УДК 004.415

АНАЛИЗ КОМПЛЕКСНЫХ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ИНЖЕНЕРИИ ОНТОЛОГИЙ

Малахов К.С., Семенков В.В.

ANALYSIS OF THE COMPLEX SOFTWARE SYSTEMS FOR ONTOLOGICAL ENGINEERING PURPOSE

Malahov K., Semenkov V.

В статье представлен обзор актуальных специализированных инструментальных средств инженерии онтологий (Инструментальный комплекс онтологического назначения, ЛоТА, SIMER+MIR, ИСИДА-Т, Protégé, Ontolingua, InTez, OntoSTUDIO, OntoEdit) для построения и объединения онтологий, а также средств аннотирования на основе онтологий. Рассмотрены основные функции и возможности данных инструментальных средств, их достоинства и недостатки, а также дан системный сравнительный анализ.

Ключевые слова: *онтология, инструментальный комплекс онтологического назначения, ИКОН, специализированное инструментальное средство, инженерия знаний, проектирование дисциплины*

Введение. Методология проектирования онтологии предметной области (ПдО) [1] предполагает формирование множеств концептов, отношений, функций интерпретации и аксиом. Построение указанных множеств вручную является трудоёмким процессом, как по времени, так и по количеству вовлечённых в процесс проектирования высококвалифицированных специалистов. Ручное проектирование онтологий мало чем отличается от проектирования экспертных систем.

Понимание важности создания инструментальных средств поддержки процесса проектирования онтологии заданной ПдО пришло практически одновременно с принятием парадигмы компьютерных онтологий. В настоящее время известно более ста инструментальных программных систем [2, 3, 4], но количество комплексных программных систем, включающих редактор онтологических структур, автоматизированное построение онтологий ПдО, поверхностный семантический анализ текстовых документов и т.д. ограничено. Ниже будут рассмотрены характеристики комплексных программных систем.

Поддерживаемые формализмы и форматы представления. Под формализмом понимается некоторая формальная теория, лежащая в основе способа представления онтологических знаний (логика предикатов, фреймовые модели, дескриптивная логика, концептуальные графы и др.). Формализм существенно влияет на внутренние (компьютерные) структуры данных и может определять их формат представления.

Формат представления онтологий задаёт вид их хранения в библиотеке и способ передачи онтологических описаний другим потребителям. В качестве форматов разработаны языки представления онтологий, наиболее известными из которых являются OWL, RDFS, KIF.

Некоторые из известных редакторов онтологий поддерживают работу с несколькими формализмами представления, но только один формализм и формат являются предпочтительными для конкретного редактора [4].

Функциональность. Является одной из самых важных характеристик редакторов онтологий, под которой понимается множество предоставляемых пользователю сценариев работы с онтологическими структурами.

Базовый набор функций обеспечивает:

- работу с одним или несколькими проектами одновременно;
- графический интерфейс с пользователем;
- редактирование онтологии (создание, редактирование, удаление концептов, отношений, аксиом и прочих структурных элементов онтологии);

Архитектура приложения, место хранения онтологий, язык программного обеспечения, интерфейс пользователя, доступность.

Дополнительные возможности. К ним относят поддержку языка запросов, анализ целостности, использование механизма логического вывода, поддержку удалённого доступа через Интернет, документирование.

Известны три группы инструментальных средств (ИнС) онтологического инжиниринга [5]. К первой группе относят инструменты создания онтологий, которые предполагают поддержку совместной разработки и просмотра, создание онтологии в соответствии с заданной

(произвольной) методологией, поддержку рассуждений. Ко второй группе относят инструменты объединения, отображения и выравнивания онтологий.

Объединение предполагает нахождение сходств и различий между исходными онтологиями и создание результирующей онтологии, которая содержит элементы исходных онтологий. Для этого ИнС автоматически определяют соответствия между концептами или обеспечивают графическую среду, в которой пользователь сам находит эти соответствия.

Процедура отображения заключается в нахождении семантических связей разных онтологий.

Процедура выравнивания онтологий устанавливает различные виды соответствия между двумя онтологиями, информация которых сохраняется для дальнейшего использования в приложениях пользователя [6].

К третьей группе относят инструменты для аннотирования Web-ресурсов на основе онтологий.

Содержательный обзор известных инструментов инженерии онтологий, в котором рассмотрены основные функции и возможности ИнС, их достоинства, недостатки, сравнительный анализ и описание известных доступных онторедакторов также приведен в [5, 7, 8, 9].

Общими недостатками известных инструментальных средств являются:

- отсутствие процедур автоматического (автоматизированного) формирования компонент онтологии;
- англоязычный интерфейс с пользователем, в котором (для большинства ИнС) не предусмотрено присвоение имён компонентам онтологии на русском или украинском языке;
- структуризация концептов выполняется только по одному типу отношений;
- для большинства общедоступных ИнС не предусмотрена работа с большими по объёму онтологиями (например, для OntoEditFree – до 50 концептов);
- большинство инструментов хранит свои онтологии в текстовых файлах, что ограничивает скорость доступа к онтологиям;
- задекларированные функциональные возможности для общедоступных инструментов зачастую так и остаются нереализованными;
- недостаток информации для пользователей в инструкциях.

Обобщённая архитектурно-структурная организация инструментальных средств систем извлечения информации из текстов. Извлечение информации (Information Extraction) [1] — это подход, который позволяет сузить круг задач, требующих специфического предметно-ориентированного решения при анализе текста. В рамках этого подхода задача обработки текста ограничена распознаванием множества классов ключевых понятий конкретной предметной области и игнорированием всякой другой информации.

Несмотря на то, что системы извлечения информации могут строиться для выполнения различных задач, подчас сильно отличающихся друг от друга, существует компоненты, которые можно выделить практически в каждой системе.

В состав почти каждой системы извлечения информации входят четыре основных компонента (рис. 1), а именно: компонент разбиения на лексемы, некоторый тип лексического или морфологического анализа, синтаксический анализ (микро- и макроуровень), модуль извлечения информации и модуль для анализа на уровне конкретной предметной области. В зависимости от требований к конкретному программному продукту, в приведённую выше схему добавляют дополнительные модули анализа (специальная обработка составных слов; устранение омонимии; выделение составных типов, которое может также быть реализовано на языке правил извлечения информации; объединение частичных результатов).

Разбиение на слова при анализе европейских языков не является проблемой, поскольку слова отделяются друг от друга пробелом (или знаками препинания). Тем не менее, для обработки составных слов, аббревиатур, буквенно-цифровых комплексов и ряда других особых случаев требуются специфические алгоритмы. С границами предложений, как правило, тоже больших проблем не возникает. Однако при анализе таких языков, как японский или китайский, определение границ слова на основе орфографии невозможно. По этой причине системы извлечения информации, работающие с такими языками, должны быть дополнены модулем сегментирования текста на слова.

В некоторые системы наряду с обычными средствами лексического и морфологического анализа могут быть включены модули для определения и категоризации атрибутов частей речи, смысловых нагрузок слов, имен или других нетривиальных лексических единиц.

Специализированная оболочка интеллектуальной системы для сложно-структурированных предметных областей (Артемьева И.Л.)

Универсальные и специализированные оболочки являются средством, упрощающим процесс создания интеллектуальной системы. Универсальные оболочки основаны на использовании некоторого универсального языка представления знаний. В специализированных оболочках при представлении знаний используется специфичная для предметной области схема, определяемая онтологией той области, для которой создается оболочка, что позволяет создавать базу знаний эксперту предметной области без участия посредника, которым является инженер знаний.

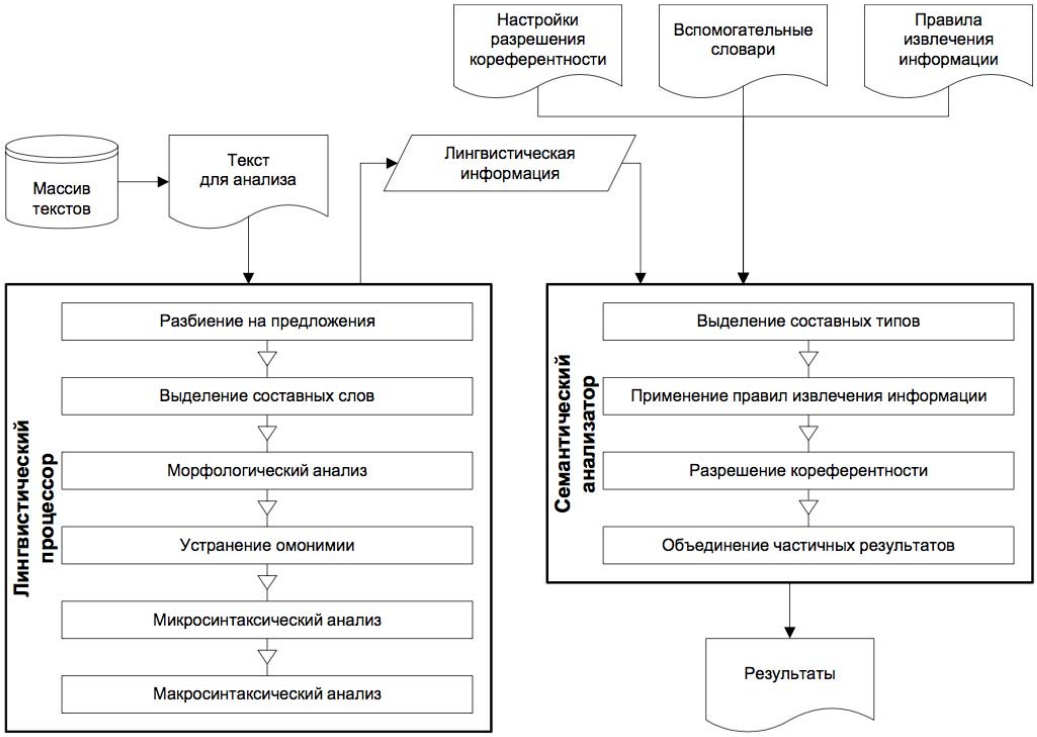

Рис. 1. Обобщённая архитектурно-структурная организация систем извлечения информации из текстов

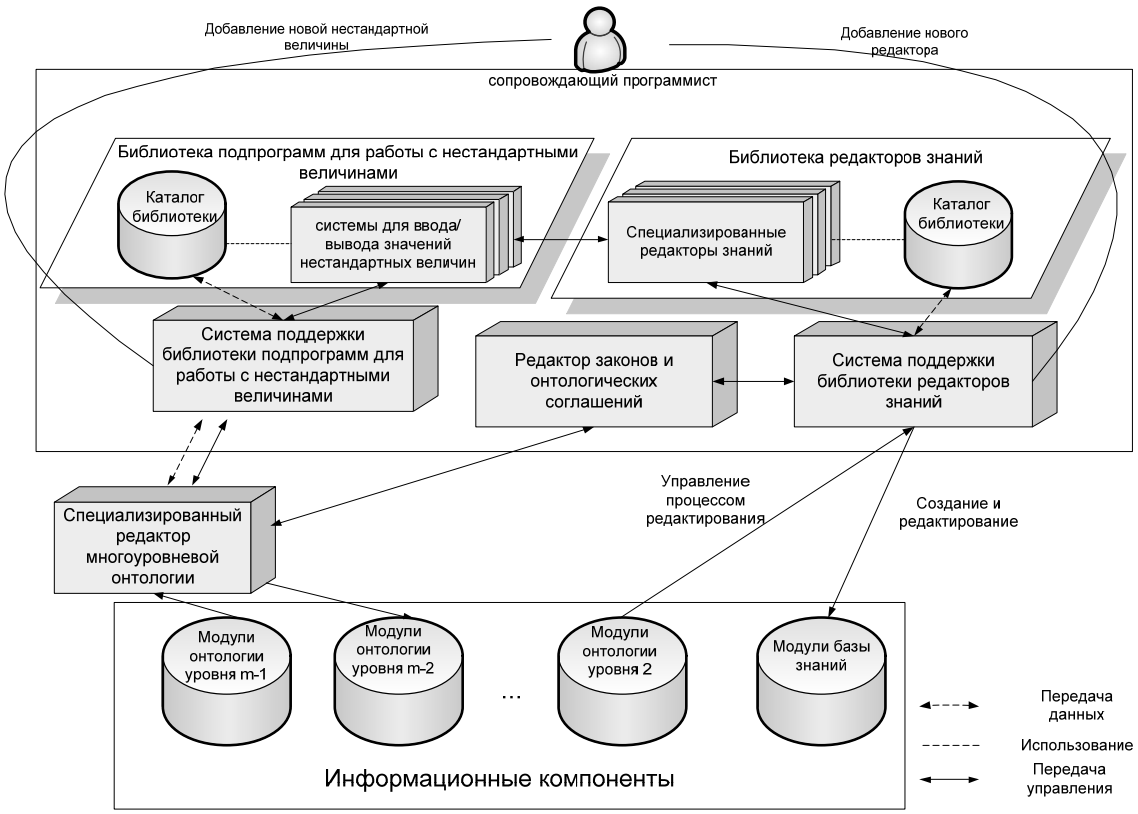

Рис. 2. Состав редакторов информационных компонент

В сложно-структурированных предметных областях, связанных с наукой, могут изменяться не только знания, но онтологии, и, как следствие, множество классов решаемых задач. Однако методы создания специализированных оболочек не учитывают данных свойств сложно-структурированных предметных областей. Целью данной работы является описание особенностей специализированных оболочек интеллектуальных систем для сложно-структурированных предметных областей.

Информационными компонентами специализированной оболочки для сложно-структурированной ПО [11] являются многоуровневая модульная онтология и модульная база знаний. Создание и редактирование информационных компонент осуществляется многоуровневым редактором онтологий и редактором знаний, разработка которых основывается на онтологии уровня n.

Редакторы многоуровневых онтологии и знаний должны позволять создание и редактирование модульных онтологий и знаний, а также обеспечивать возможность повторного использования модулей при создании онтологий и знаний новых разделов и подразделов области, причем процесс создания и редактирования модуля онтологии уровня i-1 должен управляться онтологией уровня i, а процесс создания и редактирования модуля знаний – онтологией уровня 2.

Редактор онтологии должен обеспечивать возможность выбора того из существующих модулей онтологии уровня i, который управляет процессом редактирования создаваемого модуля. Аналогично при редактировании модуля знаний должна обеспечиваться возможность выбора «управляющего» модуля онтологии уровня 2.

Редакторы онтологии и знаний должны обеспечивать возможность задания структурированной и неструктурированной части онтологии, а также структурированной и неструктурированной части знаний, т.е. программным компонентом этих редакторов должен быть специализированный редактор утверждений, позволяющий вводить онтологические соглашения и законы предметной области.

Редактор знаний должен обеспечивать возможность ввода/вывода значений нестандартных величин при редактировании знаний. Для значений нестандартных величин в предметной области может существовать способ их графического представления. Например, для химии [5] графически может быть задана краткая структурная формула или структурная формула химического соединения. Поэтому редактор знаний должен обеспечивать возможность использования принятого в предметной области графического способа представления значений нестандартных величин при создании и редактировании знаний. Величина, которой принадлежит значение некоторого свойства, задается онтологией уровня 2. Поэтому редактор знаний должен обеспечивать автоматический выбор (управляемый онтологией уровня 2) средств для графического представления значений нестандартных величин при редактировании знаний.

Редактор онтологии интерпретирует онтологию уровня i при создании модуля онтологии уровня i-1. Редактор знаний интерпретирует онтологию уровня 2 при создании модуля знаний. Одна и та же онтология может интерпретироваться разными способами в разных редакторах знаний. Редакторы знаний могут отличаться не только способом интерпретации знаний, но и интерфейсом. Очевидно, что более удобный интерфейс и более понятный эксперту способ интерпретации можно обеспечить для редактора, предназначенного для интерпретации одной онтологии, а не класса онтологий. Поэтому специализированная оболочка должна позволять использование редакторов, поддерживающих разные способы интерпретации модуля онтологии уровня 2 и предоставлять возможность эксперту выбора требуемого ему редактора знаний.

Значения нестандартных величин используются не только при редактировании знаний, но также при вводе исходных данных задач. Графический способ задания исходных данных задач более удобен для специалиста предметной области, поскольку в этом случае отсутствует необходимость громоздкого вербального описания этих данных [5]. Графическое представление результатов решения является более наглядным способом представления. Поэтому оболочка должна обеспечивать возможность ввода/вывода значений нестандартных величин при задании исходных данных задач, а также позволять использование принятого в предметной области графического способа представления значений нестандартных величин при вводе исходных данных задач и выводе результатов их решения.

Как уже отмечалось, величина, которой принадлежит значение некоторого свойства, задается онтологией уровня 2. Оболочка должна обеспечивать автоматический выбор (управляемый онтологией) средств для графического представления значений нестандартных величин при задании исходных данных задач.

Каждый раздел сложно-структурированной ПО характеризуется своим множеством классов прикладных задач, причем разные множества могут содержать как общие классы задач, так и специфичные для раздела. Решатель задач может быть предназначен для решения классов задач одного раздела (в этом случае он использует онтологию и знания этого раздела), либо разных разделов (в этом случае он может использовать разные онтологии и знания). В первом случае используемая решателем онтология определяется

классом задач. Во втором случае требуется дополнительное указание, какие онтология и знания должны использоваться в процессе решения. Специализированная оболочка интеллектуальных систем для сложно-структурированной предметной области должна обеспечивать возможность решения задач разных классов, причем пользователь должен иметь возможность указания модуля онтологии и модуля знаний, которые надо использовать при решении задач.

Таким образом, специализированная оболочка должна содержать расширяемые библиотеки систем для решения задач разных классов, системы автоматического построения методов решения задач по их спецификации (рис. 3). Метод решения задач может быть представлен либо в виде алгоритма, либо в виде множества правил системы продукций. В первом случае для создания решателя задач используется процессор алгоритмического языка, во втором случае – процессор языка, основанного на правилах, который является одним из программных компонент специализированной оболочки.

Специализированная система обработки текстовых документов "ЛоТА" (Невзорова О.А.)

Специализированная система обработки текстовых документов "ЛоТА" [8] является системой класса Text Mining. Система предназначена для анализа специализированных текстов "Логика работы", описывающих логику работы сложной технической системы в различных режимах функционирования. Основной задачей анализа является извлечение из данных текстов информационной модели алгоритмов, решающих определенную задачу в определенной проблемной ситуации, и контроль структурной и информационной целостности выделенной схемы алгоритмов.

Информационная модель алгоритма включает:

- описание входного информационного потока (типы информационных сигналов или семантическое описание информационного потока с указанием источника информации - конкретный алгоритм, конкретное измерительное устройство);
- описание процессов преобразования входных данных в выходные (допустимый способ разрешения проблемы);
- описание выходного информационного потока (типы информационных сигналов или семантическое описание информационного потока с указанием точки приема информации).

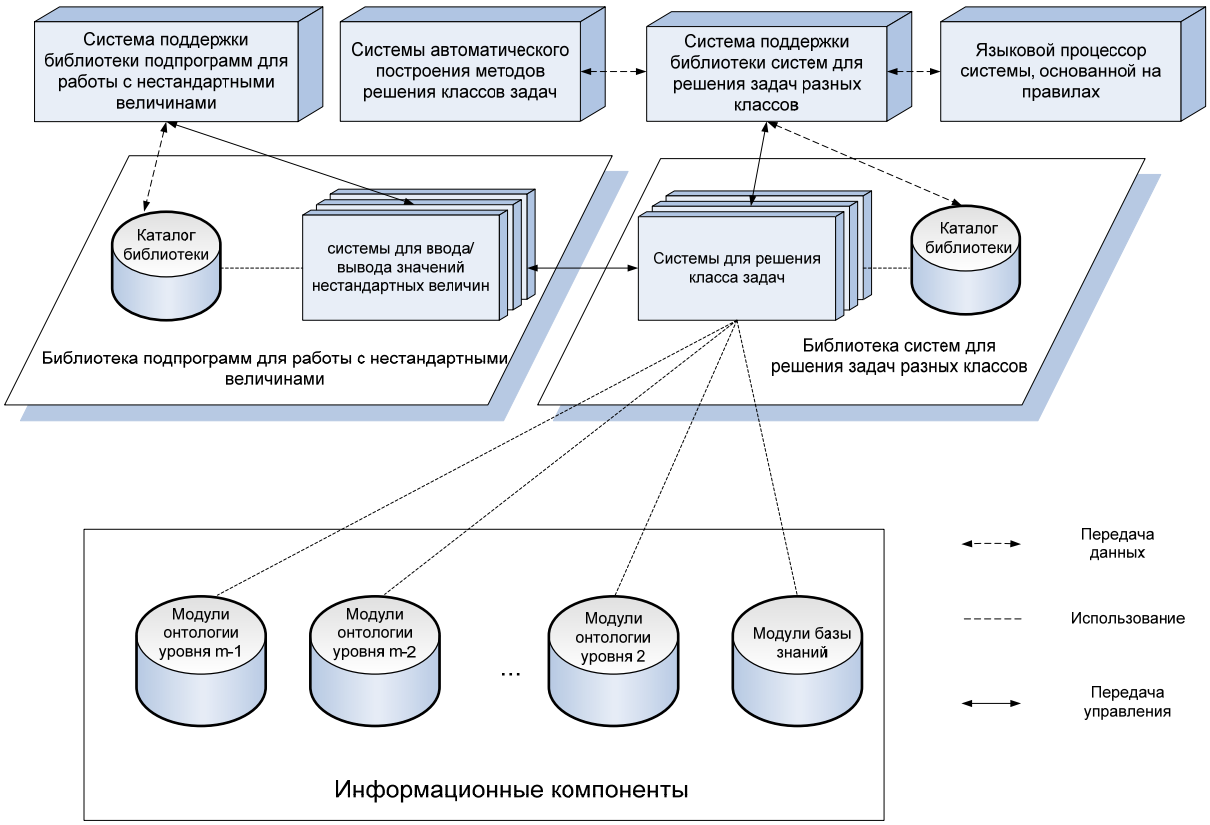

Рис. 3. Системы для решения задач и средства их разработки

Решение основной задачи обеспечивается комплексом технологий обработки текстов, включающих:

- технологии морфосинтаксического анализа;
- технологии семантико-синтаксического анализа;
- технологии взаимодействия с прикладной онтологией.

Указанная сумма технологий формируется на основе центрального ядра – прикладной онтологии (в дальнейшем, авиаонтология), обеспечивающей согласованное взаимодействие различных программных модулей. Авиаонтология концептуально описывает предметную область информационного обеспечения различных полетных режимов антропоцентрических систем [2]. Авиаонтология представляет собой сеть понятий предметной области. Текущий размер онтологии - свыше 1600 понятий (около 5000 текстовых входов понятий). Авиаонтология относится к классу лингвистических (лексических) онтологий и предназначена для встраивания в различные лингвистические приложения.

Программный комплекс состоит из трех взаимодействующих подсистем: подсистемы лингвистического анализа технических текстов "Анализатор", подсистемы ведения онтологии "OntoEditor+" и подсистемы "Интегратор". Взаимодействие подсистем реализовано на базе технологии "клиент-сервер", причем в различных подзадачах подсистемы выступают в различных режимах (режим сервера или режим клиента).

Инструментальная система визуального проектирования "OntoEditor+" [5] является специализированной СУБД. Система предназначена для ручного редактирования онтологий, хранящихся в реляционной базе данных в формате TPS, а также обслуживания запросов пользователей и внешних программ. Новые возможности системы обеспечиваются функциональным набором "Лингвистический инструментарий", посредством которого реализуется встраивание прикладной онтологии в лингвистические приложения. Наиболее типичными задачами, решаемыми с помощью инструментария системы "OntoEditor+", являются: изучение структурных свойств прикладной онтологии с помощью исследовательского инструментария системы "OntoEditor+"; построение лингвистической оболочки прикладной онтологии; задача покрытия текста онтологическими входами; построение выводов по прикладной онтологии и др.

Подсистема "Анализатор" реализует основные этапы лингвистической обработки текста (графематический, морфосинтаксический и частичный синтаксический анализ). В статье будет рассмотрена интегральная технология разрешения многозначности, которая ориентирована прежде всего на разрешение функциональной, морфологической и лексической омонимии.

Подсистема "Интегратор" исполняет внешний запрос на извлечение знаний из текста. Структура внешнего запроса содержит компоненты информационной модели алгоритма. Внешний запрос интерпретируется при взаимодействии с подсистемой "OntoEditor+" как структура, привязанная к прикладной онтологии. Выделение компонент информационной модели происходит на основе механизмов отождествления элементов дерева сегментов входного текста (взаимодействие с подсистемой "Анализатор") и элементов структуры запроса (взаимодействие с подсистемой "OntoEditor+").

Инструментальная система визуального проектирования онтологий "OntoEditor+" включает:

- лингвистический инструментарий (задачи корпусного исследования (загрузка корпуса; сегментация на предложения; автоматическое ведение статистики по различным объектам корпуса), построение лингвистической оболочки онтологии, задача покрытия текста онтологическими входами, построение выводов по онтологии, поддержка протоколов информационного обмена системы "OntoEditor+" с внешними программными модулями, в том числе с внешними информационными ресурсами);
- исследовательский инструментарий;

Основные функции подсистемы “Анализатор”:

- графематический анализ
- морфосинтаксический анализ
- покрытие текста онтологическими входами (взаимодействие с системой «OntoEditor+»)

Основные функции подсистемы “Интегратор”:

- анализ и исполнение внешнего запроса (информационная модель алгоритма)
- интерпретация внешнего запроса в терминах прикладной онтологии (взаимодействие с системой «OntoEditor+»)
- интерпретация внешнего запроса в структурных компонентах дерева сегментов (взаимодействие с системой «Анализатор»)
- контроль информационной целостности (анализ компонент внешнего запроса).

Метод контекстного разрешения омонимии является базовым методом в интегральной технологии разрешения омонимии в системе "ЛоТА". Однако, практические задачи системы выявили ряд важных аспектов лингвистического анализа, которые стимулировали развитие новых методов разрешения многозначности.

Интегральная технология разрешения многозначности, разрабатываемая в системе "ЛоТА" включает следующие методы:

- метод контекстного разрешения функциональной омонимии;
- метод разрешения функциональной, грамматической и лексической омонимии на основе индексируемой базы устойчивых коллокаций;

- метод разрешения функциональной, грамматической и лексической омонимии на основе лингвистической оболочки онтологии.

Для эффективного встраивания в лингвистические приложения система "OntoEditor+" поддерживает группу протоколов информационного обмена с внешними программными модулями системы и внешними словарными базами данных, обеспечивая работу в режиме клиент-сервер. Разрешение многозначности (функциональной, морфологической и лексической) во входных текстах происходит на основе механизма распознавания контекстов омонимов, зафиксированных в индексируемой базе контекстов.

Разработаны три основных механизма пополнения индексируемой базы контекстов функциональных омонимов:

- ручной ввод и редактирование данных по типовым контекстам омонимов;
- импорт типовых контекстов омонимов из текстового файла, подготовленного в специальном формате представления данных;
- импорт типовых контекстов омонимов, обнаруженных специальными механизмами поиска подсистемы "Анализатор".

Данный механизм организован как запрос к подсистеме "Анализатор" с передачей ему от подсистемы "OntoEditor+" текстового корпуса, по которому проводится поиск. В процессе обработки подсистема "Анализатор" передает подсистеме "OntoEditor+" информацию об обнаруженных контекстах омонимов, которая записывается либо в индекс омонимов либо в автоматическом режиме, либо в режиме диалога с оператором. Отличительной особенностью режима диалога является режим самообучения, который реализуется с использованием механизма журнала событий. В данном журнале в зависимости от его настройки фиксируются те или иные важные события в системе, например, изменение информации в индексе омонимов или операции взаимодействия с подсистемой "Анализатор". В режиме самообучения сохраняется и контролируется последовательность ранее сгенерированных диалогов, что обеспечивает генерацию только уникальных диалогов на разрешение омонимии без повторений.

Лингвистический инструментарий подсистемы "OntoEditor+" обеспечивает встраивание онтологии в различные приложения, связанные с обработкой текстов. Лингвистический инструментарий реализует функции загрузки корпуса текстов; автоматическое ведение статистики по различным объектам корпуса; функции предсинтаксической обработки текста (сегментация предложений, распознавание аббревиатур, разрешение омонимии на основе специальных протоколов взаимодействия с внешними словарными ресурсами); построение лингвистической оболочки онтологии; распознавание терминов прикладной онтологии во входном тексте (задача покрытия). Сопряжение онтологического и лингвистического (грамматического) ресурсов реализуется через механизмы лингвистической оболочки онтологии. Лингвистическая оболочка онтологии создается с помощью разработанного программного инструментария, посредством которого фиксируется грамматическая информация об онтологических концептах и их текстовых формах. Каждый онтологический вход (как правило, многословный термин) снабжается соответствующей грамматической информацией, при этом для омонима разрешается соответствующая (функциональная, лексическая, морфологическая) омонимия. Грамматическая информация передается в подсистему "OntoEditor+" от подсистемы "Анализатор" на основе специальных протоколов взаимодействия. Разрешение лексической, функциональной и морфологической омонимии выполняется на основе специальных диалогов с экспертом-лингвистом. Отдельные процедуры реализуют проверки словоформ в составе терминологического входа на согласованность их грамматических характеристик, также осуществляется контроль достоверности словарной информации. Контроль достоверности обеспечивает отслеживание изменений, как в составе грамматического словаря, так и в составе онтологии. Учитывая сложность и многоступенчатость вышеперечисленных процедур, в подсистеме "OntoEditor+" разработан мастер построения лингвистической оболочки, который вызывается командой основного меню.

Подсистема "Анализатор" обеспечивает реализацию метода разрешения омонимии на основе контекстных правил, т.е. фактически используются лингвистические знания системы. Этот метод является универсальным, не зависит от специфики предметной области и обеспечивает в текущей версии точность распознавания не ниже 95%. Однако, для данного метода существуют крайне сложные типы функциональной омонимии, например, тип "частица/союз". Разрешение данной омонимии возможно во многих случаях лишь после завершения полного синтаксического анализа.

Взаимодействие подсистемы "OntoEditor+" и подсистемы "Анализатор" осуществляется на основе специальных протоколов взаимодействия. При применении интегральной технологии разрешение многозначности происходит в два этапа. На первом этапе подсистема "Анализатор" (клиент) передает запрос на разрешение омонимии входного текста подсистеме "OntoEditor+" (сервер). Подсистема "OntoEditor+" возвращает подсистеме "Анализатор" информацию о разрешенных омонимах на основе своих методов. На втором этапе подсистема "Анализатор" разрешает омонимию оставшихся неразрешенных омонимов на основе метода контекстных правил.

Интегральная технология разрешения многозначности эффективно применяется на этапе предсинтаксического анализа в системе "ЛоТА". По существу, интегральная технология представляет собой сочетание инженерного и лингвистического подхода к решению поставленной задачи. В основе проектирования интегральной технологии лежат процессы скоординированного взаимодействия различных языковых уровней, прежде всего онтологического уровня (обеспечивающего системные модели знаний о мире) и различных языковых уровней (морфологического и синтаксического). В системе реализован эффективный механизм взаимодействия различных подсистем, обеспечивающих реализацию различных методов в составе интегральной технологии.

Интеллектуальная система извлечения данных и их Анализа(на основе текстов) ИСИДА-Т.

Целью ИСИДА-Т [10, 11], является извлечение значимой информации определенного типа из (больших массивов) текста для дальнейшей аналитической обработки. Результатом работы систем является получение структурированных данных и отношений на них.

Основные компоненты ИСИДА-Т:

- Инфраструктурные службы (конфигурирование, параллельная обработка, взаимодействие модулей);
- Лингвистический процессор;
- Модули работы со знаниями ПдО;
- Интерпретатор правил извлечения информации;

Разработанные в рамках проекта Исида-Т технологии, инструменты и продукты позволяют:

- обнаруживать в электронных документах, извлекать и структурировать информацию о представляющих интерес фактах, событиях, объектах и отношениях;
- выполнять мониторинг сайтов в сети Интернет на предмет появления там значимой для пользователя информации.

Основные рабочие характеристики технологии и продуктов:

- поддержка русского языка;
- быстрая настройка на предметную область при помощи эффективных инструментальных средств;
- высокая точность и полнота анализа за счет использования предметных знаний;
- наличие встроенных средств визуализации результатов анализа в виде диаграмм и схем;
- легкая интегрируемость в другие информационные системы на любом уровне (программный или сетевой интерфейс, БД);
- функционирование под управлением ОС Windows и большинства Linux-систем;
- близкая к линейной масштабируемость при параллельной архитектуре анализа. Возможность работы на вычислительных машинах кластерного типа.

Некоторые области применения технологий семантического анализа и структурирования текстовой информации:

- информационная поддержка бизнеса (business intelligence) и управление знаниями (knowledge management);
- маркетинговые исследования;
- финансовая аналитика;
- военная и коммерческая разведка и мониторинг;
- информационная поддержка органов государственной власти (в рамках направления "Электронное правительство");
- работа библиотек, издательств и СМИ.

Рассмотрим общую организацию инфраструктуру системы ИСИДА-Т. Краеугольным камнем системы ИСИДА-Т является точная настройка на предметную область и конкретную задачу извлечения. С одной стороны, это достигается за счет редактирования лингвистических ресурсов, ресурсов знаний, правил извлечения и правил трансформации. С другой стороны, настройка может потребовать включения в процесс обработки дополнительных специализированных методов обработки текста. Кроме того, для каждой задачи необходимо подобрать наиболее подходящие алгоритмические средства анализа из набора имеющихся. Эти аспекты требуют создания такой архитектуры, при которой легко могут добавляться и замещаться алгоритмические компоненты процесса извлечения.

Проблема конфигурирования на алгоритмическом уровне потребовала создания модульной архитектуры и декларативного подхода к определению процесса извлечения. Модули получили название обрабатывающих ресурсов в противовес лингвистическим ресурсам и ресурсам знаний. В конфигурации декларируется порядок обработки документа аналитическими модулями, потоки данных между ними, а также параметры их работы.

Обрабатывающие ресурсы можно разделить на следующие группы.

– Ресурсы предобработки. Сюда относятся средства определения кодировки документа, извлечения текста и стилевой разметки из документа, предварительной фильтрации.

– Ресурсы лингвистического анализа. Осуществляют разбор текста на отдельные слова, морфологический анализ (в том числе специализированные варианты для различных категорий имен собственных), поверхностный синтаксический анализ и определение границ предложений.

– Ресурсы извлечения. Осуществляют поиск в документе целевой лексики и синтаксических конструкций, а также первичное структурирование информации.

– Ресурсы унификации знаний и вывода. Осуществляют унификацию и отождествление элементов знаний, вывод производных знаний.

– Ресурсы подготовки результата. Осуществляют приведение извлеченной информации к определенному формату и передачу за пределы последовательности обработки (в БД, глобальный ресурс знаний, файл, приложение).

В системе ИСИДА-Т все модули, в том числе средства общего лингвистического анализа, используют структуру данных – аннотация. Аннотация — объект, который приписывается фрагменту текста (например, слову, словосочетанию, предложению, ссылке на сущность предметной области и т.д.) и описывает свойства этого фрагмента. Аннотации разбиты на конечное множество классов. Каждый класс аннотаций описывает текст в определенном аспекте. Информация о фрагменте представлена значениями именованных атрибутов аннотации. Наборы классов и атрибутов аннотаций намеренно не специфицированы, чтобы можно было использовать произвольный набор обрабатывающих модулей и представлять необходимую лингвистическую и предметную информацию. Обмен данными между модулями тоже идет в терминах аннотаций: новые аннотации могут строиться на основании полученных на предыдущих этапах анализа [5]. В реализации системы ИСИДА-Т модель аннотаций была дополнена некоторыми полезными средствами. В частности, было снято ограничение на атомарность атрибутов и добавлена возможность устанавливать ссылки между аннотациями.

Для распознавания текстовых ситуаций используется набор правил, описывающих характерные для конкретной задачи способы выражения ситуации в тексте. Эти правила задают образец для сопоставления и действия, которые должны быть произведены после успешного сопоставления. Ряд современных систем извлечения информации (в том числе, система ИСИДА-Т) берут за основу различные диалекты языка CPSL [6]. Использование этого языка подразумевает разметку текста при помощи аннотаций.

Язык правил, используемый в системе ИСИДА-Т, является расширением CPSL. Предлагаемые расширения преследуют две цели: 1) обеспечить возможность описывать более сложные контексты, в которых встречается целевая информация, и 2) снизить объем рутинной работы при создании системы правил за счет более компактного описания контекста [2].

Отличия от других реализаций, например, JAPE [7] или диалекта CPSL состоят в следующем.

– Реализована встроенная поддержка расширенного спектра типов данных, в том числе, ссылок на аннотации и множественных значений. Данные этих типов могут использоваться в качестве значений переменных и значений атрибутов аннотаций.

– Логика работы интерпретатора правил приведена в максимальное соответствие поведению интерпретатора обычных регулярных выражений. Отличия от современной реализации JAPE и Montreal transducer [8] заключаются в поддержке «жадных» и «нежадных» квантификаторов и опережающей проверки.

– Поддерживаются кванторы существования (по умолчанию) и всеобщности, связывающие элементарные тесты. К кванторам может добавляться отрицание.

– Существуют языковые средства, позволяющие гибко проверять взаимное расположение аннотаций, рассматриваемых в контексте сопоставления, и прочих аннотаций во входной коллекции.

– В тестах могут использоваться функции для обращения к ресурсу знаний, например, проверки таксономической принадлежности элементов. Для более сложных запросов к ресурсу знаний используется предметно-ориентированный язык, совпадающий с языком описания левой части правил трансформации.

– Для передачи информации между элементарными тестами, а также в правую часть правил могут использоваться именованные переменные, значения которых присваиваются явно в ходе сопоставления. Множество значений переменных входит в контекст сопоставления.

– Инструментальное средство проектирования онтологий Protégé

Protégé [12] – локальная, свободно распространяемая Java-программа, разработанная группой медицинской информатики Стэндфордского университета. Программа предназначена для построения (создания, редактирования и просмотра) онтологий прикладной области. Её первоначальная цель – помочь разработчикам программного обеспечения в создании и поддержке явных моделей предметной области и включение этих моделей непосредственно в программный код. Protégé включает редактор онтологий, позволяющий проектировать онтологии разворачивая иерархическую структуру абстрактных или конкретных классов и слотов. Структура онтологии сделана аналогично иерархической структуре каталога. На основе сформированной онтологии, Protégé может генерировать формы получения знаний для введения экземпляров классов и подклассов. Инструмент имеет графический интерфейс, удобный для использования неопытными пользователями, снабжен справками и примерами.

Protégé основан на фреймовой модели представления знания OKBC (Open Knowledge Base Connectivity) [12] и снабжен рядом плагинов, что позволяет его адаптировать для редактирования

моделей хранимых в разных форматах (стандартный текстовый, в базе данных JDBC, UML, языков XML, XOL, SHOE, RDF и RDFS, DAML+OIL, OWL).

Используемые формализмы и форматы

Изначально единственной моделью знаний, поддерживаемой Protégé, была фреймовая модель. Этот формализм сейчас является "родным" для редактора, но не единственным.

Protégé имеет открытую, легко расширяемую архитектуру и помимо фреймов поддерживает все наиболее распространенные языки представления знаний (SHOE, XOL, DAML+OIL, RDF/RDFS, OWL). Protégé поддерживает модули расширения функциональности (plug-in). Расширять Protégé для использования нового языка проще, чем создавать редактор этого языка "с нуля".

Protégé основан на модели представления знаний OKBC (Open Knowledge Base Connectivity). Основными элементами являются классы, экземпляры, слоты (представляющие свойства классов и экземпляров) и фасеты (задающие дополнительную информацию о слотах).

Пользовательский интерфейс

Пользовательский интерфейс состоит из главного меню и нескольких вкладок для редактирования различных частей базы знаний и ее структуры. Набор и названия вкладок зависят от типа проекта (языка представления) и могут быть настроены вручную. Обычно имеются следующие основные вкладки: Классы, Слоты (или Свойства для OWL), Экземпляры, Метаданные.

Инструментальный комплекс автоматизированного построения онтологий ПдО

Инструментальный комплекс онтологического назначения для автоматизированного построения онтологии в произвольной предметной области [1] является системой, реализующей одно из направлений комплексных технологий Data & Text Mining, а именно – анализ и обработку больших объёмов неструктурированных данных, в частности лингвистических корпусов текстов на украинском и/или русском языке, извлечение из них предметных знаний с последующим их представлением в виде системно-онтологической структуры или онтологии предметной области. ИКОН предназначен для реализации множества компонентов единой информационной технологии:

- поиск в сети Internet и/или в других электронных коллекциях (ЭлК) текстовых документов (ТД), релевантных заданной ПдО, их индексацию и сохранение в базе данных;
- автоматическая обработка естественно-языковых текстов (Natural Language Processing);
- извлечение из множества ТД знаний, релевантных заданной ПдО, их системно-онтологическая структуризация и формально-логическое представление на одном (или нескольких) из общепринятых языков описания онтологий (Knowledge Representation). Кроме того, внутри этой технологии реализуется процедура построения, визуализации и проверки семантических структур синтаксических единиц ТД и понятийных структур заданной ПдО в виде несильно связанного онтографа, названного начальной онтологией ПдО (НО ПдО);
- создание, накопление и использование больших структур онтологических знаний в соответствующих библиотеках;
- системная интеграция онтологических знаний как одна из основных компонент методологии междисциплинарных научных исследований;
- другие процедуры, связанные с автоматизацией приобретения знаний из множества естественно-языковых объектов.

ИКОН состоит из трёх подсистем и представляет собой интеграцию разного рода информационных ресурсов (ИР), программно-аппаратных средств обработки и процедур естественного интеллекта (ЕИ), которые, взаимодействуя между собой, реализуют совокупность алгоритмов автоматизированного, итерационного построения понятийных структур предметных знаний, их накопления и/или системной интеграции. Обобщённая блок-схема ИКОН представлена на рис. 4.

Подсистема Информационный ресурс включает блоки формирования лингвистического корпуса текстов, баз данных языковых структур и библиотек понятийных структур. Первый компонент представляет собой различные источники текстовой информации, поступающей на обработку в систему. Второй компонент представляет собой различные базы данных обработки языковых структур, часть из которых формируется (наполняется данными) в процессе обработки ТД, а другая часть формируется до процесса построения О ПдО и, по сути, является ЭлК различных словарей. Третий компонент представляет собой совокупность библиотек понятийных структур разного уровня представления (от наборов терминов и понятий до высокоинтегрированной онтологической структуры междисциплинарных знаний) и является результатом реализации некоторого проекта (проектирования онтологии ПдО и/или системной интеграции онтологий).

Подсистема Программно-аппаратные средства включает блоки обработки языковых и понятийных структур и управляющую графическую оболочку (УГО). Последняя, во взаимодействии с инженером по знаниям, осуществляет общее управление процессом реализации связанных информационных технологий.

Подсистема Естественный интеллект осуществляет подготовку и реализацию процедур предварительного этапа проектирования, а на протяжении всего процесса осуществляет контроль и проверку результатов выполнения этапов проектирования, принимает решение о степени их

завершённости (и в случае необходимости – повторении некоторых из них).

Результаты системного анализа приведены в таблице.

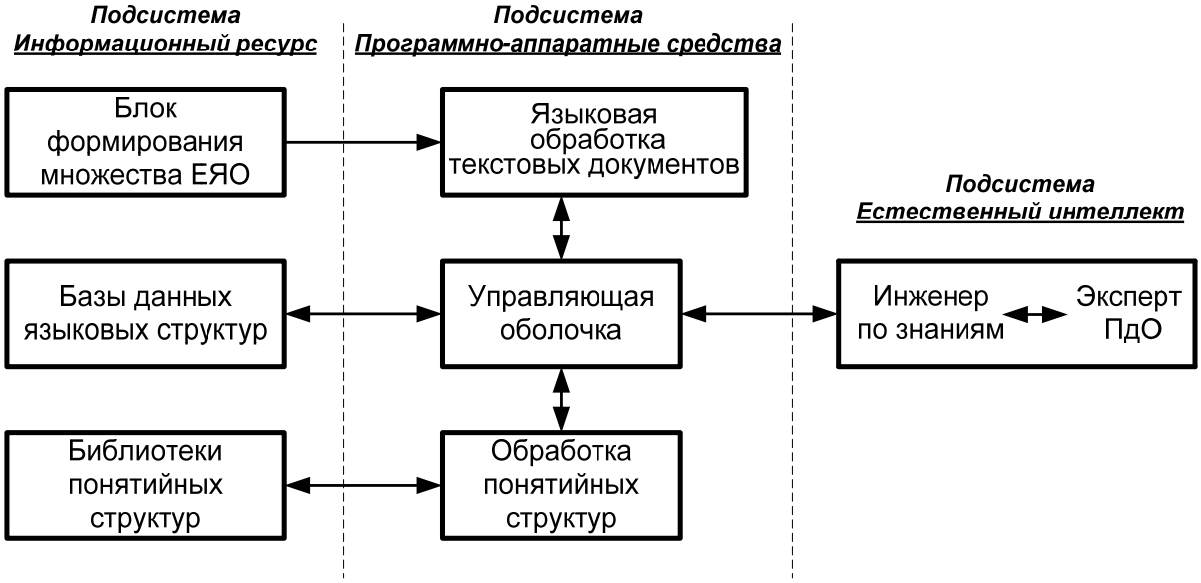

Рис. 4. Обобщённая блок-схема ИКОН

Таблица

Результаты системного анализа инструментальных средств инженерии онтологий

	ИКОН	ЛоТА	SIMER+MIR	ИСИДА-Т	Protégé	Ontolingua	InTez	OntoSTUDIO	OntoEdit
1	2	3	4	5	6	7	8	9	10
Общая информация									
Разработчик	ИК НАНУ	?	РАН ИПС ИЦИИ	РАН ИПС ИЦИИ	SMI, Stanford University	KSL, Stanford University	С-ПТ ГУ	Semafora systems GmbH	Ontoprizse
Версия	1.0	?	?	?	4.1	4.0	?	3.1	
URL	http://aduis.com.ua/	–	–	http://isida-t.ru/	http://protege.stanford.edu/	www.ksl.stanford.edu/software/ontolingua/	http://intte z.ru	www.semafora-systems.com/en/products/ontostudio	www.semafora-systems.com/products/ontoedit
Доступность	Закрытый код, ограниченный доступ	Закрыт ый доступ	Закрытый доступ	Закрытый доступ	Закрытый код, свободный доступ	Свободны й доступ	Закрытый код, свободны й доступ	Закрытый код, ограниченный доступ	Свободна я лицензия
Интерфейс пользователя	Локально е приложен ие	?	Локально е приложен ие	Локальное приложение	Локальное приложение, Веб-приложение	Веб-приложен ие	Локально е приложен ие	Локально е приложен ие, Веб-прилож.	Локально е приложен ие
Язык интерфейса ПО	Русский, Украинск ий	Русский	Русский	Русский, Английский	Английский	Английск ий	Русский	Английск ий	Английск ий
Архитектура программного обеспечения									
Расширяемость	Плагины	–	–	–	Плагины	Плагины	–	Плагины	Плагины
Язык ПО (платформа)	Java, Delphi	C++	?	?	Java	Lisp	?	Построен на платформ е Eclipse	Java
Хранение онтологий	Файлы, СУБД	СУБД (формат TPS)	Файлы	Файлы	Файлы, СУБД	Файлы	Файлы, СУБД	Файлы, СУБД	Файлы
Архитектура приложения	n-уровневая	Клиент-сервер		3-х уровневая	3-х уровневая	Клиент-сервер		3-х уровневая	3–х уровневая
Функционал инструментального средства									
Язык обрабатываемых ЛКТ	Украинск ий, Русский	Русский	Русский	Русский	Английский	Английск ий	Русский	–	–
Управляющая граф. оболочка системы	+	+	+	+	+	+	+	+	+

Продолжение табл.

1	2	3	4	5	6	7	8	9	10
Систем. интеграция онтологий	+	–		–	+	–	–	+	+
Система поиска ТД (Поиск ТД во внешних и внутренних информационны х ресурсах)	+	–		–	–	+			–
Визуальное проектирование, редактирование онтологических структур	+	+		+	+	+		+	+
Автоматиз. построение онтологии ПдО	+–	–		–	-	–		–	–
Ручное построение онтологии ПдО	+	–		+	+	+		+	+
Автомат. лингвистический анализ ТД, описывающих заданную ПдО (синтактико-семантический, морфо-синтакс. анализ)	+	+		+	-	–		–	–
Формально-логическое представления и интеграция онтологических структур в онтологическую базу знаний ПдО	+	+		+	+	+		+	+
Хранение и управление электронными коллекциями энциклопедическ их и толковых словарей, тезаурусов	+	+		+	-	+		–	–
Лингвистический корпус текстов (ЛКТ) (База данных ТД)	+	+		+	-	–		+	–
Эксперт ПдО	+	+	+	+	+	+	+	+	+
Инженер по знаниям	+	+	+	+	+	+	+	+	+
Интероперабельность									
Конвертация файлов описания онтологий	+	–	–	–	+	KIF, Prolog, Ontolingua LOOM, CLIPS, IDL	Экспорт в OWL	Импорт из UML 2.0, Database schemas (Oracle, MySQL), Excel tables,	Импорт из OXML, RDF(S) DAML+O IL, FLogic Экспорт в OXML, RDF(S)
Интеграция с другими инструментальн ыми средствами	Protégé	?	?	?	Возможна по средствам плагинов	OKBC, Chimaera		Protégé	OntoAnnot ate, OntoMat, Semantic-Miner
Интеграция с базой знаний WolframAlpha	+	–	–	–	–	–	–	–	–
Интеграция с Google, Bing		–	–	–	–	–	–	–	–
Параллельная обработка данных	+	–	–	+	–	–	–	–	–

Литература

1. Палагин А. В. Онтологические методы и средства обработки предметных знаний / А.В. Палагин, С. Л. Крывый, Н.Г. Петренко. – [монография] – Луганск: изд-во ВНУ им. В. Даля, 2012. – 323 с.
2. Гаврилова Т. А. Базы знаний интеллектуальных систем / Гаврилова Т. А., Хорошевский В. Ф. // Учебник для вузов. – СПб, Изд-во “Питер”, 2000. – 384 с.
3. Палагин А. В. Системно-онтологический анализ предметной области / Палагин А. В., Петренко Н. Г. // УСиМ. – 2009. – № 4. – С. 3 – 14.
4. Палагин А. В. К анализу естественно-языковых объектов / Палагин А. В., Крывый С. Л., Величко В. Ю., Петренко Н. Г. // "Intelligent Processing" International Book Series “INFORMATION SCIENCE & COMPUTING”, Number 9, Supplement to the International Journal “INFORMATION TECHNOLOGIES & KNOWLEDGE”. – V. 3. – 2009. – pp. 36 – 43.
5. Овдей О. М. Обзор инструментов инженерии онтологий / О. М. Овдей, Г. Ю. Проскудина. – Российский научный электронный журнал «Электронные библиотеки» [Электронный ресурс]. – 2004. – т. 7. – Вып. 4. – Режим доступа : http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4/op.
6. Noy N. SMART: Automated Support for Ontology Merging and Alignment / N. Noy, M. Musen. – Stanford Medical Informatics, Stanford Univ [Электронный ресурс]. – 1999. – 24 p. – Режим доступа: http://ais-portal.ru/2009/03.
7. Филатов В. А. Разработка высокоэффективных средств создания и обработки онтологических баз знаний / Филатов В. А., Щербак С. С., Хайрова А. А. // Системи обробки інформації [Электронный ресурс]. – Вип. 8 (66), 2007. – С. 120 – 124. – Режим доступа: www.nbuv.gov.ua/portal/natural/soi/2007_8/Filatov.pdf.
8. Невзорова О. А. Система анализа технических текстов "ЛоТА": основные концепции и проектные решения / Невзорова О. А., Федунов Б. Е. // Изв. РАН. Теория и системы управления . – 2001. – № 3. – С. 138 – 149.
9. Артемьева И. Л. Интеллектуальная система, основанная на многоуровневой онтологии химии / Артемьева И. Л., Рештаненко Н. В. // Программные продукты и системы. – 2008. – № 1. – С. 84 – 87.
10. Кормалев Д. А. Развитие языка правил извлечения информации в системе ИСИДА-Т / Кормалев Д. А., Куршев Е. П. // Труды международной конференции «Программные системы: теория и приложения». – Т. 2. – М. : Физматлит. – 2006. – С. 365 – 377.
11. Киселев С. Л. Поиск фактов в тексте естественного языка на основе сетевых описаний / Киселев С. Л., Ермаков А. Е., Плешко В. В. // Компьютерная лингвистика и интеллектуальные технологии: труды Международной конференции Диалог’2004. – М. : Наука. – 2004.
12. OKBC: A Programmatic Foundation for Knowledge Base Interoperability / V. Chaudhri, A. Farquhar, R. Fikes P. Karp J. Rice // Fifteenth National Conf. on Artificial Intelligence. AAAIPres [Электронный ресурс]. – The MIT Press, Madison. – P. 600 – 607. – Режим доступа : 1998.http://www.oracle.com/technetwork/java/javase/index.html.

References

1. Palagin A. V. Ontologicheskie metody i sredstva obrabotki predmetnyh znanij / A.V. Palagin, S. L. Kryvyj, N.G. Petrenko. – [monografija] – Lugansk: izd-vo VNU im. V. Dalja, 2012. – 323 s.
2. Gavrilova T. A. Bazy znanij intellektual'nyh sistem / Gavrilova T. A., Horoshevskij V. F. // Uchebnik dlja vuzov. – SPb, Izd-vo “Piter”, 2000. – 384 s.
3. Palagin A. V. Sistemno-ontologicheskij analiz predmetnoj oblasti / Palagin A. V., Petrenko N. G. // USiM. – 2009. – № 4. – S. 3 – 14.
4. Palagin A. V. K analizu estestvenno-jazykovyh ob#ektov / Palagin A. V., Kryvyj S. L., Velichko V. Ju., Petrenko N. G. // "Intelligent Processing" International Book Series “INFORMATION SCIENCE & COMPUTING”, Number 9, Supplement to the International Journal “INFORMATION TECHNOLOGIES & KNOWLEDGE”. – V. 3. – 2009. – pp. 36 – 43.
5. Ovdej O. M. Obzor instrumentov inzhenerii ontologij / O. M. Ovdej, G. Ju. Proskudina. – Rossijskij nauchnyj jelektronnyj zhurnal «Jelektronnye biblioteki» [Jelektronnyj resurs]. – 2004. – t. 7. – Vyp. 4. – Rezhim dostupa : http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4/op.
6. Noy N. SMART: Automated Support for Ontology Merging and Alignment / N. Noy, M. Musen. – Stanford Medical Informatics, Stanford Univ [Jelektronnyj resurs]. – 1999. – 24 p. – Rezhim dostupa: http://ais-portal.ru/2009/03.
7. Filatov V. A. Razrabotka vysokojeffektivnyh sredstv sozdanija i obrabotki ontologicheskih baz znanij / Filatov V. A., Shherbak S. S., Hajrova A. A. // Sistemi obrobki informacii [Jelektronnyj resurs]. – Vip. 8 (66), 2007. – S. 120 – 124. – Rezhim dostupa: www.nbuv.gov.ua/portal/natural/soi/2007_8/Filatov.pdf.
8. Nevzorova O. A. Sistema analiza tehnicheskih tekstov "LoTA": osnovnye koncepcii i proektnye reshenija / Nevzorova O. A., Fedunov B. E. // Izv. RAN. Teorija i sistemy upravlenija . – 2001. – № 3. – S. 138 – 149.
9. Artem'eva I. L. Intellektual'naja sistema, osnovannaja na mnogourovnevoj ontologii himii / Artem'eva I. L., Reshtanenko N. V. // Programmnye produkty i sistemy. – 2008. – № 1. – S. 84 – 87.
10. Kormalev D. A. Razvitie jazyka pravil izvlechenija informacii v sisteme ISIDA-T / Kormalev D. A., Kurshev E. P. // Trudy mezhdunarodnoj konferencii «Programmnye sistemy: teorija i prilozhenija». – T. 2. – M. : Fizmatlit. – 2006. – S. 365 – 377.
11. Kiselev S. L. Poisk faktov v tekste estestvennogo jazyka na osnove setevyh opisanij / Kiselev S. L., Ermakov A. E., Pleshko V. V. // Komp'juternaja lingvistika i intellektual'nye tehnologii: trudy Mezhdunarodnoj konferencii Dialog’2004. – M. : Nauka. – 2004.
12. OKBC: A Programmatic Foundation for Knowledge Base Interoperability / V. Chaudhri, A. Farquhar, R. Fikes P. Karp J. Rice // Fifteenth National Conf. on Artificial Intelligence. AAAIPres [Jelektronnyj resurs]. – The MIT Press, Madison. – P. 600 – 607. – Rezhim dostupa : 1998.http://www.oracle.com/technetwork/java/javase/index.html.

Малахов К.С., Семенков В.В. Аналіз комплексних інструментальних засобів інженерії онтологій

У статті представлений огляд актуальних спеціалізованих інструментальних засобів інженерії онтологій (Інструментальний комплекс онтологічного призначення, Лота, SIMER+MIR, ІСІДА-Т, Protégé, Ontolingua, InTez, OntoSTUDIO, OntoEdit) для побудови і об'єднання онтологій, а також засобів анотування на основі онтологій. Розглянуті основні функції і можливості цих інструментальних засобів, їх достоїнства і недоліки, а також даний системний порівняльний аналіз.

***Ключові слова**: онтологія, інструментальний комплекс онтологічного призначення, ІКОП, спеціалізований інструментальний засіб, інженерія знань, проектування дисципліни*

Malakhov K., Semenkov V. Analysis of the complex software systems for ontological engineering purpose

The article presents an overview of current specialized ontology engineering tools (Tool Complex Ontological Destination, LoTA, SIMER+MIR, ICIDA-T, Protégé, Ontolingua, InTez, OntoSTUDIO, OntoEdit) for the construction and integration of ontologies and annotation tools based on ontologies. The main features and capabilities of these tools, their advantages and disadvantages, as well as a comparative analysis of the data system.

***Key words:** ontology, Instrumental Complex Ontological Destination, ICON, custom tool, knowledge engineering, design discipline*

Семенков Віталій Васильович – аспірант кафедри ІТС, Луганського національного університету ім. Тараса Шевченко; e-mail: semvitaliy@gmail.com

Малахов Кирило Сергійович – младший науковий співробітник, Ін-т кібернетики ім. В.М. Глушкова НАН України; e-mail: malahovkirill@gmail.com

Рецензент: ***Даніч В.М.,*** д.т.н., професор.

Статтю подано 27.05.14