

# ОНТОРЕДАКТОР КАК КОМПЛЕКСНЫЙ ИНСТРУМЕНТ ОНТОЛОГИЧЕСКОЙ ИНЖЕНЕРИИ<sup>1</sup>

## ONTOLOGY EDITOR AS INTEGRATED DEVELOPMENT ENVIRONMENT

Рубашкин В. Ш. (*VRubashkin@yandex.ru*), Пивоварова Л. М. (*pivovarova@iphil.ru*)

Санкт-Петербургский государственный Университет

В докладе представлен опыт разработки и использования онторедатора, ориентированного на модель знаний онтологии *InTez*. Рассматриваются функции просмотра, ввода и редактирования, тестирования и др. Проводится сопоставление с зарубежным опытом аналогичных разработок.

Онторедаторы представляют сравнительно новый вид информационных технологий; требования к ним и представления об их функциональности еще только формируются. Разрабатываемые средства для работы с онтологиями весьма разнородны: они могут быть ориентированы на определенную модель знаний; иметь многомодульную или интегрированную архитектуру; поддерживать тот или иной набор функций, использовать разные методы и технологии. Несомненно, однако, что критическая масса результатов уже налицо;<sup>2</sup> движение в сторону унификации, как и в сторону объединения разных по назначению инструментов в интегрированный комплексный продукт (*integrated development environment - IDE*) достаточно хорошо различимо. Цель настоящего доклада – попытаться обозначить общие тенденции и сформулировать некоторую, как мы надеемся, последовательную концепцию построения такого рода инструментов. При этом авторы опираются – не в последнюю очередь – и на собственный опыт (онторедатор *InTez*) и иллюстрируют им возможность предлагаемых решений. Доклад не является обзором конкретных онторедаторов – существующие обзоры (см. [1, Ch 5], [2, Part II], [3]) в своей совокупности дают достаточно полную картину сложившегося в этой области исследований и разработок положения.

### 1. Общие замечания

Границы понятия "онтология" разными авторами проводятся по-разному (ср., напр., [1, Ch 5]). Не имея возможности здесь входить в обсуждение этого вопроса, обозначим коротко то понимание, которое далее будет иметься в виду.<sup>3</sup>

1. Онтология есть формальная модель **лексической системы** профессионального языка. Единицей описания в онтологии является **понятие**; термин *концепт* мы будем употреблять просто как его синоним.
2. Онтология базируется на некоторой **модели знаний**. Под моделью знаний мы понимаем язык представления знаний (ЯПЗ) вместе с некоторым набором схем аксиом, определяющих возможности системы вывода. Существенным аспектом модели знаний является принятая в ней система **категоризации** понятий. Весьма желательной является возможность логической интерпретации конструкций используемого ЯПЗ.
3. Онтология обладает вычислительной функциональностью. Можно считать, что эта функциональность воплощена в онтологическом API, реализующем некоторый доступный любым приложениям набор программных функций. Среди них обязательно присутствуют функции, реализующие процедуры ограниченного логического вывода.

Многие характеристики онторедатора, включая его функциональные возможности, существенно зависят от базовой модели знаний, принимаемой в том классе онтологий, на который редактор ориентирован. В определенной степени характеристики и возможности онторедатора зависят также от используемой операционной среды (СУБД, XML,

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (проект № 06-06-80434)

<sup>2</sup> В обзоре [3], например, дано краткое описание 93-х онторедаторов.

<sup>3</sup> Подробнее см. в [4].

текстовый процессор и др.) и диктуемого ею представления данных. В частности, выбор, например, в онторедакторе *InTez* в качестве базовой операционной среды реляционной СУБД и естественной для нее SQL-техники манипулирования данными практически предопределяют способ реализации таких функций как поиск и формирование выборок; реализация других функций существенно опирается на возможности такой операционной среды.

Таким образом, можно сказать, что онтология (понимаемая как информационно-вычислительный ресурс) предоставляет **программный интерфейс** приложениям; онторедактор реализует **человеко-машинный интерфейс**, обеспечивающий администрирование онтологий; для реализации части функций онторедактора должна использоваться функциональность самой онтологии.

## 2. Функциональность онторедактора

Вполне очевидны такие функции как навигация, броузинг и поиск; ввод и редактирование. В силу специфики инструмента к ним прибавляются другие, в том или ином виде реализуемые в разных проектах: поддержка (или использование) машины ограниченного вывода (*reasoner, inference engine*); средства тестирования онтологии.

Есть и другие аспекты, так или иначе определяющие характер функционирования онторедактора:

- организация взаимодействия с пользователем (включая наличие графического интерфейса);
- возможности и средства доменного редактирования и интеграции разнородных концептуальных систем; основным условием здесь является наличие встроенной онтологии верхнего уровня (*Top-Level Ontology*), без которой, как нам представляется, онторедактор теряет способность объединять и интегрировать концептуальные модели разных предметных / проблемных областей;
- средства и способы представления экземпляров, являющихся "примерами" (instance) концептов онтологии; способы работы с "описаниями экземпляров".

## 3. Навигация, броузинг и поиск

Специфика онторедактора такова, что даже эти вполне традиционные для любого редактора функции требуют обсуждения. Просмотр и навигация предполагают, прежде всего, некоторую "естественную" упорядоченность материала. В текстовом редакторе смысл этого выражения вполне очевиден – это порядок следования слов и предложений в тексте. Применительно к онторедактору, мы, возможно, склонимся к выводу, что естественного порядка в концептуальной модели вообще не существует. Действительно, мы можем говорить о физическом порядке следования записей, об упорядоченности по ключу, или об алфавитном порядке терминов, но все это, с точки зрения концептуальной модели, не касается существа представляемого онторедактором материала. Алфавитный порядок имеет значение, но, скорее, как поисковый индекс, обеспечивающий быстрый поиск нужного пользователю термина. В такой ситуации поиск, формирование выборок (фильтры) и навигация по связям разного типа оказывается существенной поддержкой для реализации комфортного взаимодействия пользователя с концептуальной системой. "Естественной" для концептуальной системы можно считать, скорее, таксономическую (*общее - частное*) упорядоченность концептов; она образует ядро всякой концептуальной модели. Так что "естественным" порядком просмотра и навигации здесь скорее является просмотр "сверху вниз" (от общего к частному). А также, возможно, просмотр групп концептов связанных иерархическими связями другого типа (например, *целое - часть*). Но иерархическая упорядоченность не является линейной, и уже одно это порождает совсем другие требования к интерфейсу. В частности, возникает потребность графического

представления всех или некоторых связей между концептами и поддержки процедур графического редактирования, - что становится стандартом де-факто для такого рода инструментов.

Помимо перечисленного, к этой группе функций следует отнести операции, позволяющие устанавливать взаимное соответствие концептов и единиц ЕЯ: получение множества слов / словосочетаний, выражающих данное понятие (список синонимов), либо множества концептов, которые может выражать отдельно или совместно с другими словами данное слово – омонимия.

Возможности онторедатора *InTez* по этой группе функций можно дополнительно охарактеризовать следующими замечаниями. В главное окно онторедатора (см. рис. 1) всегда выводится дерево признаков (аналог таксономической иерархии в используемой модели знаний). Опционально выводятся таблицы, представляющие общий список дескрипторов, и набор специализированных вкладок. Дополнительно может быть выведено окно словарной статьи текущего или вновь вводимого концепта, где наименования и значения словарных характеристик представлены в вербальной форме. Имеется возможность поиска термина и синонимов по строке-образцу (с использованием простейших регулярных выражений) – с последующим последовательным просмотром всех найденных, а также возможность поиска концепта по ключу. В строке состояния при этом отображается число найденных по образцу концептов.

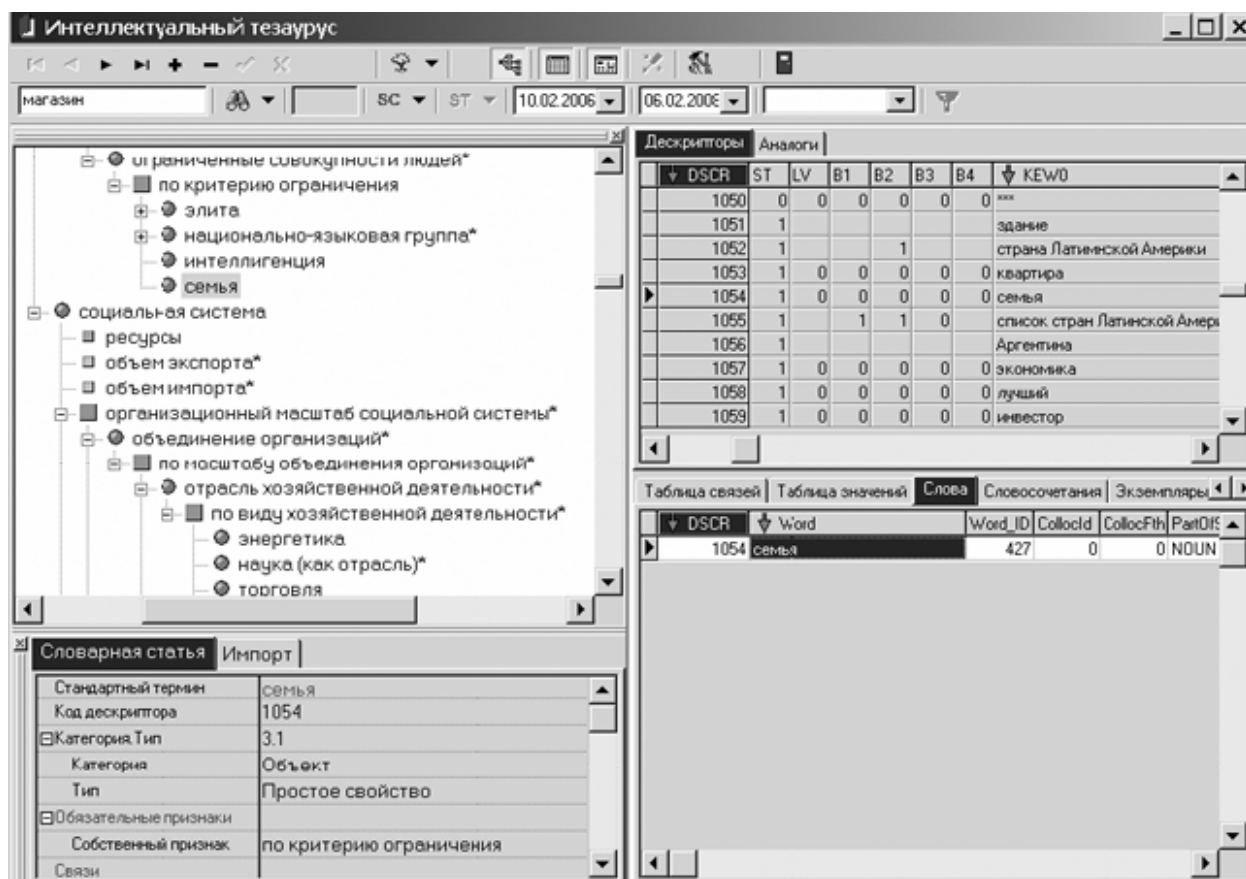


Рис.1 Главное окно онторедатора *InTez*

Графический интерфейс реализуется посредством стандартного объекта *TreeView*, соответственно, пользователю доступна вся его функциональность – как в отношении просмотра, так и в отношении графического редактирования. При этом обеспечивается графическое различие разных категорий концептов, отображаемых в дереве признаков (классификационные, количественные, строковые признаки, наименования групп

признаков, базовые свойства). Для удобства просмотра к стандартной функциональности *TreeView* добавлена опция "Показать куст", позволяющая компактно вывести на экран всех "братьев" (*sibling nodes*) указанного узла.

На альтернативных вкладках можно просматривать и редактировать:

- связи текущего концепта;
- представляющие его слова и словосочетания ("Лексикон");
- тексты определений.

При просмотре связей пользователь имеет возможность выбрать просмотр исходящих либо входящих связей, а также осуществлять переход по связям к любому из ассоциированных концептов. При работе с Лексиконом доступны обе указанные выше опции ("синонимия" и "омонимия").

При необходимости могут быть установлены фильтры, ограничивающие доступ и просмотр только концептами определенной категории и типа (семантический класс - подкласс), а также фильтр по параметрам администрирования (имя администратора и/или дата редактирования).

#### 4. Ввод и редактирование

Словарная статья концепта всегда включает **унарные характеристики** концепта и **отношения**, связывающие его с другими концептами. При этом нужно различать 2 типа отношений:

а) **Бинарные отношения** концепт – концепт. На языке логики они суть не что иное как представление логических постулатов значения на языке описания словарных статей. ("*тигр - хищник*"; "*травоядное – не хищник*"; "*всякое животное имеет голову*"; "условие применимости признака *должность – концепт работающий по найму*" и т.п.) Следует различить два типа таких отношений: объемные (*включение, совместимость, несовместимость*) и все прочие; последние – по аналогии с терминологией, сложившейся в практике разработки традиционных информационно-поисковых тезаурусов – можно именовать *ассоциативными*.

б) В онтологии *InTez* представлены также **дефиниционные отношения** – отношения между определяемым концептом и концептами, входящими в формальное толкование определяемого (предварительно должны быть специфицированы допустимые схемы формальных толкований).

Кроме того, словарные характеристики могут быть обязательными либо необязательными, повторяющимися либо уникальными.

В современной онтологической инженерии рассматриваются три возможных способа пополнения онтологий:

- а) "ручной" ввод;
- б) автоматический или автоматизированный ввод на основе анализа корпуса текстов;
- с) автоматический или автоматизированный ввод с использованием традиционной лексикографической информации (энциклопедических и толковых словарей).

Поскольку варианты, указанные в п.п. б) и с) фактически представляют собой высокоспециализированные и не достигшие еще достаточной зрелости технологии,<sup>4</sup> здесь мы будем обсуждать только процедуры ввода в смысле п. а) - ввод в собственном смысле слова.

Онтология, рассматриваемая как информационно-вычислительный ресурс для поддержки широкого спектра интеллектуальных информационных технологий, предъявляет жесткие требования к достоверности ввода. Это, собственно говоря, и есть основная проблема, которая должна решаться при проектировании процедур ввода. Другая актуальная в любых системах ввода и редактирования проблема – проблема эргономичности – здесь

<sup>4</sup> В англоязычной литературе они объединяются термином *ontology learning* (ср. [1, § 3.5]).

тесно связана с первой, и способ и качество решения второй в значительной степени зависит от способов и качества решения первой.

Требование достоверности ввода может быть конкретизировано в следующих пунктах

1) Неизбыточность и полнота описания – должны быть определены те и только те словарные признаки, которые релевантны для концептов данного типа.

2) Непротиворечивость описания – словарные характеристики не должны противоречить друг другу. Скажем, для концепта, определяемого конъюнкцией объектных классов (в терминах онтологии *InTez* - *И-толкование*; в терминах многих других онтологий – класс, характеризуемый через множественное наследование), определяющие концепты должны быть *совместимы* (в терминах *OWL* – не должны находиться в отношении *Disjoint*). Так что процедура ввода должна обнаруживать и блокировать ввод, например, И-толкования вида *X*  $\equiv$  *животное* *And* *металлический*.

3) Правильность означивания – вводимые значения определяемых словарных признаков должны принадлежать области их допустимых значений. Скажем, формально неправильным будет указание в качестве базового признака для единицы измерения *метр* концепта *перемещение* (имеем легко контролируруемую категориальную ошибку – базовым признаком может быть только концепт класса *сочетающийся с числом*; правильно будет *линейный размер*). Однако ошибка, состоящая в указании в той же ситуации в качестве базового признака концепта *масса*, уже не является формально контролируемой и может оставаться не выявленной до тех пор, пока онтология не начнет использоваться в приложениях, для которых именно эта связь окажется существенной. Если допустить, например, что концепт *лед* администратор пытается определить как конъюнкцию концептов *агрегатное состояние* и *химический состав*, будет обнаружена формальная ошибка, состоящая в том, что формальное толкование типа "конъюнкция" для объектного термина может содержать только объектные термины, либо означенные признаки. Однако определение типа *лед*  $\equiv$  *квазиобъект* *And* *цилиндрической формы* уже не содержит ошибок такого рода и является формально правильным.

4) Содержательная правильность – вводимые словарные характеристики должны быть адекватны смыслу добавляемого или редактируемого концепта. (Скажем, ошибкой этого типа будет отнесение администратором признака '*цвет*' к группе *химические свойства вещества*; такого же рода ошибки демонстрируют примеры п. 3) ).

Конечная цель при проектировании процедур ввода состоит в том, чтобы **полностью исключить** формально определяемые ошибки, т.е. ошибки, соответствующие п.п. 1), 2) и 3). При этом технологически "хорошее" решение будет состоять не в том, чтобы уметь обнаруживать формальные ошибки *post factum*, а в том, чтобы сама процедура ввода была спроектирована так, что ввод логически некорректных элементов описания оказывается вообще невозможным. Это означает, что функцию контроля формальной корректности словарных описаний мы полагаем правильным переместить из подсистемы тестирования, куда она помещается сейчас большинством разработчиков онторедкторов<sup>5</sup>, в подсистему ввода.

Что касается содержательных ошибок, то они могут возникать в силу случайной описки или неверно выполненного действия администратора, либо вследствие неполного или неправильного понимания им смысла вводимого концепта, как и концептов отнесения, связи с которыми фиксируются при определении вводимого концепта. Выявление такого рода ошибок представляет сложную и вряд ли окончательно и полностью разрешимую проблему для службы администрирования онтологии. Эта задача находится в компетенции подсистемы тестирования онтологии.

Существенно, что решение задач формального контроля обусловлено возможностью построить формальное описание системы словарных признаков. Такое сводится к

---

<sup>5</sup> Ср. напр. [6, p 50]: "...standard service that is offered by reasoners is consistency checking".

определению области значений каждого признака и к установлению отношений зависимости по условиям применимости между признаками. С точки зрения первого требования признаки можно разделить на признаки со стандартной областью значения (вещественные, целочисленные, строковые) – здесь процедура формального контроля значения тривиальна, - и признаки, областью значений которых является некоторый класс концептов. Здесь важен выбор адекватной данной задаче схемы категоризации концептов.

Определение связей по условиям применимости в онтологии *InTez* соответствует схеме аксиом вида

$$\forall x (\exists v (P(x, v) \rightarrow U(x)),$$

где  $P$  - некоторый словарный признак,  $U$  – концепт, представляющий условие его применимости. Если принять в онтологии ограничение, что  $U$  – всегда есть значение некоторого другого по отношению к  $P$  классификационного признака, получаем структуру типа "дерево признаков" [5], для которой перебор всех релевантных признаков представляет собой алгоритмически простую задачу обхода дерева с двумя типами вершин. При этом система словарных признаков может быть включена в саму онтологию (добавлением узла *концепт* и поддерева конкретизирующих это понятие классификационных признаков). Таким образом онтология наделяется способностью **самоописания**.

## 5. Тестирование

Проверить содержательную правильность описаний концептов в рамках технологии администрирования онтологии – помимо прямого просмотра словарных статей – можно только путем организации "лабораторных" испытаний и экспертной оценки их результатов администратором. Понятно, что окончательную проверку и отладку ("бета-тестирование") онтология может пройти в рамках целевых информационных технологий, скажем, в процедурах ее использования в системах анализа текста.

"Тестирование" отдельных концептов сводится к просмотру и проверке содержимого словарных статей и, следовательно, относится к компетенции подсистемы навигации и броузинга. Собственно тестирование как отличающаяся от броузинга процедура может состоять только в тестировании **отношений** между концептами – как объемных, так и ассоциативных.

В онторедакторе *InTez* тестирование выполняется в отдельном окне (рис. 2); результаты тестирования представляются в графической (представление объемных отношений диаграммами Венна) и текстовой (представление ассоциативных связей для тестируемой пары концептов) формах. Интерфейс позволяет эксперту оценить результат тестирования, так сказать, "одним взглядом". Возможно как "точечное" тестирование конкретной пары концептов, указываемых с использованием средств навигации, так и "серийное" тестирование, при котором пары концептов выбираются из онтологии случайным образом. Последний режим позволяет производить поиск ошибок путем быстрого "листания" произвольно выбираемых пар концептов. Поскольку исполнительная система онтологии интегрирована с онторедактором, нет необходимости для запуска процедуры тестирования дополнительно загружать и инициировать соответствующий модуль.

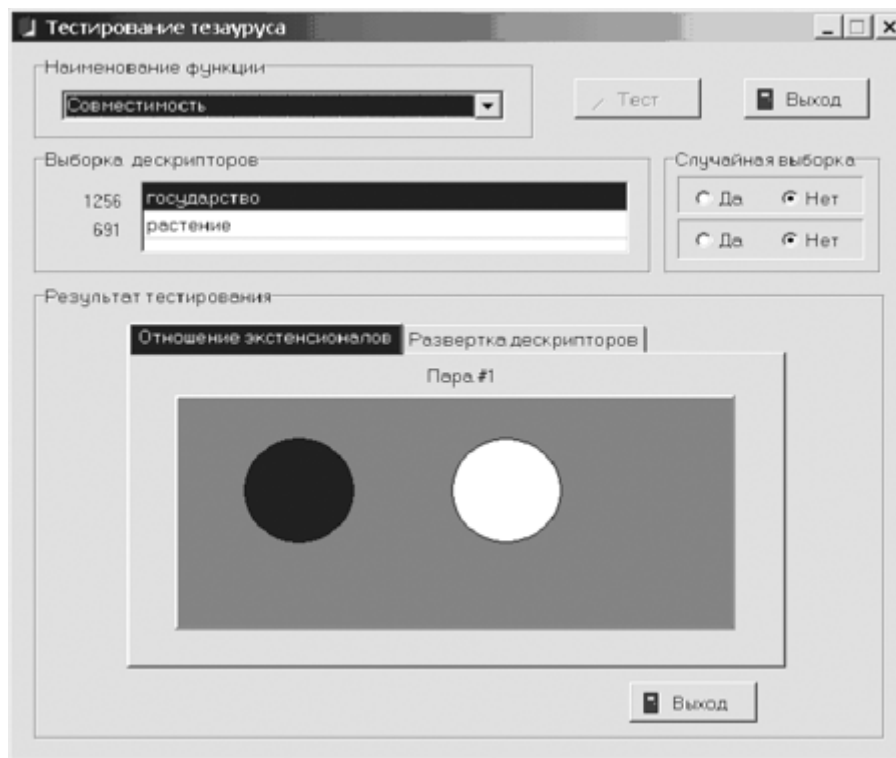


Рис. 2 Окно тестирования онтологии

## 6. Представление и работа с экземплярами

Во многих онторедаторах *экземпляры (индивиды)*, во-первых, категориально отграничиваются от *концептов*<sup>6</sup> и, во-вторых, описываются другим набором словарных характеристик. Для этого можно найти основания в самой модели знаний: *общие понятия* характеризуются **применимостью / неприменимостью** признаков ("атрибутов"), тогда как *экземпляры* требуют указания **значений** применимых к соответствующему общему концепту признаков. В самой общей форме: в терминах общих понятий формулируются законы, в терминах экземпляров – факты. Но с другой стороны, экземпляры являются конкретизацией соответствующих им общих понятий и наследуют всю относящуюся к последним словарную информацию. (Скажем, *Бразилия* унаследует от концепта *страна* информацию о том, что это понятие есть подкласс понятий *регион* и *социальный субъект*, характеризуется *численностью населения* и *размером территории*, имеет *столицу* и т.д.). На этом основании – и именно такое решение принято в онтологии *InTez* – экземпляры могут быть на общих основаниях включены в иерархию классов (с сохранением информации об "экземплярности"), будучи представлены в ней терминальными узлами.<sup>7</sup> Для удобства просмотра дерева имеется опция, позволяющая показать или скрыть все экземпляры.

В дальнейшем планируется интеграция онторедатора и поддерживаемой им онтологии с системой реляционных БД, в которых фактографическая информация об экземплярах должна храниться обычным образом в табличных записях. Имеется в виду, что при этом поддерживается связь между схемой БД (имена таблиц и полей, межтабличные связи) и онтологией таким образом, что все элементы схемы получают **концептуальную интерпретацию**. Таким способом предполагается придать онторедатору и поддерживаемой им онтологии функции интеллектуального интерфейса баз данных, обеспечивающего *прозрачный для смысла (sense transparent)* вербальный доступ к БД.<sup>8</sup>

<sup>6</sup> Термин *концепт* при этом употребляется как синоним выражения *общее понятие*.

<sup>7</sup> Не все терминальные узлы являются *экземплярными*.

<sup>8</sup> Одна из ранних попыток такого рода интеграции описана в [7, §7.2].

## Литература

1. Gomez-Perez A., Fernando-Lopez M., Corcho O. *Ontology Engineering* // Springer–Verlag, 2004.
2. Staab S., Studer R. (eds). *Handbook on Ontologies*. // Springer—Verlag, 2004.
3. Denny M. *Ontology Tools Survey, Revisited* // [Электронный ресурс]: <http://www.xml.com/pub/a/2004/07/14/onto.html> - 2004.
4. Рубашкин В. Ш. *Онтологии – концептуальные границы, проблемы и решения. Точка зрения разработчика* // *Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции "Диалог 2007"*. М.: Издательский центр РГГУ, 2007. С. 481 – 485.
5. Рубашкин В. Ш. *Представление и анализ смысла в интеллектуальных информационных системах* // М.: Наука, 1989.
6. Horridge M. et al. *A Practical Guide To Building OWL Ontologies Using The Protege-OWL Plugin and CO-ODE Tools Edition 1.0* // University Of Manchester, 2004.
7. Nirenburg S., Raskin V. *Ontological Semantics* // Cambridge, MA: MIT Press, 2004.