

# Обзор инструментов инженерии онтологий

*О.М. Овдей, Г.Ю. Проскудина*  
*Институт программных систем НАН Украины*

В статье представлен обзор известных инструментов инженерии онтологий для построения и объединения онтологий, а также средств аннотирования на основе онтологий. Рассмотрены основные функции и возможности данных инструментов, их достоинства и недостатки, а также дан сравнительный анализ.

## 1. Введение

Основной вопрос при построении цифровых (электронных) библиотек состоит в том, как организовать большие количества информации так, чтобы пользователи могли найти то, что им нужно. Описывая ресурсы, библиотекари разработали сложные схемы классификации и правила каталогизации для создания метаданных. Метаданные повышают эффективность работы, обеспечивая основу для поиска, определяя интеллектуальный и исторический контекст. Цифровые библиотеки предъявляют более высокие требования к метаданным, чем традиционные библиотеки. Они содержат большие объемы информации, и доступ к ней обеспечивается с помощью большого числа разнообразных сервисов, которые помогают пользователям определять и удовлетворять свои потребности. Большинство этих сервисов должны обеспечиваться без помощи человека. Таким образом, важно чтобы метаданные, используемые в цифровых библиотеках, были приспособлены к машинной обработке. Кроме того, общим требованием для таких систем является интероперабельность на уровне обмена данными и совместной работе сервисов. Чтобы достичь семантической интероперабельности, смысл информации, которой обмениваются, должен быть понятен во всех системах. Использование онтологий для объяснения неявного и скрытого знания – возможный подход для достижения этой цели.

Онтологии - больше чем просто сложный подход к описанию и классификации информации. Они могут использоваться для поддержки функционирования и роста нового вида цифровых библиотек, реализованных как распределенные интеллектуальные системы.

Под онтологией можно понимать [1]:

- надежный семантический базис в определении содержания;
- общую логическую теорию, которая состоит из словаря и набора утверждений на некотором языке логики;

- основу для коммуникации между людьми и компьютерными агентами.

Онтологии позволяют представить новые понятия так, что они становятся пригодными для машинной обработки. С помощью онтологии можно "перекинуть мостик" между новыми понятиями, с которыми система еще не встречалась, и описаниями уже известных классов, отношений, свойств и объектов реального мира.

Компоненты, из которых состоят онтологии, зависят от парадигмы представления. Но практически все модели онтологий в той или иной степени содержат *концепты* (понятия, классы, сущности, категории), *свойства* концептов (слоты, атрибуты, роли), *отношения* между концептами (связи, зависимости, функции) и дополнительные *ограничения* (определяются аксиомами, в некоторых парадигмах фасетами).

Термин экземпляр используется для представления элементов в предметной области, т.е. элемента данного концепта. Онтология вместе с множеством отдельных экземпляров составляет *базу знаний*.

В данной работе представлен обзор инструментов инженерии онтологий: создания, редактирования, поддержки; отображения, выравнивания и объединения нескольких онтологий, а также инструментов аннотирования.

## 2. Инструменты построения онтологий

Построение онтологий – сложный и занимающий много времени процесс. Чтобы облегчить его, в середине 90-х годов начали создаваться первые среды для процесса разработки онтологий. Они обеспечили интерфейсы, которые позволили выполнять концептуализацию, реализацию, проверку непротиворечивости и документирование. За последние годы число инструментов онтологий резко возросло (сайт консорциума W3C, например, предоставляет список более чем 50 инструментов редактирования).

Инженерию онтологий можно определить как совокупность действий, касающихся:

- процесса разработки онтологий;
- жизненного цикла онтологий;
- методов и методологий построения онтологий;
- набора инструментов и языков для их построения и поддержки.

В настоящее время для создания и поддержки онтологий существует целый ряд инструментов, которые помимо общих функций редактирования и просмотра выполняют поддержку документирования онтологий, импорт и экспорт онтологий разных форматов и языков, поддержку графического редактирования, управление библиотеками онтологий и т.д.

Рассмотрим наиболее известные инструменты инженерии онтологий, URL и основные характеристики которых представлены в Таблице 1 и Таблице 2

Приложения.

Система Ontolingua [2] была разработана в KSL (Knowledge Systems Laboratory) Стенфордского университета и стала первым инструментом инженерии онтологий. Она состоит из сервера и языка представления знаний.

Сервер Ontolingua организован в виде набора онтологий, относящихся к Web-приложениям, которые надстраиваются над системой представления знаний Ontolingua. Редактор онтологий – наиболее важное приложение сервера Ontolingua является Web-приложением на основе форм HTML. Кроме редактора онтологий Сервер Ontolingua включает сетевое приложение Webster (получение определений концептов), сервер ОКВС (доступ к онтологиям Ontolingua по протоколу ОКВС) и Chimaera (анализ, объединение, интегрирование онтологий). Все приложения, кроме сервера ОКВС, реализованы на основе форм HTML. Система представления знаний реализована на Lisp.

Сервер Ontolingua также предоставляет архив онтологий, включающий большое количество онтологий различных предметных областей, что позволяет создавать онтологии из уже существующих. Сервер поддерживает совместную разработку онтологии несколькими пользователями, для чего используются понятия пользователей и групп. Система включает графический браузер, позволяющий просмотреть иерархию концептов, включая экземпляры. Ontolingua обеспечивает использование принципа множественного наследования и богатый набор примитивов. Сохраненные на сервере онтологии могут быть преобразованы в различные форматы для использования другими приложениями, а также импортированы из ряда языков в язык Ontolingua.

Protégé [3] – локальная, свободно распространяемая Java-программа, разработанная группой медицинской информатики Стенфордского университета (первая версия – 1987, последняя Protégé-2.1.1 – июнь 2004). 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 ОКВС (Open Knowledge Base Connectivity) [4] и снабжен рядом плагинов [5], 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).

OntoEdit [6] первоначально был разработан в институте AIFB (Institute of Applied

Informatics and Formal Description Methods) Университета Karlsruhe (сейчас коммерциализован Ontoprise GmbH) выполняет проверку, просмотр, кодирование и модификацию онтологий. В настоящее время OntoEdit поддерживает языки представления: FLogic, включая машину вывода, OIL, расширение RDFS и внутреннюю, основанную на XML, сериализацию модели онтологии используя OXML - язык представления знаний OntoEdit (OntoEdit's XML-based Ontology representation Language). К достоинствам инструмента можно отнести удобство использования; разработку онтологии под руководством методологии и с помощью процесса логического вывода; разработку аксиом; расширяемую структуру посредством плагинов, а также очень хорошую документацию.

Так же как и Protégé, OntoEdit – автономное Java-приложение, которое можно локально установить на компьютере, но его коды закрыты. Архитектура OntoEdit подобна Protégé (Рис.1).

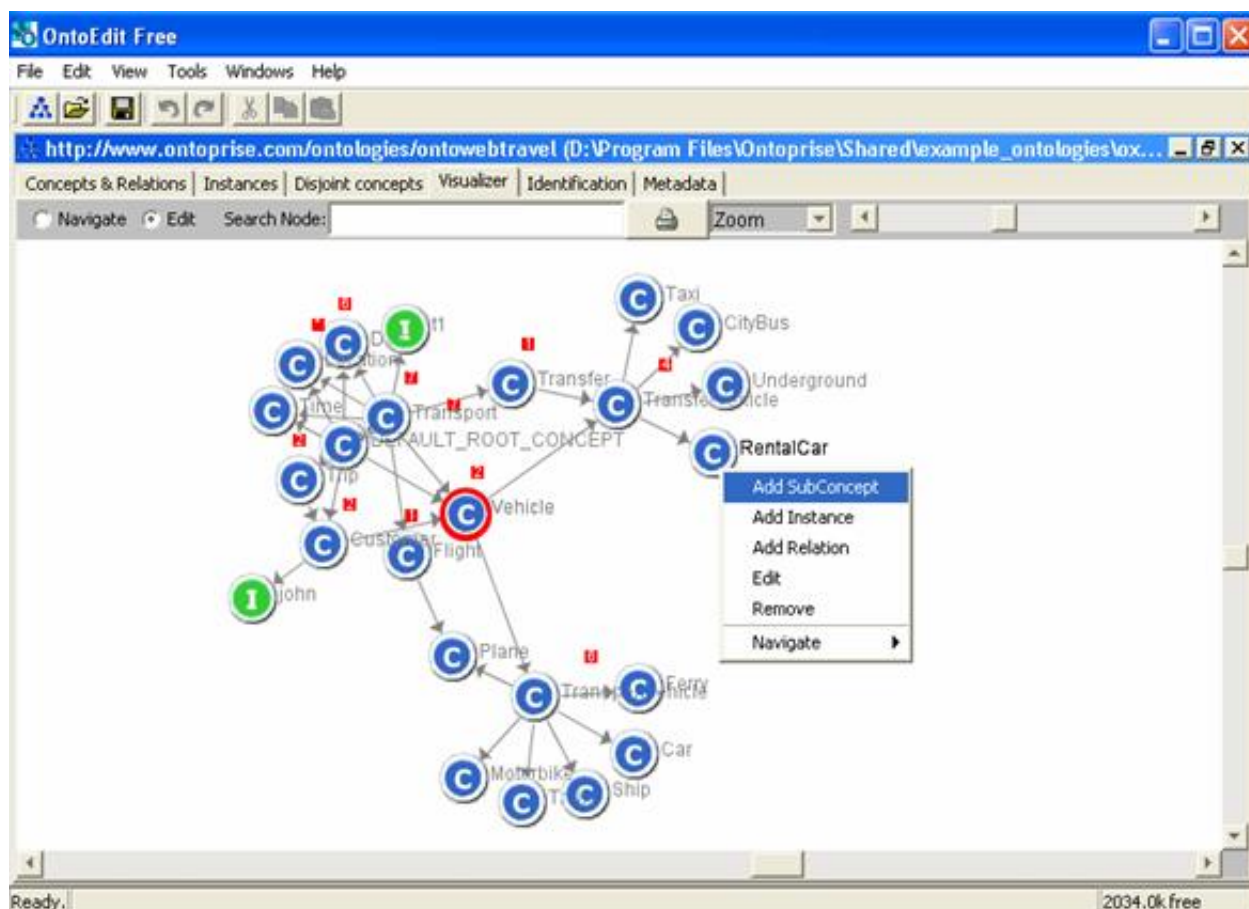


Рис. 1. Навигация по онтологии и ее редактирование с помощью OntoEdit

Существует две версии OntoEdit: свободно распространяемая OntoEdit Free (ограничена 50 концептами, 50 отношениями и 50 экземплярами) и лицензированная OntoEdit Professional (нет ограничений на размер). Естественно, что OntoEdit Professional имеет более широкий набор функций и возможностей (например, машину вывода, графический инструмент запросов, больше модулей экспорта и импорта, графический редактор правил, поддержка баз данных JDBC и т.д.).

OilEd [7] – автономный графический редактор онтологий, разработан в Манчестерском университете в рамках европейского IST проекта On-To-Knowledge. Инструмент основан на языке OIL (сейчас адаптирован для DAML+OIL, в перспективе – OWL), который сочетает в себе фреймовую структуру и выразительность дескриптивной логики (Description Logics) с сервисами рассуждения (Рис.2). Что позволило обеспечить понятный и интуитивный стиль интерфейса пользователя и преимущества поддержки рассуждения (обнаружение логически противоречивых классов и скрытых отношений подкласса).

Из недостатков можно выделить отсутствие поддержки экземпляров. Существующая версия не обеспечивает полную среду разработки – не поддерживается разработка онтологий большого масштаба, миграция и интеграция онтологий, контроль версий и т.д. OilEd можно рассматривать как “NotePad” редакторов онтологий, предлагающий достаточную функциональность, чтобы позволить пользователям строить онтологии и продемонстрировать, как можно использовать механизм рассуждения FaCT для проверки онтологии на непротиворечивость.

В последнее время наблюдается рост популярности редактора OilEd. Он используется как для обучения, так и для исследования. Инструмент свободно распространяется по общедоступной лицензии GPL.

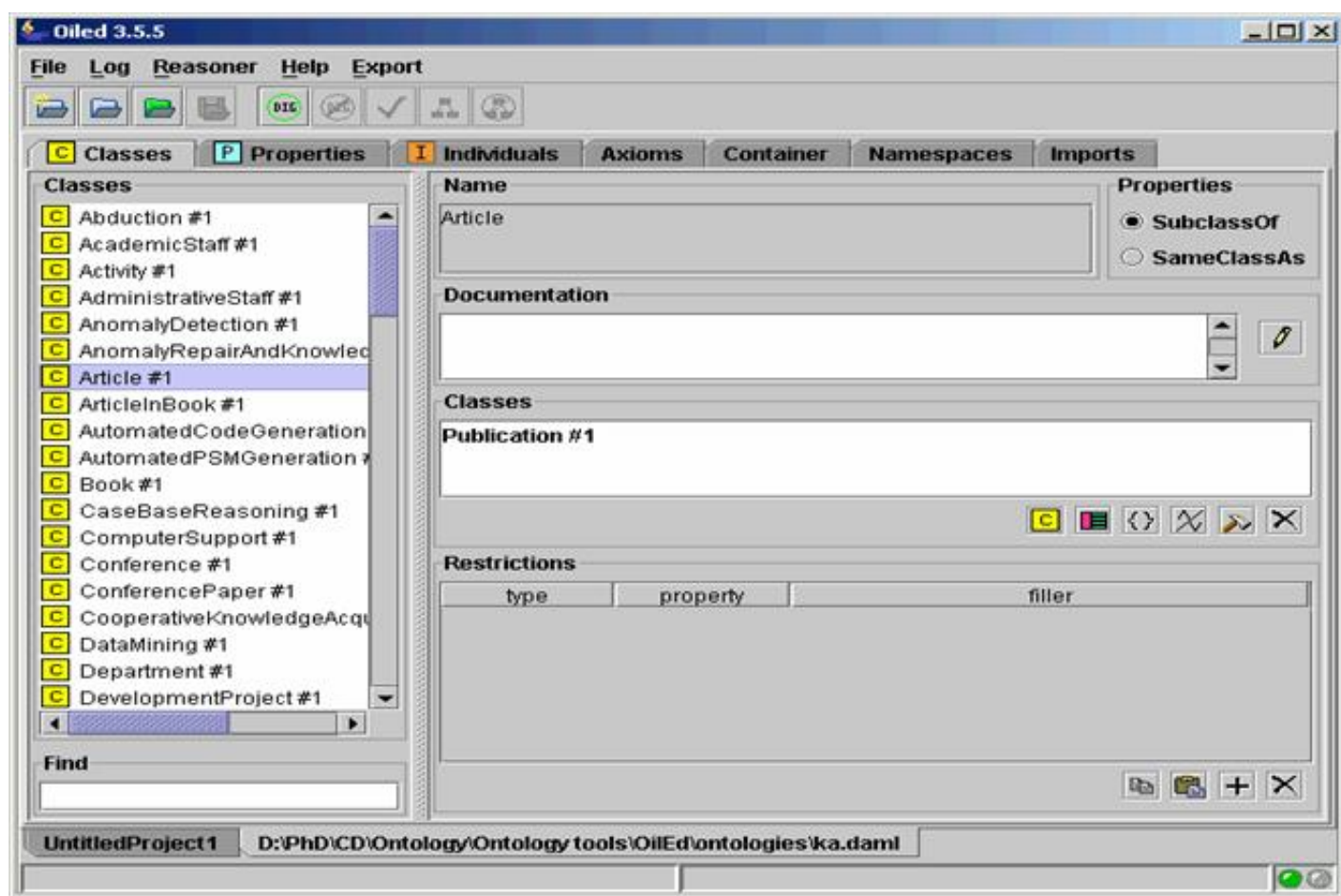


Рис. 2. Редактирование онтологии KA<sup>2</sup> с помощью редактора OilEd

WebOnto [8] разработан для Tadzebao – инструмента исследования онтологий и предназначен для поддержки совместного просмотра, создания и редактирования онтологий. Его цели – простота использования, предоставление средств масштабирования для построения больших онтологий.

Для моделирования онтологий WebOnto использует язык OCML (Operational Conceptual Modeling Language) [9]. В WebOnto пользователь может создавать структуры, включая классы с множественным наследованием, что можно выполнять графически. Все слоты наследуются корректно. Инструмент проверяет вновь вводимые данные контролем целостности кода OCML.

Инструмент имеет ряд полезных особенностей: сохранение структурных диаграмм, отдельный просмотр отношений, классов, правил и т.д. Другие возможности включают совместную работу нескольких пользователей над онтологией, использование диаграмм, функций передачи и приёма и др.

OntoSaurus является Web-браузером для баз знаний LOOM [10]. Он состоит из двух основных модулей: сервера онтологий и Web-браузера для редактирования и просмотра онтологий LOOM с помощью HTML-форм, обеспечивая для них графический интерфейс. OntoSaurus также предоставляет ограниченные средства редактирования, но его основная функция – просмотр онтологий. Но для построения сложных онтологий нужно понимать язык LOOM. Большинство пользователей строят онтологию на языке LOOM в другом редакторе, а затем для просмотра и редактирования импортируют его в OntoSaurus. В OntoSaurus реализованы все возможности языка LOOM. Обеспечиваются автоматический контроль совместимости, дедуктивная поддержка рассуждения и некоторые другие функции.

Конструктор онтологий ODE (Ontological Design Environment) [11], который взаимодействует с пользователями на концептуальном уровне в отличие от инструментов, подобно OntoSaurus, общающихся на символьном уровне. Мотивом для ODE послужило то, что людям проще формулировать онтологии на концептуальном уровне. ODE обеспечивает пользователей набором таблиц для заполнения (концептов, атрибутов, отношений) и автоматически генерирует для них код в LOOM, Ontolingua и FLogic [12]. ODE составляет часть методологии полного жизненного цикла построения онтологии согласно Methontology [13]. Инструмент получил свое дальнейшее развитие в WebODE [14], который интегрирует все сервисы ODE в одну архитектуру, сохраняет свои онтологии в реляционной базе данных, обеспечивает дополнительные сервисы (машину вывода, построение аксиом, сбор онтологий, генерацию каталогов).

KADS22 [15] – инструмент поддержки проектирования моделей знаний согласно методологии CommonKADS. Онтологии составляют часть таких моделей знаний (другая часть – модели вывода). Модели CommonKADS определены в CML (Conceptual Modeling Language). KADS22 – интерактивный графический интерфейс для CML со следующими функциональными возможностями: синтаксический анализ файлов CML, печать, просмотр гипертекста, поиск, генерация глоссария и генерация HTML.

Дальнейшее развитие в рамках проекта DWQ (Data Warehouse Quality) ведет к инструменту i.com [16], инструментальному средству поддержки концептуальной стадии проекта интегрированных информационных систем. i.com использует расширенную модель данных сущностей-связей (EER - Extended Entity-Relationship Model) дополнив ее ограничениями многомерной агрегации и промежуточных схем. Инструмент i.com полностью интегрирован с мощным сервером рассуждения на основе DL. i.com служит главным образом для интеллектуального концептуального моделирования.

### **3. Инструменты для отображения, выравнивания и объединения онтологий**

Сегодня онтологии доступны в разных представлениях. Но, что делать, когда мы находим несколько онтологий, которые бы хотели использовать, но они не соответствуют друг другу?

Инструменты объединения онтологий помогают пользователям найти сходство и различие между исходными онтологиями и создают результирующую онтологию, которая содержит элементы исходных онтологий. Для достижения этой цели они автоматически определяют соответствия между концептами в исходных онтологиях или обеспечивают среду, где пользователь может легко найти и определить эти соответствия. Эти инструменты известны как инструменты отображения, выравнивания и объединения онтологий, так как они выполняют сходные операции для процессов отображения, выравнивания и объединения.

Отображение (mapping) онтологии заключается в нахождении семантических связей подобных элементов из разных онтологий. Выравнивание (alignment) онтологий состоит в том, чтобы установить различные виды соответствия (или связи) между двумя онтологиями, а затем повторно сохранить исходные онтологии и таким образом, в дальнейшем использовать информацию друг друга. Объединение (merging) онтологий – генерация одной согласованной онтологии из двух исходных [17].

Исследователи разных областей информатики работают над автоматическим или поддерживаемым инструментально объединением онтологий (или иерархии классов, или объектно-ориентированных схем, или схем баз данных – определенная терминология изменяется в зависимости от области применения). Однако и автоматическое объединение онтологий, и создание инструментальных средств, которые бы управляли пользователем в этом процессе, находятся на ранних стадиях развития. В этом разделе представлен краткий обзор некоторых из существующих подходов.

Инструментальные средства, которые имеют дело с нахождением соответствия между онтологиями, классифицируются [18]:

- для объединения двух онтологий с целью создания одной новой (PROMPT, Chimaera, OntoMerge);
- для определения функции преобразования из одной онтологии в другую

(OntoMorph);

- для определения отображения между концептами в двух онтологиях, находя пары соответствующих концептов (например, OBSERVER, FCA-Merge);
- для определения правил отображения для связи только релевантных частей исходных онтологий (ONION).

Рассмотрим теперь вышеупомянутые средства более подробно.

PROMPT [18] - дополнение к системе Protégé, реализованное в виде плагина, служит для объединения и группировки онтологий. При объединении двух онтологий PROMPT создает список предлагаемых операций. Операция может состоять, например, из объединения двух терминов или копирования терминов в новую онтологию. Пользователь может выполнить операцию, выбирая одну из предлагаемых или определяя непосредственно операцию. PROMPT выполняет выбранную операцию и дополнительные изменения, вызванные этой операцией. Потом список предлагаемых операций модифицируется и создается список конфликтов и возможных решений этих конфликтов. Это повторяется до тех пор, пока не будет готова новая онтология (Рис. 3).

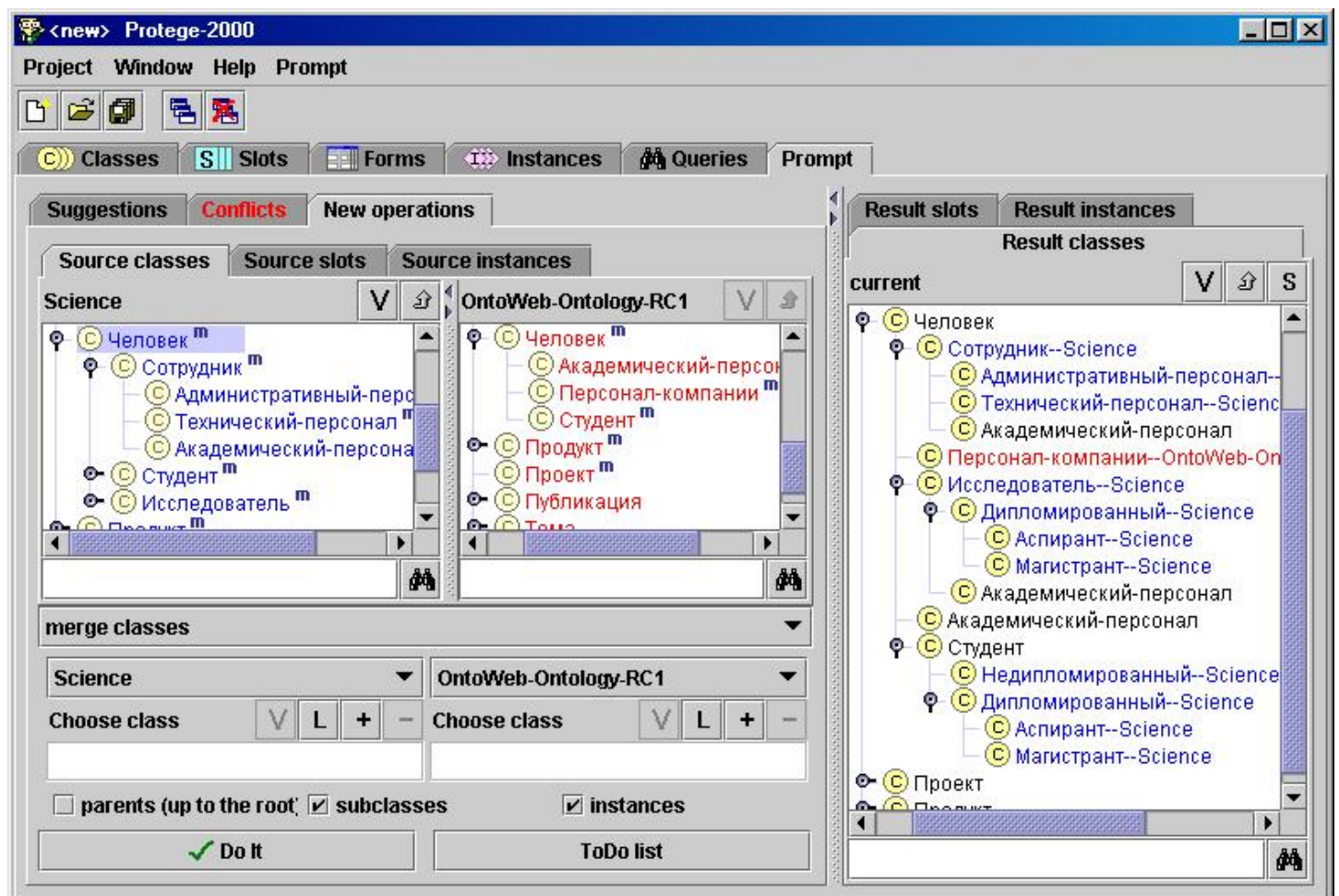


Рис. 3. Пример объединения двух онтологий при помощи Protégé и Prompt



Chimaera [19] - интерактивный инструмент для объединения, основанный на редакторе онтологий Ontolingua. Chimaera позволяет пользователю объединять онтологии, разработанные в различных формализмах. Пользователь может запрашивать анализ или руководство от Chimaera в любой момент в течение процесса объединения, и инструмент направит его на те места в онтологии, где требуется его вмешательство. В своих предложениях Chimaera главным образом полагается на то, из какой онтологии прибыли концепты, основываясь на их именах. Chimaera оставляет решение о том, что делать пользователю и не делает никаких предложений самостоятельно. Единственное таксономическое отношение, которое рассматривает Chimaera - отношение подкласс - суперкласс. Chimaera самый близкий к PROMPT. Однако поскольку он использует в своем анализе только иерархию класса, он пропускает многие из соответствий, которые находит PROMPT. Эти соответствия включают предложения по объединению слотов с подобными именами, которые относятся к объединенным классам, объединению доменов слотов, которые были объединены и т. д.

В OntoMerge [20] объединенная онтология есть объединение двух исходных онтологий и набора аксиом соединения. Первый шаг в процессе объединения в OntoMerge состоит в трансляции обеих онтологий к общему синтаксическому представлению на разработанном авторами языке. Затем инженер онтологии определяет аксиомы соединения, содержащие термины из обеих онтологий. Процесс трансляции экземпляров выглядит следующим образом: все экземпляры в исходных онтологиях, рассматриваются как находящиеся в объединенной онтологии. Затем на основе инструкций в исходных онтологиях и аксиом соединения машина вывода сделает заключение, таким образом, создавая новые данные в объединенной онтологии. OntoMerge предоставляет инструменты для трансляции данных-экземпляров в объединенную онтологию.

OntoMorph [21] определяет набор операторов преобразования, которые можно применить к онтологии. Затем человек-эксперт использует начальный список пар и исходных онтологий для определения набора операторов, которые должны примениться к исходным онтологиям для устранения различий между ними, и OntoMorph применяет эти операторы. Таким образом, совокупность операций может выполняться за один шаг. Однако, человек-эксперт не получает никакого руководства за исключением начального списка пар.

Система OBSERVER [22] применяет DL для ответа на запросы, используя несколько онтологий и информацию об отображении между ними. Вначале пользователи определяют набор межонтологических отношений. Система помогает справиться с этой задачей, находя синонимы в исходных онтологиях. Определив отображения, пользователи могут формулировать запросы в терминах DL с помощью собственной онтологии. Затем OBSERVER использует информацию отображения для формулировки запросов к исходным онтологиям. OBSERVER в значительной степени полагается на тот факт, что описания в онтологиях и запросах являются содержательными.

FCA-Merge [23] - метод для сравнения онтологий, которые имеют набор общих экземпляров или набор общих документов, аннотируемых с помощью концептов исходных онтологий. Основываясь на этой информации, FCA-Merge использует

математические методы из Formal Concept Analysis для того чтобы произвести решетку концептов, связывающую концепты исходных онтологий. Алгоритм предлагает отношения эквивалентности и подкласс-суперкласс. Затем инженер онтологии может анализировать результат и использовать его как руководство для создания объединенной онтологии. Однако предположение, что две объединяемые онтологии используют общий набор экземпляров или имеют набор документов, в котором каждый документ аннотируется терминами обоих источников слишком жесткое и на практике такая ситуация происходит редко. В качестве альтернативы, авторы предлагают использовать методы обработки естественного языка для аннотации набора документов концептами из этих двух онтологий.

Система ONION (ONtology compositiON) [24] основана на алгебре онтологии. Поэтому, она предоставляет инструменты для определения правил артикуляции (соединения) между онтологиями. Правила артикуляции обычно учитывают только релевантные части исходных онтологий. Для того чтобы предложить соединение, ONION использует лексические методы, и методы на основе графов. Метод нахождения лексического подобия между именами концептов использует словари и методы семантической индексации, основанные на местонахождении группы слов в тексте.

## **4. Инструменты аннотирования на основе онтологий**

Важнейшим предусловием реализации целей семантического Web является возможность аннотировать Web-ресурсы семантической информацией. В связи с этим в последние годы инструменты инженерии онтологий эволюционируют в сторону разработки инструментов аннотирования на основе онтологий.

Инструмент аннотации MnM [25] обеспечивает поддержку автоматической и полуавтоматической разметки Web-страниц семантическим содержанием. MnM интегрирует Web-браузер и редактор онтологии и обеспечивает открытые интерфейсы связи с серверами онтологий и инструментами извлечения информации. MnM можно рассматривать в качестве одного из первых примеров следующего поколения редакторов онтологий, на основе Web, ориентирующихся на семантическую разметку и обеспечивающих механизм полномасштабной автоматической разметки Web-страниц.

С помощью SHOE's Knowledge Annotator [26] пользователь может также описывать содержание Web-страниц. Инструмент имеет интерфейс, который отображает экземпляры, онтологии и утверждения (собранные документы). Также обеспечивается проверка целостности. SHOE's Knowledge Annotator позволяет пользователям выполнять разметку страниц в SHOE, под управлением онтологий доступных локально или через URL. Эти размеченные страницы могут быть проанализированы инструментальными средствами, знающими язык SHOE, типа SHOE Search. Аннотируемые Web-страницы могут быть также проанализированы другим инструментом по имени Expos'e, а содержание будет сохранено в репозитории. Это SHOE-знание затем сохраняется в базе знаний Parka [27].

Инструмент Metabrowser [28] также частично решает проблему аннотирования

Web-ресурсов. Он может работать, например, на базе онтологии Дублинского ядра (Дублинское ядро можно рассматривать как простейшую онтологию) и предлагать ряд возможностей для автоматического создания и просмотра метаданных. Metabrowser (включая свободно распространяемую версию), отображает метаданные Web-страницы вместе с самой Web-страницей (Рис. 4).

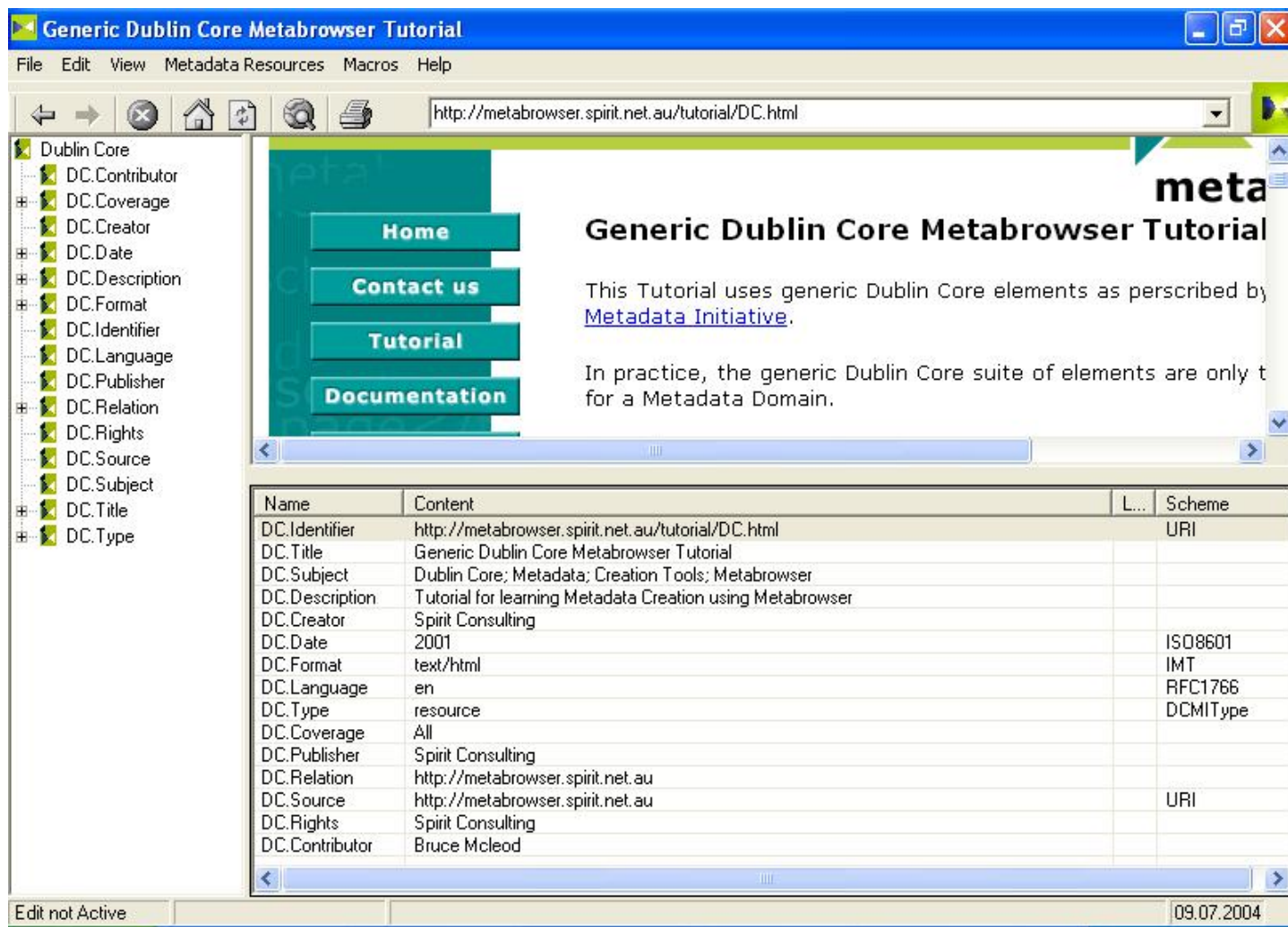


Рис. 4. Metabrowser - инструмент создания и просмотра метаданных Web-страниц на основе онтологий

## 5. Сравнительный анализ инструментов

Мы вкратце рассмотрели три группы инструментов: построения онтологий; отображения, выравнивания и объединения онтологий и аннотирования на основе онтологий. В соответствии с каждой группой инструментов попытаемся сравнить их между собой (см. Табл. 2 Приложения)

Инструменты построения онтологий можно разделить на два типа: разработанные для редактирования онтологий на определенном языке онтологий и интегрированные наращиваемые инструментальные сайты (Web-приложения, на основе форм HTML и/или Java-апплетов), большинство из которых не зависит от языка представления.

Следует подчеркнуть, что большинство из рассмотренных инструментальных средств разрабатываются университетскими исследовательскими группами, которые предоставляют открытый код, либо предлагают свободный доступ к функциям. Однако наиболее перспективные из них передаются коммерческим компаниям (например, OntoEdit Professional - лицензированный продукт).

Инструменты OntoEdit, WebODE и KADS22 дают поддержку методологиям построения онтологий, соответственно On-To-Knowledge, METHONTOLOGY и CommonKADS, что не мешает им использоваться в других методологиях или вообще без них.

Касаясь технического аспекта, а именно архитектуры программного обеспечения (локальная, клиент-серверная, n-уровневая), расширяемости, языков программирования на которых реализованы инструменты, способов хранения онтологий (в файлах или базах данных), необходимо отметить следующее.

Более ранние инструменты Ontolingua, OntoSaurus и WebOnto имеют клиент-серверную архитектуру. Protégé, OntoEdit и OilEd имеют 3-х уровневую архитектуру, где существует четкое разделение между хранением онтологий, модулями бизнес-логики приложений и приложениями интерфейса пользователя. Эти инструменты обладают большими возможностями по наращиванию (например, при помощи плагинов). Большинство инструментов хранит свои онтологии в текстовых файлах, что ограничивает размер онтологий. Только Protégé и WebODE могут хранить свои онтологии в базах данных и таким образом управлять большими онтологиями. Наконец, большинство инструментов реализовано на Java.

Выше уже говорилось о том, что модели знания инструментов определяют компоненты, которые должны использоваться при построении онтологии. Большинство инструментов представляет онтологии, комбинируя фреймы и логику первого порядка (First Order Logic - FOL). Однако это еще не означает, что они могут представлять одни и те же компоненты с одним и тем же количеством информации. Только два из перечисленных инструментов, OilEd и OntoSaurus, основаны на дескриптивной логике (DL).

Далее остановимся на некоторых свойствах редакторов онтологий. Интерфейс пользователя редакторов онтологий может быть Web-приложением, на основе форм HTML (Ontolingua, OntoSaurus и WebODE) и/или Java-апплетов (WebOnto) или локальным приложением (Protégé, OntoEdit, OilEd).

Все редакторы онтологий за исключением OilEd, Ontolingua и OntoSaurus обеспечивают графические средства редактирования и просмотра онтологий, где классы обычно представлены узлами на графах, а отношения - дугами между ними. Дополнительно к этим графическим функциям, OilEd, OntoEdit Professional, Protégé и WebODE предоставляют некоторую поддержку в написании формальных аксиом и сложных выражений.

OntoEdit, Ontolingua, OntoSaurus, WebODE и WebOnto поддерживают совместную разработку онтологий, предоставляя отдельным пользователям или группам

пользователей разрешение на доступ и написание различных наборов онтологий.

Разнообразие инструментов для отображения, выравнивания и объединения онтологий делает сложным их непосредственное сравнение. Фактически, когда разработчик должен решить вопрос, какой инструмент является наиболее подходящим, все будет зависеть от конкретной задачи. Например, если объединяемые онтологии совместно используют набор экземпляров, то лучше всех может работать FCA-Merge. Если онтологии имеют экземпляры, но совместно их не используют, и многие значения слотов содержат текст, лучшим выбором может стать GLUE. Если только части онтологий должны быть отображены, можно было бы выбрать инструмент ONION. Если онтологии имеют очень ограниченную структуру, а концепты имеют подробные определения на естественном языке (одном), инструментальные средства ISI/USC могут обеспечивать лучшие ответы. Если экземпляры вообще не доступны, и онтологии содержат много отношений между концептами, лучше всех может работать Prompt .

## Литература

1. Maedche A., Staab S. Tutorial on Ontologies: Representation, Engineering, Learning and Application // ISWC'2002.
2. Farquhar A., Fikes R., Rice J. The Ontolingua server: A tool for collaborative ontology construction // International Journal of Human-Computer Studies, 46(6), pages 707-728, 1997.
3. Musen, M. Domain Ontologies in Software Engineering: Use of Prot?g? with the EON Architecture // Methods of Inform. in Medicine, pages 540-550,1998.
4. 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/The MIT Press, Madison, P.600-607, 1998.
5. Creating Semantic Web Contents with Prot?g?-2000. N. Noy, M. Sintek, S. Decker, M. Crubezy, R. Fergerson, M. Musen // IEEE Intelligent Systems, March/April pages 60-71, 2001.
6. OntoEdit: Collaborative ontology development for the Semantic Web. Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, D. Wenke // In Proc. of the Inter. Semantic Web Conference (ISWC 2002), Sardinia, Italia, June 2002.
7. Bechhofer S.,Horrocks I., Goble C., Stevens R. OilEd: A Reason-able Ontology Editor for the Semantic Web // Joint German/Austrian conf. on Artificial Intelligence (KI'01). Lecture Notes in Artificial Intelligence LNAI 2174, Springer-Verlag, Berlin, pages.396-408, 2001.
8. Domingue J. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web // Proc. of the Eleventh Workshop on Knowledge Acquisition, Modeling and Management, KAW'98, Banff, Canada, 1998.
9. Motta E. Reusable Components for Knowledge Modelling // Ph.D. Thesis. The Open University, 1997.
10. MacGregor R. Inside the LOOM classifier // SIGART bulletin, Vol.3, No.2, pages 70-76, 1991.
11. Fernandez M, Gomez-Perez A., Pazos J. A Building a Chemical Ontology Using Methodology and the Ontology Design Environment // IEEE Intelligent Systems, Jan./Feb. pages 37-46, 1999.
12. Kifer M., Lausen G., Wu J. Logical Foundations of Object-Oriented and Frame-

Based Languages // Journal of the ACM, 1995.

13. Fernandez M., Gomez-Perez A., Juristo N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering // AAAI-97 Spring Symposium on Ontological Engineering, Stanford University, 1997.
14. Arpirez JC, Corcho O, Fernandez-Lopez M, Gomez-Perez A. WebODE: a scalable workbench for ontological engineering // First Intern. Conf. Knowledge Capture (KCAP2001). Victoria. Canada, Oct. 2001.
15. The CommonKADS Methodology. G. Schreiber, H. Akkermans, A. Anjewierden, R. Hoog, N. Shadbolt, W. Van de Velde, B. Wielinga // Knowledge engineering and management. MIT press, Massachusetts, 1999.
16. Franconi E., Ng G. The i.com tool for intelligent conceptual modeling // 7th Intl. Workshop on Knowledge Representation meets Databases, KRDB'00, Berlin, Germany, 2000.
17. Noy N., Musen M. SMART: Automated Support for Ontology Merging and Alignment // Stanford Medical Informatics, Stanford Univ., 1999.
18. Noy N., Musen M. The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping // Stanford Medical Informatics, Stanford Univ., 2003.
19. McGuinness D., Fikes R., Rice J., Wilder S. An environment for merging and testing large ontologies // In Proc. of the Seventh Int. Conf., KR2000, Morgan Kaufmann Publishers, San Francisco, CA, 2000.
20. Dou D., McDermott D., Qi P., Ontology translation by ontology merging and automated reasoning // EKAW'02 workshop on Ontologies for Multi-Agent Systems. SigÈuenza, Spain, 2002.
21. Chalupsky H, OntoMorph: A translation system for symbolic knowledge // In: Cohn A. G., Giunchiglia F., Selman B. (Eds.), Principles of Knowledge Representation and Reasoning // Proc. of the Seventh Intern. Conf. (KR2000). Morgan Kaufmann Publishers, San Francisco, CA, 2000.
22. Mena E., Illarramendi A., Kashyap V., Sheth A. OBSERVER: An approach for query processing in global information systems based on interoperation across preexisting ontologies // Distributed and Parallel Databases, An International Journal, Vol.8, No.2, 2000.
23. Stumme G., Medche A. FCA-Merge: Bottom-up merging of ontologies // 7th Int. Conf. on Artificial Intelligence, (IJCAI'01), Seattle, WA, 2001, P.225-230.
24. Mitra P., Wiederhold G., Decker S. A scalable framework for interoperation of information sources //The 1st Intern. Semantic Web Working Symp., SWWS'01, Stanford University, CA, 2001.
25. Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A., Ciravegna F. MnM: Ontology-Driven Tool for Semantic Markup // European Conf. on Artificial Intelligence (ECAI 2002). In Proc. of the Workshop Semantic Authoring, Annotation & Knowledge Markup (SAAKM 2002). Lyon France, July 22-23, 2002.
26. Heflin J. Hendler J. A Portrait of the Semantic Web in Action // IEEE Intelligent Systems, March/April, pages 54-59, 2001.
27. Stoffel K., Taylor M., Hendler J. Efficient management of very large ontologies // American Association for Artificial Intelligence Conf., (AAAI97), Menlo Park, CA, pages 442-447, 1997.
28. <http://metabrowser.spirit.net.au/tutorial/DC.html>

# ПРИЛОЖЕНИЕ

## Таблица 1. Инструментальные средства онтологий.

Имя	Описание	URL
<i>Создание онтологий</i>		
Ontolingua	Поддержка совместной разработки	<a href="http://www.ksl.stanford.edu/software/ontolingua/">http://www.ksl.stanford.edu/software/ontolingua/</a>
WebOnto	Поддержка совместного просмотра	<a href="http://webonto.open.ac.uk">http://webonto.open.ac.uk</a>
Prot?g?	Создание, просмотр, поддержка	<a href="http://protege.stanford.edu">http://protege.stanford.edu</a>
OntoSaurus	Web-браузер баз знаний LOOM	<a href="http://www.isi.edu/isd/ontosaurus.html">http://www.isi.edu/isd/ontosaurus.html</a>
ODE, WebODE	Создание, методология Methontology	<a href="http://delicias.dia.fi.upm.es/webODE/">http://delicias.dia.fi.upm.es/webODE/</a>
KADS22	Проектирование моделей знания по методологии CommonKADS.	<a href="http://www.swi.psy.uva.nl/projects/kads22/index.html">http://www.swi.psy.uva.nl/projects/kads22/index.html</a>
OntoEdit	Разработка и поддержка онтологий	<a href="http://www.ontoprise.de/products/ontoedit">http://www.ontoprise.de/products/ontoedit</a>
OilEd	Поддержка рассуждения	<a href="http://oiled.man.ac.uk">http://oiled.man.ac.uk</a>
i.com	Поддержка интеллектуального концептуального моделирования	<a href="http://www.inf.unibz.it/~franconi/icom/">http://www.inf.unibz.it/~franconi/icom/</a>
<i>Объединение и отображение онтологий</i>		
PROMPT	Приложение объединения к Prot?g?	<a href="http://protege.stanford.edu/plugins/prompt/">http://protege.stanford.edu/plugins/prompt/</a>
Chimaera	Инструмент объединения, на основе	<a href="http://www.ksl.stanford.edu/software/chimaera/">http://www.ksl.stanford.edu/software/chimaera/</a>





Версия	3.5.5 Oct2003	2.6.6 Mar2004	1.0.650 Oct2002	1.9 Mar2002	2.1.1 June2004	2.1 Mar2003	2.3 May2001
Доступность	Открытый код	Свободная лицензия	Свободный доступ	Открытый код, свободный доступ	Открытый код	Свободный доступ	Свободный доступ
Поддержка методологией	-	On-To-Knowledge	-	-	-	METH-ONTOLOGY	-
<i>Архитектура программного обеспечения</i>							
Архитектура приложения	3-х уровневая	3-х уровневая	Клиент/сервер	Клиент/сервер	3-х уровневая	n-уровневая	Клиент/сервер
Расширяемость		Плагины			Плагины	Сервер приложения	
Хранение онтологий	файлы	файлы	файлы	файлы	файлы, СУБД	СУБД	файлы
Язык ПО	Java	Java	Lisp	Lisp	Java	Java	Java+ Lisp
<i>Модель знания</i>							
Формализм	DL	Фреймы + FOL	Фреймы + FOL	DL	Фреймы + FOL	Фреймы + FOL	Фреймы + FOL
Основной язык представления знания	DAML+OIL	OXML	Ontolingua	LOOM	OKBC	-	OCML
Формальный язык аксиом	-	FLogic	KIF	LOOM	PAL	WAB	OCML
<i>Редакторы онтологий</i>							

Интерфейс пользователя	Локальное приложение	Локальное приложение	HTML	HTML	Локальное приложение	HTML и апплеты	Апплеты
Графическое редактирование таксономии концептов	-	+	-	-	+	+	+
Редактор формальных аксиом	+	-	-	-	+	+	-
Совместная разработка онтологии	-	+	+	+	-	+	+
Машина вывода	FaCT (встроенная) RACER DIG	OntoBroker	JTP	классификаторLOOM	PAL (встр.) Jess FaCT Prolog FLORA Algermon	Prolog (встроенная) Jess	Система представления знаний OCML
Проверка непротиворечивости	+	+	-	+	+	+	+
<i>Интероперабельность</i>							
С другими инструментами	-	OntoAnnotate OntoMat Semantic-Miner	Chimaera OKBC	OKBC	Prompt OKBC ArgoUML	ODE-KM ODE-SeW ODE-SWS ODE-Annotate Prot?g?	MnM
Импорт	RDF(S) OIL DAML+OIL SHIQ	OXML RDF(S) DAML+OIL FLogic	Ontolingua KIF CML IDL	LOOM PowerLOOM Stella IDL	XML RDF(S) XML Schema XMI	XML RDF(S) DAML+OIL OWL CARIN	OXML RDF(S)
Экспорт	RDF(S) OIL DAML+OIL OWL SHIQ DIG	OXML RDF(S) DAML+OIL FLogic	Ontolingua KIF LOOM CLIPS CML Epikit Prolog IDL	LOOM PowerLOOM KIF Ontolingua Stella IDL C++	XML RDF(S) XML Schema FLogic CLIPS Java XMI	XML RDF(S) OIL DAML+ OIL OWL CARIN FLogic Prolog, Jess, Java	OXML Ontolingua RDF(S)

---

## **Об авторах**

**Овдей Ольга Михайловна** - аспирант Института программных систем НАН Украины,

г. Киев

E-mail: [olga\\_ovdiy@ukr.net](mailto:olga_ovdiy@ukr.net)

**Проскудина Галина Юрьевна** - научный сотрудник Института программных систем НАН Украины,

г. Киев

E-mail: [gupros@isofts.kiev.ua](mailto:gupros@isofts.kiev.ua)

---

© Овдей О.М., Проскудина Г.Ю., 2004