.man.ac.uk/~horrocks/Publications/download/2001/oiled-dl.pdf>) и *SWOOP* (<http://code.google.com/p/swoop/>), соответственно. Оба редактора являются приложениями, созданными на базе Java-технологии и способными отображать и редактировать OWL онтологии. Единственным отличием является поддержка редактором *OilEd* возможности многопользовательской работы. В настоящее время разработка новых версий редакторов больше не ведётся.

Стэнфордскому университету принадлежит разработка 3 редакторов онтологий – *Protégé* (<http://protege.stanford.edu/>), *Chimaera* (<http://www.ksl.stanford.edu/software/chimaera/>), *Ontolingua* (<http://www.ksl.stanford.edu/software/ontolingua/>). Наиболее известным и успешным является *Protégé*. Этот бесплатный редактор с открытым исходным кодом был создан совместно с университетом Манчестера на базе Java-технологий с использованием набора графических библиотек Swing. Платформа *Protégé* поддерживает 2 различные возможности создания онтологий – с помощью редактора *Protégé-Frames* и с помощью редактора *Protégé-OWL*. Редактор имеет и онлайн-версию под названием *WebProtégé*. Функции *Protégé* постоянно расширяются за счёт создания плагинов сторонними разработчиками. Имеется поддержка возможности экспорта созданных онтологий в различные форматы.

Два других проекта оказались менее успешны. *Chimaera* поддерживает лишь работу с форматами OWL и DARPA Agent Markup Language (DAML). Основная функция данного редактора – объединение нескольких онтологий. Он способен решить задачи реорганизации таксономии и разрешение конфликта имён. *Ontolingua* является веб-сервисом, способным создавать, редактировать и модифицировать онтологии. Сервер поддерживает 150 активных пользователей.

Очень успешным редактором, способным составить конкуренцию *Protégé*, является *TopBraid Composer* ([http://www.topquadrant.com/products/TB\\_Composer.html](http://www.topquadrant.com/products/TB_Composer.html)), созданный компанией TopQuadrant. Полная версия редактора коммерческая, но имеется также и бесплатная версия с меньшим функционалом.

Программа написана на Eclipse, имеется графический интерфейс, возможность загрузки, редактирования и сохранения RDF/XML файлов, создания и выполнения запросов SPARQL (SPARQL Protocol and RDF Query Language), импорта и преобразования в RDF данных из XML, UML, каналов RSS, Atom и реляционных баз данных, создание XML-схемы из RDF/OWL, работа с базами данных RDF, визуализация RDF данных с помощью графиков и схем и многое другое.

Стоит отметить и редактор с открытым исходным кодом KAON (KArlsruhe ONtology) (<http://kaon.semanticweb.org/>), созданный университетом Карлсруэ. KAON является редактором онтологий, ориентированных на бизнес-приложения. Включает в себя полный набор инструментов, позволяющих легко создавать онтологии. Стойкость механизмов KAON основывается на реляционных базах данных.

---

### Ключевые характеристики редакторов онтологий

---

Анализ наиболее популярных редакторов онтологий позволяет заметить, что все они созданы с использованием Java-технологии, что обеспечивает возможность их использования на различных программных платформах.

Кроме этого все редакторы обеспечивают графический интерфейс для редактирования и навигации. Наличие удобного графического интерфейса с набором простых форм для редактирования и простого меню позволяет завоевать популярность у пользователей.

Большинство редакторов поддерживает несколько форматов описания онтологий, что обусловлено отсутствием единого широко используемого формата и оставляет право выбора за пользователем.

Следует отметить, что в последнее время становится обязательным для редакторов поддерживать выполнение логических выводов, а именно выполнять проверку непротиворечивости построенной модели и обеспечивать возможность выполнения запросов.

Положительным фактором является использование в редакторах онтологий реляционных баз данных, использование которых позволяет сделать работу редактора более стойкой, увеличить скорость работы редактора, значительно повысить максимальное количество объектов, с которыми способен работать редактор и уменьшить время восстановления.

Следует отметить тенденцию к обеспечению поддержки многопользовательского режима, что особенно актуально в крупных проектах. Но отсутствие этой возможности нельзя назвать критической.

Основные характеристики проанализированных редакторов приведены в табл.1.

Анализ таблицы позволяет сформулировать три критерия принятия решений при выборе редактора онтологий:

1. Если определена базовая онтологическая модель, то следует использовать специализированные критерии.
2. Если необходимо обеспечить поддержку и возможность выполнения импорта/ экспорта онтологий в различные форматы, то в качестве альтернатив следует рассматривать универсальные редакторы. Для специализированных редакторов характерна поддержка ограниченного количества форматов, как правило, формата расширяемого языка разметки (XML) и формата языка онтологии для Интернет (OWL-RDF).
3. Если необходимо обеспечить возможность выполнения запросов к онтологической базе знаний, то следует использовать специализированный редактор. В силу того, что специализированные редакторы разрабатываются на основе базовой онтологической модели, существует возможность обеспечить эффективный механизм выполнения запросов.

Таблица 1

Редактор онтологий	Поддерживаемые форматы	Наличие базовой модели	Проверка непротиворечивости	Выполнение запросов	Многопользовательский режим	Открытый код
Универсальные редакторы						
Be Informed Suite	XML; RDF; OWL; XTM; LKIF	-	-	+	-	-
СmapTools Ontology Editor	XML; RDF; OWL	-	+	-	-	-
Hozo	XML; RDF; OWL; CSV; DAML	-	+	+	-	-
KAON	RDF(S); OWL	-	+	+	-	+
OiiEd	DAML+OIL/OWL; RDFS; SHIQ	-	-	-	+	-
Ontolingua with Chimaera	DAML+OIL; KIF; OKBC; Loom; Prolog; Ontolingua; CLIPS	-	-	-	-	-
Protégé	RDF(S) ; OWL; XML Schema; RDB schema	-	-	+	-	+
SWOOP	OWL	-	+	+	-	+
TopBraid Composer	XML; RDF; OWL; RDBM; RSS/Atom; UML; HTML	-	+	+	-	-
Специализированные редакторы						
LinKFactory	XML; RDF(S); DAML+ OIL/OWL	+	+	+	+	-
myWeb	OWL-RDFS	+	-	+	-	-
OBO-Edit	XML; RDF	+	-	+	-	+
OntoBuilder	XML; RDF; HTML	+	-	+	-	-

### Заключение

Для определения критериев принятия решений при выборе редактора онтологий в условиях существующей базовой онтологической модели были проанализированы характеристики наиболее популярных редакторов онтологий. Анализ показал, что все редакторы можно разбить на две группы: универсальные редакторы, которые не привязаны к конкретной предметной области и конкретному формату представления, и специализированные редакторы, которые разрабатываются на основе существующей базовой онтологической модели. Таким образом, для работы с онтологической моделью предметной области учебного курса необходимо использовать специализированный редактор онтологий.

При разработке этого редактора не стоит обеспечивать поддержку импорта/экспорта в различные форматы. Если учесть, что результаты работы этого редактора онтологий должны использоваться различными информационными системами, точный состав которых и требования к формату входных/выходных данных неизвестны, то данное утверждение является обоснованным. При уточнении требований проблему несовместимости форматов можно будет решить программными средствами, путем введения мостов или оберток.

Следует обратить внимание на то, что при разработке редактора онтологической модели предметной области учебного курса в обязательном порядке должен быть обеспечен механизм выполнения запросов к модели. Кроме того, желательно обеспечить поддержку проверки непротиворечивости формируемой модели.

---

## Литература

[Любченко, 2009] Любченко В.В. Онтологічна модель знань про предметні області навчальних курсів // IX міжнародна наукова конференція ім. Т.А.Таран «Інтелектуальний аналіз інформації ІАІІ-2009», Київ, 19-22 мая 2009 г.: сб.тр. — К.: Просвета, 2009. — С. 269 — 275.

---

## Информация об авторах



**Дмитрий Климань** – бакалавр, Одесский национальный политехнический университет, пр. Шевченко, 1, Одесса, 65044, Украина; e-mail: 2842150@gmail.com

Научные интересы: системы онлайн-обучения, онтологии



**Вера Любченко** – доцент, Одесский национальный политехнический университет, пр. Шевченко, 1, Одесса, 65044, Украина; e-mail: vira.lyubchenko@gmail.com

Научные интересы: системы электронного обучения, системы поддержки принятия решений, формальные методы для разработки программного обеспечения