УДК 519.683

А.Н. Глибовец, Н.Н. Глибовец, Д.Е. Покопцев, М.О. Сидоренко

СЕМАНТИЧЕСКАЯ ПАУТИНА И WIKI-СИСТЕМЫ

Представлен аналитический обзор современных результатов, связанных с созданием семантического Weba и целого ряда проблем, систем, технологий, активно разрабатываемых в общем направлении развития Интернета. Особое внимание уделяется исследованиям онтологий и Wiki технологиям.

Введение

Интернет – всемирная система объединенных компьютерных сетей, использующая для связи и маршрутизации пакетов данных протоколы TCP/IP. Паутина (Web) – глобальное информационное пространство, работающее на физической инфраструктуре Интернета, используя протокол HTTP и идентификаторы ресурсов URI (Universal Resource Identifier). Для улучшения автоматизированной (машинной) обработки информации необходимо совершенствовать форму ее представления [1–3]. Технология Semantic Web определяет подход к решению этой задачи путем трансформации информационного наполнения Интернета в глобальную базу знаний с помощью созданной сети документов – семантической паутины (СП), содержащих метаданные о ресурсах. Для построения такого решения требовалось разработать и реализовать новую информационно-коммуникационную модель, аналогичную семиуровневой модели OSI, но в приложении к Web и с ориентацией на обмен информацией, а не данными.

Семантическая паутина (Semantic Web) – это направление развития Всемирной паутины, целью которого является представление информации в виде, пригодном для машинной обработки на базе технологических стандартов разрабатываемых и внедряемых World Wide Web Consortium (W3C). СП предполагает запись информации в виде семантической сети с помощью онтологий, что позволяет специальной программе (агенту) непосредственно извлекать из паутины факты и делать из них логические заключения, что позволяет человеку и компьютеру эффективней взаимодействовать.

Понятие «семантическая паутина» было введено в 2001 году Тимом Бернерсом-Ли [4]. Магистральной линией разработки СП он назвал поэтапное и распределенное создание универсального языка описания данных и правил рассуждений об этих данных. Этот язык должен допускать не только простую визуализацию данных, но и легкую переносимость правил вывода, существовавших в некоторой системе представления данных, в Паутину, что превратит традиционную паутину (Web) в систему семантического уровня. При разработке такого языка задания метаданных требовалось решить две основные проблемы: он должен позволять определять выражения произвольной сложности, но эти выражения должны иметь форму, достаточно простую для понимания машиной.

Пользователь, используя уникальные идентификаторы URI, может легко выражать введенные им понятия, даже интегрировать в Интернет объекты реального мира. Универсальный язык позволяет постепенно связать все эти понятия в универсальную сеть, переведя все сайты на этот язык. Позже потребуется написать программы (агенты), обрабатывающие знания на этом языке. Таким образом, СП предусматривает объединение разнообразных видов информации в единую структуру, где каждому смысловому элементу данных будет соответствовать специальный синтаксический блок (тэг). Тэги должны составлять единую иерархическую структуру, на основе которой и должна функционировать СП.

Параллельно разрабатывалась и другая ветвь СП названая онтологическим подходом. Этот подход включает в себя

развитие средств аннотирования документов, которыми могли бы воспользоваться Web-сервисы и специализированные компьютерные программы-агенты при обработке сложных пользовательских запросов.

Консорциум W3 решил [5], что для практического использования СП достаточно разработать: универсальный язык представления знаний, использующего ссылки на онтологии (RDF); языки описания онтологий (OWL); языки описания Web-сервисов (WSDL, OWL-S); инструментарий создания и обработки документов (Jena, Haystack, Protege); языки запросов к знаниям (SPARQL); логический вывод знаний; семантические поисковые системы (SHOE); программы-агенты.

Анализу последних тенденций развития СП и посвящена данная работа.

1. Семантическая паутина

СП представляет собой систему с зачатками искусственного интеллекта. В этой паутине компьютеры могут взаимодействовать друг с другом без участия человека, а приложения автоматически распознают информацию. Для построения глобальной базы знаний паутины проект предполагает внедрение во все документы, Web-страницы и файлы специальных метаданных, указывающих на то, где, когда, кем был создан файл, как он отформатирован, для чего предназначен, а также использование расширений языка онтологий OWL (Web Ontology Language) вместе с RDF (Resource Definition Framework).

Основные функциональные возможности СП можно разбить на несколько базисных групп в соответствии с типом сервиса, который они предоставляют пользователям. Рассмотрим боле детально каждую из них [6–8].

Первую группу представляет тип сервиса названый обзором ресурсов. Они должны помочь пользователю в осознании и интерпретации связей среды разнородного контента. Типичным сценарием для таких систем является указание пользователем некоторого понятия, отправной точки и вывод в той или иной форме всего множества связанных с ним ресурсов. Для таких систем нет аналогов внутри предыдущего поколения Web. Понятно, что реализация сервиса обзора ресурсов базируется на сервисах поиска.

Последние, широко распространенные в Интернете, но для их трансформации в СП требуется существенное изменение основной функции – результатом поиска являются факты, а не контент.

Реализацией сервиса обзора ресурсов занимаются многие проекты [3, 6]. В основном они различаются используемыми методами поиска, формами представления и обработки "развёрнутых ответов", обзором контекста. Для оптимизации работы таких сервисов видится необходимым коллективное накопление знаний в форме разнообразных хранилищ знаний.

Основную функцию пополнения знаний в хранилищах несут непосредственно пользователи (пользователи социальных сетей). Ярким примером таких систем являются Wiki-системы. Характерной особенностью последних являются развитые методологии способности пополнения знания из внешних источников и их модификации. Все это обуславливает создание эффективных сервисов интеллектуальной обработки данных паутины: интерпретирование данных, построение метаданных, определение семантических связей, построение логических выводов и т. п.

В сети Интернет на момент появления СП уже были созданы специализированные хранилища больших массивов данных. Основной проблемой их прямого использования в СП была (есть) либо их разнообразная структурированность, либо ее отсутствие вообще. Анализ таких данных затруднителен не только для машины, но и для человека. Для устранения этих недостатков было предложено использование языка описания ресурсов Resource Description Framework (RDF) – низкоуровневого языка описания метаданных [9].

Смысл в нем кодируется с помощью деревьев глубины 3. Каждое дерево состоит из субъекта (подлежащего), свойства (сказуемое) и объекта (дополнение). В языке используется модель представления знаний объект-субъект-свойство, но все элементы таких триплетов должны являть-

ся уникальными идентификаторами ресурсов (URI согласно стандартам W3C – строка определенного формата, адресующая реально существующий объект). Благодаря использованию такого URI, при доступе к одному из звеньев триплета можно автоматически восстановить всю цепочку в целом, формировать сети из взаимосвязанных объектов и обеспечивается привязка каждого понятия к единому определению в Сети.

В начале RDF документа идет список ссылок н а онтологии и каждая вершина может задаваться строкой или ссылкой на объект из некоторой онтологии. Вершины могут иметь дополнительные квалификаторы. Универсальность объектов, служащих элементами логических выражений языка позволяет добиться требуемого уровня повторного использования данных и их переносимости, унификации представления информации об объекте. На основе RDF строятся более высокоуровневые языки (RDF schema, OWL), позволяющие создавать специализированные форматы для представления многообразия различных типов объектов и полноценные онтологии соответственно.

Консорциумом W3 был разработан и специализированный язык запросов к RDF-хранилищам – SPARQL, позволяющий осуществлять сложные выборки из массивов метаданных [10]. Все это позволяет нам утверждать о существовании уже сейчас формальной базы построения необходимого для СП полностью распределённого, но единого хранилища данных. Консорциум провел значительную работу как по созданию стандартов представления данных, так и разработке набора соглашений об именовании ресурсов и способах выдачи метаинформации.

На определенном уровне абстракции можно сказать, что построение и развитие СП для практического использования основывается на оптимизации использования трех базисных компонент: программ-агентов, расширяемого языка разметки XML и Web-онтологий. Чаще всего набор используемых технологий оптимизации представляют в виде специализированного «пирога», изображенного на рис. 1 [11].

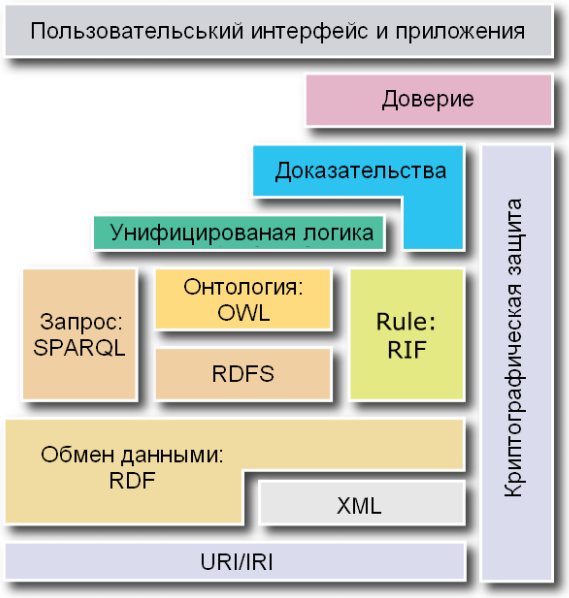

Рис. 1. «Пирог» уровней СП

Нижние слои пирога (по крайней мере до RDF + RDF Schema) уже стали реальностью. Прослойка онтологий пока частично реализована и продолжает активно совершенствоваться [12, 13]. Верхняя часть стека требует существенного развития на уровне Паутины, но успешно используется на локальном и отраслевом уровне. Нельзя сказать, что структура СП этим ограничивается, поскольку появляются новые технологии благодаря исследованиям и практическим испытаниям.

В основе семантической сети лежат три принципа: агрегация, безопасность и логика. Агрегация означает совместное использование любых данных, путем создания соответствующей семантической информация (онтологии). В основу безопасности предоставления и работы с информацией положены цифровые подписи (Crypto). Логика (Unifying Logic) – это набор правил описания информационной структуры данных, протоколы и язык описания страниц. Самый нижний уровень СП – это URI, унифицированный идентификатор, определяющий способ записи адреса произвольного ресурса и дающий возможность просто выражать те понятия, которыми пользуется потребитель.

Типичными примерами URI-идентификаторов являются URL-адреса в Интернете (ссылка на адресуемый объект, например, Web-страницу, файл или ящик электронной почты). В СП URI используются также для именования объектов, то есть каждый URI однозначно называет некоторый объект. Свои URI в СП есть не только у страниц, но и у объектов реального мира (людей), и даже у абстрактных понятий (например, у свойств «название», «сложность»). Поскольку URI глобально уникальны, они позволяют называть одни и те же предметы в разных местах в семантической паутине. При этом URI протокола HTTP (т. е. начинающиеся с http://) можно одновременно использовать как адреса документов, содержащих «машинно-читаемые» описания этих предметов.

Следующий уровень – язык XML как базовая форма разметки и средств, предназначенных для определения и описания классов XML-документов (DTD, XML-схемы), развертывания средства описания ресурсов RDF и RDF-схемы, объясняющие, как состыковывать XML-данные в сети и строить каталоги и словари понятий. RDF позволяет выполнять поиск необходимых понятий в СП.

Основное методологическое назначение модели RDF в СП состоит в описании связей (отношений) между сетевыми ресурсами и информацией. Система, работая только с тегами XML, не может найти суть их смыслового наполнения. Поэтому была начата разработка стандарта языка формального описания содержания сетевых ресурсов RDF. Фактически он является связующим звеном между XML-документами и средствами, обеспечивающими поиск и навигацию на основе логических утверждений. RDF представляет собой технологию выражения смысла терминов и понятий в виде, доступном для обработки программами. Эта технология предназначена для стандартизации определений и использования метаданных, описывающих Web-ресурсы, а также для представления самих данных, содержащихся в этих ресурсах. Использование URI для кодирования информации в документе обеспечивает единственность привязки понятия к его определению в СП. Утверждения, кодируемые с помощью RDF, в дальнейшем можно интерпретировать с помощью онтологий, созданных по стандартам RDF Schema и OWL, чтобы получать из них логические заключения.

Техническую часть семантической паутины составляет семейство стандартов, включающее XML, XML Schema, RDF, RDF Schema, OWL и другие. XML предоставляет синтаксис для определения структуры документа, подлежащего машинной обработке, не неся семантической нагрузки. XML Schema определяет ограничения на структуру XML-документа. Стандартный синтаксический анализатор языка XML в состоянии проверить произвольный XML-документ на соответствие его структуры, так называемой схеме документа, описанной в XML Schema. RDF представляет собой простой способ описания экземплярных данных в формате субъект-отношение-объект. Существует стандартизованное отображение этих троек на XML-документы предопределённой структуры (то есть консорциумом W3 определена схема XML-документов, содержащих RDF-описания), а также на другие форматы представления (например, в нотацию N3). RDF Schema описывает набор атрибутов (здесь их точнее назвать отношениями), таких, как rdfs:Class, для определения новых типов RDF-данных. Языком поддерживается также отношение наследования типов rdfs:subClassOf. OWL расширяет возможности по описанию новых типов (в частности, добавлением перечислений), а также позволяет описывать новые типы данных RDF Schema в терминах уже существующих (например, определять тип, являющийся пересечением или объединением двух существующих). Он также может использоваться для явного представления значения терминов и отношений между терминами в словарях. SWRL (Semantic Web Rule Language) [14] расширяет OWL возможностью определения Хорн-подобных правил.

Необходимость описания метаданных приводит к дублированию информации. Каждый документ должен быть создан в двух экземплярах: размеченным для

чтения людьми, а также в машинно-ориентированном формате. Потребность объединения форматов привела к созданию так называемых микроформатов [15] и языка RDFa [16]. Последний является вариантом языка RDF и отличается от него тем, что не определяет собственного синтаксиса, а предназначен для внедрения в XML-атрибуты XHTML-страниц.

Существуют и другие модификации RDF, например RSS (Really Simple Syndication) версий 0.90 и 1.0 – это семейство форматов для распространения контента (описания лент новостей, анонсов статей, изменений в блогах), основанных на стандарте XML [17]. Формат быстро стал чрезвычайно популярным за счёт узкой категоризации подмножества используемых метаданных. В нем субъектом тройки всегда является сайт-источник RSS-файла, а в качестве отношений используются самые очевидные свойства документов, имеющие отношение к часто обновляющимся источникам информации: дата написания, автор, постоянная ссылка. Заметим, что формат RSS версии 2.0, хотя и не является форматом, основанным на RDF, позволяет внедрение произвольного XML-содержимого, используя пространство имён rdf.

Микроданные (HTML microdata) – это стандарт семантической разметки HTML-страниц, с помощью атрибутов, описывающих смысл информации, содержащейся в тех или иных HTML-элементах, и позволяющие «понимать» контент веб-страниц программами-агентами (находить и извлекать необходимые данные). RSS используется для нахождения информации на сайтах, подкастах и торренткастах. Подкаст – цифровой медиа-файл или набор таких файлов, которые распространяются Паутиной. Он представляет новый способ распространения аудио и видео в Паутине, который позволяет создавать такие материалы каждому желающему. Торренткасты – RSS-потоки с прикрепленными к ним .torrent-файлами (файл метаданных в bencode формате), работающими с P2P-протоколом BitTorrent [18]. BitTórrent («битовый поток») –пиринговый сетевой протокол для кооперативного обмена файлами через Паутину. Файлы передаются частями, каждый torrent-клиент, получая (скачивая) данные части, при этом отдаёт (закачивает) их другим клиентам, что снижает нагрузку и зависимость от каждого клиента-источника, обеспечивает решение задачи избыточности данных.

Протокол не имеет системы поиска. Для каждого файла создаётся информационный сопровождающий .torrent-файл, распространяющийся через любые каналы связи: специализированные Web-сервера, домашние страницы пользователей сети, электронную почту, публикацию в блогах или новостных лентах RSS. Он содержит метаинформацию (к примеру, хэш-сумму, адрес трекера) о распределяемых данных. Данные распределяются с помощью собственного коммуникационного протокола на базе TCP/IP.

Основной принцип работы протокола: раздача файла полностью контролируется трэкером (адрес которого находится в torrent-файле), поэтому пользователь, качающий себе файл (он называется личер) сам начинает раздавать, как только скачивает первую пригодную для этого часть.

Архитектура BitTorrent предусматривает наличие у файла, выкладываемого в сеть, единственного владельца, который и заинтересован в его распространении. Именно первоначальный обладатель файла генерирует torrent-файл. Клиент, в свою очередь, загружает файл (на HTTP, FTP или просто раздаёт каким либо образом) с расширением torrent, где содержится информация об адресе владельца в Интернете, имени и размере нужного файла, а также его хеш. Это всё необходимо для отслеживания хода процесса, контроля над ним и ликвидации возможности загрузки пользователями неполного или пустого файла.

Микроформаты (microformats, иногда сокращенно μF или uF) – это способ семантически размечать сведения о разнообразных сущностях на Web-страницах, используя стандартные элементы языка HTML (или XHTML) [19]. Пользователь может воспринимать страницу с размеченным микроформатом как обычную Web-

страницу (через браузер); при этом программы-обработчики способны извлечь из такой страницы структурированную информацию, следуя определенным соглашениям.

При использовании микроформатов к существующей HTML-разметке добавляются новые составляющие, наполненные особым, заранее определенным смыслом. Например, с помощью атрибута class можно обозначить смысл того или иного HTML-элемента на странице (этот атрибут определен для всех элементов) и в дальнейшем такую разметку можно обрабатывать машинными средствами.

Каждый микроформат решает определенную, отдельную задачу. Например, hCalendar (сокращенно от HTML iCalendar) – микроформат для представления семантической информации о событиях в формате календаря iCalendar на (X)HTML-страницах. Обогатить пользовательское взаимодействие с Web-фрагментом можно с помощью визуальных элементов и встроенных (или глобальных) стилей CSS [20].

Каскадные таблицы стилей CSS (Cascading Style Sheets) фактически являются формальным языком описания внешнего вида документа созданного с использованием языка разметки. Таблицы используются создателями Web-страниц в качестве средства описания, оформления внешнего вида Web-страниц для задания цветов, шрифтов, расположения отдельных блоков. Язык может также применяться к любым XML-документам.

Форматы описания метаданных в СП предполагают проведение логического вывода на этих метаданных. Формализм, лежащий в основе обработки формата, даёт возможность делать заключения о свойствах данных, представленных в этом формате. Особенно это относится к языку OWL. Его базой являются дескриптивные логики, а сам язык разбит на три вложенных подмножества (в порядке вложенности): OWL Lite, OWL DL и OWL Full [21]. В работе [22] доказано, что задача логического вывода на метаданных с выразительностью OWL Lite принадлежит к классу P. OWL DL описывает максимально разрешимое подмножество дескриптивных логик, правда, некоторые запросы здесь могут выполняться за экспоненциальное время. OWL Full реализует все существующие конструкторы дескриптивных логик, но не каждый запрос в этом подмножестве языка может быть удовлетворен.

Язык сетевых онтологий OWL предназначен для описания классов и отношений между ними, которые присущи как для сетевых документов, так и приложений. Многие приложения могут "понимать" данные и работать с ними как с информацией, а также корректно проверять данные благодаря синтаксическому взаимодействию сетей. Такое взаимодействие требует проведения преобразования между терминами с помощью контент-анализа. Сами онтологии образуют систему, состоящую из наборов понятий и утверждений об этих понятиях, на основе которых можно строить классы, объекты и отношения. Отдельная онтология определяет семантику конкретной предметной области и способствует установлению связей между значениями ее элементов.

Класс – это концепция в онтологии, основной блок OWL. Классы традиционно образуют таксономическую иерархию в виде системы «подкласс-надкласс». Для описания классов поддерживается шесть базовых способов: именования (named), пересечения (intersection), объединения (union), дополнения (complement), ограничения (restrictions), перечисления (enumerated). Элементами классов являются индивидуальные элементы, которые в RDF будут объектами и субъектами. Кроме определения таксономии, свойства позволяют делать общие утверждения (строить факты) о элементах классов и особые утверждения об индивидах. Свойства-объекты связывают индивидуальные элементы между собой, а свойства-значения (datatype properties) – индивидуальные элементы со значениями типов данных, определенных с помощью XML. Характеристиками свойств являются симметричность, транзитивность и функциональность. К классам и свойствам применяются различные ограничения, например, огра-

ничение мощности множества, и команды для склеивания (эквивалентности) классов.

Динамическую часть СП представляют семантические Web-сервисы SWS (Semantic Web Services) – доступные через Паутину и пригодные для поиска, композиции и выполнения [23]. Семантический Web-сервис предоставляет пользователю как описание интерфейса, так и описание его семантики (что сервис делает, его предметную область, назначения и т. п.). Традиционно для описания интерфейса используют язык описания Web-сервисов и доступа к ним WSDL (Web Services Description Language), уточняющий типы передаваемых сервису данных, возвращаемые значения и генерируемые ошибки. WSDL-описания сервисов изначально были предназначены для машинной обработки. Стандарт WSDL допускает наличие в описаниях дополнительного XML-содержимого, что позволяет не выносить метаданные из WSDL-файлов. Для построения SWS используются языки RDF, RDF Schema, OWL и онтологии OWL-S, описывающие базовую терминологию предметной области. Онтология OWL-S состоит из четырех онтологий – онтологии сервиса, онтологии модели сервиса, онтологии процесса и онтологии базы. Можно рассматривать OWL-S как семантическое расширение UDDI-описания (Universal Description Discovery & Integration) – инструмента для расположения описаний WSDL, чтобы другие организации смогли их найти и интегрировать в свои системы. В этом случае, семантика сервиса характеризуется семантикой четырех его характеристик (IOPE): входных параметров (inputs), выходных параметров (outputs), предварительных условий (preconditions), эффектов выполнения (effects). Использование семантических Web-сервисов позволяет программным агентам реализовывать автоматический поиск и композицию подходящих сервисов для решения поставленных задач. Правда, ощутимый эффект от внедрения сервисно-ориентированной архитектуры пока наблюдается только в узкоспециализированных отраслях, например, в интеграции корпоративных приложений.

Базисные принципы реализации СП были использованы в разных отдельных проектах. Среди них традиционно выделяют два: Dublin Core и DBpedia. Проект «Дублинское ядро» (Dublin Core), реализуемый инициативной организацией Dublin Core Metadata Initiative (DCMI). Это открытый проект разработки стандартов метаданных, которые были бы независимы от платформ и подходили бы для использования в различных областях. DCMI занимается разработкой словарей метаданных общего назначения, стандартизирующих описания ресурсов в формате RDF [24]. Второй проект DBpedia реализовывал извлечение структурированной информации из данных, созданных в рамках проекта Wikipedia. Для этого пользователь должен запрашивать информацию, основанную на отношениях и свойствах ресурсов Википедии, в том числе ссылки на соответствующие базы данных. Проект DBpedia использует RDF для представления извлеченной информации. По состоянию на сентябрь 2011, базы данных DBpedia состоят из более чем 3,64 млн. понятий [25].

Подводя итоги общего обзора нельзя упустить список основных действующих рекомендаций W3C, связанных с существованием СП:

- XML (www.w3c.org/XML) обеспечивает синтаксис для структурированных документов, но не налагает никаких семантических ограничений на содержание этих документов.
- XML Schema (www.w3c.org/XML/-Schema) определяет структуру документов XML, а также дополняет XML конкретными типами данных.
- RDF (www.w3c.org/TR/2002/WD-rdf-concepts-20021108) позволяет описать модель данных для ресурсов и отношения между ними, обеспечивает простую семантику для этой модели данных, представляя их в синтаксисе XML.
- RDF Schema (www.w3c.org/TR/-2002/ WD-rdf-schema-20021112) предоставляет средства для описания свойств и классов RDF-ресурсов, а также семантику для иерархий-обобщений таких свойств и классов.

- OWL (http://www.w3.org/TR/owl-features/) расширенные возможности описания свойств и классов.

Онтологии составляют фундамент СП и представляют собой описание на некотором формальном языке понятий некоторой предметной области и отношений между ними [26]. Онтологии во многом похожи на тезаурусы и таксономии, но на самом деле шире их, поскольку предоставляют дополнительные средства для описания структуры описываемых данных. Поскольку по своей сути онтологии – это информация об информации, то они являются метаданными. Онтологический уровень формализует накопленные знания, определяя и объединяя терминологию различных предметных областей.

Среди наиболее используемых определений онтологии выделяют следующее [27]. Онтология – это "спецификация концептуализации предметной области", или упрощенно, документ или файл, формально задающий отношения между терминами (словарь понятий предметной области и совокупность явным образом выраженных предположений относительно смысла этих понятий). Чаще всего онтология представляется как иерархия понятий, связанных отношениями некоторых определенных видов. Такие определения онтологий используются в различных классификациях. Развитые онтологии формализуются средствами языков логики и допускают возможности логического вывода.

Разработка языка описания структурированных онтологий OWL стала в последнее время одним из наиболее важных звеньев работ по усовершенствованию СП, проводимых консорциумом W3C. Еще в 2004 году консорциум присвоил языку OWL статус рекомендованной к реализации технологии. В рамках OWL онтология – это совокупность утверждений, задающих отношения между понятиями и определяющих логические правила для рассуждений о них. Программы-агенты могут "распознавать" смысл семантических данных на Web-страницах, используя гиперссылки, ведущие на соответствующие онтологические ресурсы. Онтология может включать описания классов, свойств и их примеры (индивиды). Формальная семантика OWL описывает, как получить логические выводы на основе онтологий, т. е. получить факты, которые не представлены буквально, а следуют из семантики онтологии. Эти выводы могут базироваться на анализе одного документа или множества документов, распределенных в Паутине.

На практике создание онтологий начинается с построения одной или нескольких иерархий классов понятий, составляющих предметную область, каждый из которых может иметь подклассы, представляющие собой более точные понятия, чем исходный класс [28]. Классы могут содержать атрибуты, которые описывают свойства и внутреннюю структуру понятий, лежащих в основе классов. Фундаментальным таксономическим конструктором для классов является rdfs:subClassOf. Он связывает более частный класс с более общим классом. Если X – подкласс Y, то каждый представитель X – также представитель Y. Отношение rdfs:subClassOf является транзитивным. Если X – подкласс Y, и Y – подкласс Z, то X – подкласс Z.

Все подклассы наследуют атрибуты родительских классов. Каждый атрибут класса помимо названия имеет тип значения, разрешенные значения, число значений (мощность). Традиционно онтологии содержат и экземпляры классов (классы, в которых установлены значения всех их атрибутов). Это позволяет плавно переходить от построения онтологий к построению баз знаний.

Процесс разработки онтологии традиционно включает: выделение глоссария терминов (понятий) для исследования свойств и характеристик представленных в нем терминов; построение списка точных определений терминов из глоссария; построение деревьев классификации понятий (иерархии классов) на базе таксономических отношений; выделение из не задействованных при составлении деревьев

классификации понятий атрибутов классов и их возможных значений для установки основных связей между классами; необязательное добавление экземпляров классов; создание экспертами правил логических выводов для манипулирования и извлечения новых знаний. Этот процесс похож на проектирование классов в объектно-ориентированном программировании. Отличием является подход к принятию решения. Программист проектирует, базируясь в основном на методы классов. Разработчик онтологии принимает аналогичные решения на основе структурных свойств классов.

Для описания онтологий используются в основном традиционные языки описания онтологий (Interlinguas, CycL); языки, основанные на дескриптивных логиках (LOOM); языки, основанные на фреймах (OKBC, OCML, Flogic) и языки, основанные на Web-стандартах (XOL, UPML, SHOE, RDF с RDFS, DAML, OIL, OWL). Языки различаются предоставляемыми средствами описания предметной области и механизма логического вывода. Каждый из них имеет свои преимущества и недостатки. Особое место занимает язык онтологий OWL. Он выступил решающей компонентой интеллектуализации, базисом для построения семантических сетей.

Представлениям знаний в СП присущи универсальные выразительные возможности, синтаксическая и семантическая интероперабельность. Семантическая интероперабельность реализуется, например, в онтологиях путем установлением соответствия между используемыми терминами.

Онтологии различаются по многим параметрам, поэтому исследователи выделяют различные основания для их классификации. Эдуард Хоув [29] производит деление в зависимости от набора элементов, содержащихся в них, а также от типов отношений.

Классификация онтологий возможна и по количеству включенных в нее понятий.

Первая классификация онтологий проходит на основе анализа уровней.

Онтология верхнего уровня (top-ontology) насчитывает от 100 до 3000 концептов и содержит наиболее абстрактные категории, такие например как: «сущность», «явление», «роль», «объект», «процесс».

Онтологии среднего уровня (mid-level ontology) составляют приблизительно 500-100000 концептов. Их особенность в том, что они могут представлять реальный мир как, в общем, так и в частном случае, но требуют использования многих аксиом.

Наиболее распространенные – онтологии нижнего уровня, или так называемые онтологии предметной области (domain ontologies). Они содержат 2000-20000 концептов и набор отношений, специфических для конкретной области. Для них можно создавать много аксиом и правил. Типичными представителями таких онтологий являются SCTG (Standard Classification of Transported Goods), RosettaNet, The United Nations Standard Products and Services Codes (UNSPSC), NAICS (North American Industry Classification System).

Еще один вид онтологий – лексические или лингвистические, они связаны с семантикой грамматических элементов. К ним относятся WordNet, MikroKosmos, Sensus и другие. Такие онтологии применяются к задачам, связанным с обработкой естественного языка.

Особенно важна классификация онтологий – по степени выразительности. В работе [29] спектр выразительности онтологий представлен в виде, показанным на рис. 2.

Выделяются две основные группы – это легкие и тяжелые онтологии, которые уже в свою очередь делятся на 8 подкатегорий:

- список термов (term list) – список ключевых слов, который обычно используется для ограничения значений определенных свойств;
- тезаурус определяет отношения между термами;
- неформальная таксономия определяет явную иерархию обобщения и специализации, однако не поддерживает строгого наследования;

Рис. 2. Уровень выразительности онтологий

- формальная таксономия обеспечивает строгое наследование;
- онтология на основе фреймов или классов/отношений подобная объектно-ориентированным моделям: класс определяется по месту в иерархии и свойствами, вытекающие из подклассов, реализованных в сущностях;
- ограничение области значений приобретает силу для свойств и может быть проведено по типу данных или предметной области;
- с помощью логических условий значения свойств можно еще больше ограничивать;
- онтологии с наибольшей отчетливостью часто используют ограничения: отношение-целое, инверсные отношения, дизъюнктивное покрытия и т. п.

Тяжелые онтологии очень результативны, что было проверено на секторальном уровне, однако такие системы не толерантны к нецелостности. С другой стороны, легкие онтологии не позволяют делать столь качественный логический вывод, однако здесь на нестрогие онтологические договоренности не так сильно влияет нецелостность. Известная фраза Джима Хендлера "A little semantics goes a long way" [30] выражает одну из идеологий современного развития СП. Поэтому легкие онтологии стали наиболее распространенными и применяемыми.

Интеллектуальные агенты

Интеллектуальные агенты – специализированные компьютерные программы, в большинстве случаев использующиеся при обработке сложных пользовательских запросов. Запускаются на выполнение пользователем, находят, используя информацию о пользователе (профиль пользователя), подходящее решение и в удобном виде предоставляют это решение. В роли агентов в СП часто выступают и различные Web-сервисы.

Для удовлетворения запроса агент может параллельно или последовательно обращаться к разным сервисам, пока не будет найдено решение или его отсутствие в СП. Здесь можно провести полную аналогию с поисковыми системами [31]. Структура такого взаимодействия показана на рис. 3.

Агенты обмениваются информацией и правилами логических выводов, используемых в онтологиях, цепочками построенных ими рассуждений и профилями пользователей. Разрешение на обмен информацией между агентами дается либо пользователем, либо происходит автоматически, путем использования различных методов проверки, например посредством цифровых подписей. Развитие СП существенно изменяет работу поисковых систем [32]. Используя онтологии, поисковые движки (search engine), специальные

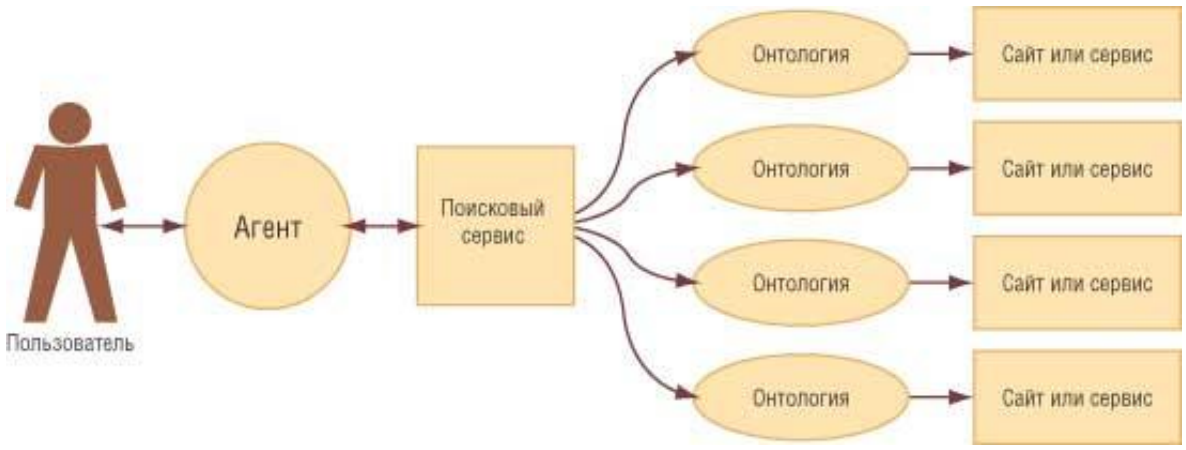

Рис. 3. Поиск информации через поисковый сервис

логические движки (logical engine) они могут существенно повысить качество и точность поиска. В языке OWL имеется средства реализации описанного взаимодействия агентов. За основу были взяты разработки из языка общения автономных агентов DARPA Agent Markup Language (DAML).

Основные принципы функционирования автономных агентов следующие: агент не имеет полной информации, необходимой для решения поставленной задачи; обрабатываемые данные распределены в сети; вычисления выполняются агентами асинхронно; взаимодействие агентов между собой и с человеком происходит на высоком семантическом уровне; отсутствует глобальный контроль над деятельностью всей системы агентов. Создаются агенты с использованием различных технологий (CORBA, EJB, .NET) и языков [33]. Программные агенты создаются больше всего на Java, учитывая направленность языка на сетевое программирование и независимость от платформы.

Linked Data.Термин Linked Data (связанные данные (СД)) предложен Тимом Бернерсом-Ли в 2006 году в его заметке об архитектуре Weba [34]. Linked Data – это метод выявления, совместного использования и объединения структурированных данных в СП для более оптимального использования информации.

В СП люди преимущественно публикуют неструктурированные документы и устанавливают связь между ними с помощью гиперссылок. СД меняют парадигму распространения документов на парадигму распространения данных. Эта технология призвана сделать данные доступными, формируя сеть данных, которую называют «Webом данных» (Web of Data). СД объединяет желающих сделать свои данные общедоступными. Тимоти Бернерс-Ли предложил для этого очень простые правила [34]:

- разнообразные объекты и понятия «именуются», и эти имена суть строки синтаксиса URI;
- где только можно, URI начинаются с http://..., чтобы получать информацию об объекте простым вводом его имени в адресную строку браузера;
- если клиент запрашивает URI и может принимать данные в виде RDF, они передаются клиенту для дальнейшей обработки. Если есть техническая возможность, то желательно предоставить возможность выборки этих данных стандартными средствами и в стандартных форматах (например, используя язык и протокол SPARQL);
- описание одного объекта или понятия не должно быть изолированным и самодостаточным. Оно должно содержать ссылки на связанные понятия, опять же в виде URI.

Благодаря СД, связи между объектами реального мира отображаются на связи между ресурсами в Интернете, используя которые программные агенты могут накапливать факты о реальности в специализированных базах знаний (RDF могут содержать связные «метаданные»). Избыточность данных не составляет проблемы.

Важно, что они доступны, и определенная их часть из множества RDF-документов может быть автоматически отобрана для дальнейшей обработки.

Специализированные приложения, использующие данные из СД, позволяют существенно улучшить обработку запроса пользователя. Многими из таких приложений мы пользуемся и не думая, что они работают с СД. Примером, такого приложения может служить Amarok, определяющее автора песни, ее название и т.п. Разнообразие данных и лёгкость их связывания дают простор для их использования в различных предметных областях с различными механизмами построения вывода. Простота представления данных в виде графа с вершинами трёх видов и дугами разных типов на первый взгляд существенно ограничивает возможности их обработки. К счастью это не так. Такое представление характерно только для процесса ввода-вывода. Внутри хранилища мы можем использовать стандартные типы SQL, все типы XML Schema или полную коллекцию базовых типов. Тесная связь представления данных в виде графа с реляционными СУБД дает повод говорить о возможности использования развитого аппарата работы с данными последних. Несомненно, в модели RDF плохо с поддержкой обработки физических величин, которые невозможно представить в виде литерала. Но запись этих величин при необходимости может уточняться определенной структуризацией. Проблемы, связанные с громадными объёмами обрабатываемых данных, по объёму схемы обрабатываемых данных или по числу запросов, могут быть решены в СД при широком использовании возможностей cloud computing [35] и универсальных онтологий.

Среди существующих решений для работы с данными в СП выделяют систему с открытым кодом Protégé [36], коммерческий аналог TopBraid Composer [37] и Oracle.

TopBraid Composer определяется как профессиональная среда разработки для RDF, RDFS, OWL, SWRL, SPARQL, предоставляющая полный набор средств для покрытия всего жизненного цикла разработки семантических приложений. TopBraid Composer имеет схожие Protégé пользовательский интерфейс и функциональность. Оба средства с помощью доступных механизмов поддержки рассуждений позволяют просто выполнять классификацию и проверку логической целостности на основе OWL.

Наибольшими коллекциями входящими в реестр Linked Data есть: LOD (Linking Open Data) Cloud, Climate Data, International Development Data, Public Domain, Bibliographic Data, Energy Data, Art. Крупнейшими поставщиками Linked Data являются правительства Великобритании и США, компания BBC, газета NY Times, сеть магазинов Best Buy, CNET, Dbpedia. На сентябрь 2010 года проект LOD содержал 220 объединенных наборов данных и составлял 24 миллиарда RDF триплетов, среди которых Data.gov и data.gov wiki – 11.5 млрд., LinkedGeoData – 3 млрд., UniProt – 1.1 млрд., DBpedia – 1 млрд., US Census Data – 1 млрд., Freebase – 0.1 млрд.

2. Организация коллективных и персональных хранилищ знаний Wiki

Многие эксперты традиционно делят информацию на две части – знания и оперативные сведения. Оперативные сведения – это данные, которые требуются в определенный момент в текущем месте для принятия решения. От скорости принятия решения зависит во многом достижение цели. Системы комплексного анализа, которые создаются для обработки оперативной информации, позволяют применять такую информацию раньше, чем она устареет. Однако сам процесс добывания оперативных сведений является достаточно сложным и затратным.

С развитием СП и специализированных технологий типа GooglDocs [38], Cloud computing возникли новые возможности построения коллективных и персональных баз знаний. В принципе, можно унифицировать требования к их построению, если отбросить тот факт, что персональная система организации знаний не требует коллективного доступа. Поэтому,

требования к системе организации знаний в основном включают: обеспечение коллективного доступа, поддержку сетевого взаимодействия, возможность работы с гипертекстом, предоставление облегченного процесса разметки, возможность загрузки файлов различных форматов и привязка их к конкретной теме, наличие системы контроля версий.

Одна из особенностей Web 2.0 – это создание контента совместными усилиями пользователей. Миллионы людей создают коллекцию ресурсов, содержащую информацию, которая, благодаря постоянному обновлению, является актуальной, образуя *социальную паутину* (*Web*). Выделяют наиболее типичные применения социального Weba: блоги, Wiki, сайты, социальные сети.

Социальная сеть – идеальная структура для распространения информации. Она имеет большой успех среди пользователей благодаря своей простоте и понятности, что способствует активному сотрудничеству. Например, если раньше браузер воспринимался людьми исключительно как средство просмотра информации, то теперь это простой, мощный и удобный инструмент создания контента в социальной паутине. Феномен простоты социальной паутины вдохновил миллионы пользователей к созданию огромного массива информации, который может обогатить СП достаточным количеством данных для зарождения новой эры паутины.

С другой стороны, из-за больших объемов неструктурированной информации, социальная паутина также страдает и требует для дальнейшего развития внедрения определенных семантических стандартов и понятной семантики данных. Социальная и семантическая паутины могут дополнять друг друга для решения общих проблем. Однако, это довольно непростая задача, соединить два разных мира: один, понятный людям, другой – машинам. Одним из вариантов ее решения есть использование интенсивно развивающейся технологии Wiki для построения хранилищ коллективных и персональных знаний [39].

Wiki – Web-сайт, структуру и содержимое которого пользователи могут самостоятельно изменять с помощью инструментов, предоставляемых сайтом. Форматирование текста и вставка в него различных объектов производится с использованием Wiki-разметки.

Термин «вики» был введен в 1995 году Уордом Каннингемом при разработке первой Wiki-системы «Портлендского хранилища образцов» с целью упростить создание и документирование образцов программ. Он называл систему «простейшей онлайн-базой данных, которая может быстро функционировать» [40] и отражала понятие «быстрый». Позже этому слову был придуман английский кроним "What I Know Is…" («то, что я знаю, это…»).

Позже Wiki-системы получили большое распространение, в основном в различных энциклопедиях, и сейчас вышли на уровень коммерческого использования в корпоративных хранилищах знаний. Развитием подобных исследований заинтересовались крупные компании, многие из них сейчас разрабатывают свои Wiki-системы. В частности, разработкой собственной системы занимается и Майкрософт. Существуют персональные Wiki-системы [41].

Технология Wiki может изменить и многие традиционные формы работы с информацией. Например, одной из наиболее распространенных функций Wiki-систем является функция RSS-потоков, используемая для создания новых документов или внесения изменений в документе. В потенциале она может заменить наиболее распространенный сейчас способ обмена информацией – электронную почту. В этом случае, важнейшим инструментом внутреннего документооборота станет не почтовый клиент, а RSS-клиент, что приведет к изменению уже привычной концепции рабочего общения и поможет решить проблемы «неполучения письма/не прочтение важной информации» и другие.

Далее опишем сущность концепции Wiki, используя материалы книги [42].

Пользователям разрешено редактирование любой страницы или создание новых страниц на Wiki-сайте, используя обычный Web-браузер без каких-либо его расширений. При редактировании имеется

возможность сравнения редакций и восстановления ранних версий, проявления изменений сразу после их внесения, разделения содержимого на именованные страницы. Связи между разными страницами поддерживаются за счёт почти интуитивно понятного создания ссылок на другие страницы. Wiki ориентированы на постоянных посетителей сайта, и инициирует их к непрерывному процессу создания и сотрудничества по изменению сайта. Для разметки текста используется так называемая Wiki-разметка, которая позволяет легко и быстро размечать в тексте структурные элементы и гиперссылки; форматировать и оформлять отдельные элементы.

Существует два базовых принципа работы Wiki-систем: первые работают с контентом, именование ссылок тут имеет лишь второстепенное значение; другие – делают акцент на аннотациях. Разные по уровню семантических возможностей и формализации Wiki-системы имеют характерные общие черты: аннотации ссылок, представление информации в соответствии с контекстом, улучшенная навигация, семантический поиск и поддержка логического вывода [43].

Для создания Wiki-среды необходимо особое программное обеспечение, которое называют Wiki. Wiki-движок – набор программ, служащий для преобразования Wiki-разметки в удобочитаемое представление на языке HTML и взаимодействия пользователя с базой данных (знаний). Традиционно он довольно прост, ибо почти все действия по структурированию и обработке содержимого делаются пользователями вручную.

Понятен и основной недостаток Wiki-систем – возможность изменять содержимое всем желающим. Для его устранения разработаны развитые средства восстановления содержания с использованием понятия версийности и дублирования. Но появляется новая проблема – согласованность содержимого. Постоянное дублирование данных обуславливает возможность содержания одной информации на нескольких разных страницах. Ее изменение на одной странице приводит к потребности отслеживания соответствующего обновления на всех остальных страницах. Доступ к знаниям Wiki-систем затруднен из-за большого объема информации на сайте и ручной обработки результатов запросов. Возникают здесь и классические проблемы повторного использования знаний. Невозможность использования типизированных свойств порождает огромное количество тэгов или категорий.

Устранить эти недостатки должны семантические Wiki (СВ). Они предоставляют пользователю удобные средства для добавления семантической разметки информации. Впервые термин СВ был употреблен Энди Динглеем (Andy Dingley) в телеконференции Usenet comp.infosystems.www.authoring.site-design, а в научно-технический обиход он введен в работе Лео Зауэрмана (Leo Sauermann) [44].

СВ – Web-приложение, использующее обрабатываемые машиной данные со строго определённой семантикой. Последнее позволяет существенно расширить функциональность Wiki-системы. Например, разрешение использования в СВ типизированных ссылок между статьями, типов данных внутри статей, а также информации о страницах позволяет говорить о возможности работы с метаданными. В СВ структурированные данные хранятся либо прямо в тексте страниц (например, Semantic MediaWiki), либо отдельно (например, Ontowiki). В первом случае используется расширенная Wiki-разметка, во втором – специальный интерфейс ввода данных (форму), отдельный по отношению к содержимому статей.

Ссылки между статьями несут в своем имени информацию о типе связи. Например, KiWi позволяет связывать структурированные данные с помощью средств RDF, а затем соотносить RDF-термины с текстом в статье. Многие СВ добавляют семантические аннотации автоматически (ACEWiki) и позволяют изменять способы представления содержимого страниц с их помощью. Контекстное представление включает отображение статей, близких к данной и информации, которая может быть выведена из базы зна-

ний. У большинства СВ имеется возможность получения дополнительной информацию о связи (ссылке), что позволяет реализовать новые способы навигации и семантический поиск. Возможность построения многокритериального запроса на формальном языке, подобном SPARQL существенно улучшает поиск, а использование семантических аннотаций позволяет реализовать фасетный поиск и уточняющий поиск за счет оптимизации путем фильтрации результатов [45].

Например, при поиске слова «Иванов» пользователь после ввода этого слова в поисковую строку, используя поисковые фасеты, фильтрует результаты поиска: выбирает категорию «Ученые» и время жизни – 1900–2000 год.

Большинство СВ хранят данные в форматах СП или предоставляют возможность импорта/экспорта в RDF и OWL. Языком запросов к Wiki часто служит SPARQL, что обеспечивает поддержку логического вывода в системе.

В разработке первых СВ (Platypus Wiki [46] и Rhizome Wiki [47]) ударение сделано на создание средств редактирования RDF-содержимого. RDF-данные представлялись как свободно редактируемый текст, не связанный с неструктурированным содержимым Wiki-разметки. Это обеспечивало импорт RDF-данных, но проверку непротиворечивости и классификации невозможно реализовать.

Существует два основных подхода к продуцированию структурированных данных [44]: получение структурированных данных из уже существующих источников информации и прямое создание семантического контента.

Первый подход предлагает пользователям создавать контент паутины с помощью существующих Web-приложений социальной сети. Структурирование данных (экспорт информации из баз данных, преобразования или реализация соответствия между открытыми семантическими стандартами, например SIOC) и их семантическая интерпретация, определение смысла тегов (онтологии MOAT, SCOT) получаются из сети автоматически с помощью специализированных программ. Этот процесс включает построение смысловых связей из неструктурированного материала (обработка естественного языка, извлечение информации), определение сути (группировка подобных тегов, формирование результирующих концептов из них) с использованием внешних ресурсов.

При прямом создании семантического контента пользователям сразу предоставляются специализированные программные средства создания структурированных (размеченных специальным образом) данных. К ним можно отнести семантические блоги, семантические закладки, семантические десктопы, семантические аннотации, семантические Wiki. К сожалению, только СВ, во многом благодаря коллаборативной природе функционирования, способны продуцировать новые концепты и онтологии [48].

Чаще всего семантизация разметки достигается путем обогащения (аннотацией) ссылок текстом, который описывает их значение. Например, в статье о Киеве, ссылку на статью Украина можно именовать как «is capital» или «located in». Такое аннотирование позволяет значительно улучшить визуализацию информации (демонстрация контекстуальной информации), навигацию (доступ к релевантной информации), поиск (поиск по контексту, а не только по тексту).

Существует два вида взаимообогащения онтологий и Wiki: системы, в которых Wiki используются для построения онтологий и системы, использующие онтологии для работы Wiki [49]. В большинстве случаев Wiki-системы выступают как внешний интерфейс коллаборативной системы разработки онтологий. Примерами таких систем являются Semantic MediaWIki, PlatypusWiki. Во втором подходе онтологиии применяются для улучшения самой Wiki-системы (IkeWiki, SWIM, SweetWiki). Существуют СВ, которые не подпадают под эту классификацию – это коллаборативные системы разработанные и применяемые исключительно для создания хранилищ знаний, не совмещая это с редактированием текста. К таким системам относится AceWiki. В последний

воплощен интересный подход. Благодаря применению специализированного языка ACE, формальные утверждения Wiki выглядят как утверждения на английском языке. Рассмотрим подробнее каждый из подходов, сделав обзор главных представителей каждого.

3. Современные семантические Wiki

Среди новых движков семантических Wiki выделяют движки с четким разделением структурированной и неструктурированной информации (Ikewiki [50], OntoWiki [51]), либо включением семантических аннотаций в Wiki-разметку (WikiSAR [52] и Semantic MediaWiki). Стали развиваться и новые типы Wiki-систем, например, персональные.

Персональные Wiki позволяют организовать информацию на своём компьютере или мобильном устройстве в форме традиционной Wiki системы, но только для личного использования. Например, многопользовательские с возможностью организации личного пространства (DokuWiki – простой, но достаточно функциональный Wiki-движок, который может быть использован для создания любой документации) и однопользовательские Wiki-приложения, не имеющие в Web-сервера и сервера баз данных (WikidPad – бесплатная программа, написанная на языке Python, позволяющая хранить списки дел, контакты, заметки и другую информацию с использованием Wiki-разметки).

На данный момент существует множество различных реализаций технологии Wiki. Они направлены на различные платформы, написанные на разных языках, предоставляют разные возможности. В таблице приведен краткий обзор наиболее распространенных систем.

Таблица. Обзор наиболее распространенных Wiki-систем

Название	Ссылки	Платформа/ Язык	Комментарий
NoodleWiki	http://flangy.com/dev/asp/noodle/	ASP	Может использовать базу данных Access или запись в файлы
OpenWiki	http://www.OpenWiki.com	ASP	Поддержка SQL Server, MSDE, Oracle и My SQL
WikiAsp		ASP	Использует MS-Access.
AwkiAwki	http://awkiawki.bogosoft.com/	Awk	Быстрая небольшая Wiki-система
WikicWeb	http://c2.com/cgi-bin/wiki?WikicWeb	C	
DidiWiki	http://didiwiki.org/	C	Простая Wiki-система со встроенным http-сервером
EddiesWiki	http://c2.com/cgi-bin/wiki?EddiesWiki	C++	Система с встроенным Web-сервером,
WikiCpp	http://wikicpp.sourceforge.net/	C++	
DotNetWiki	http://www.dotnetwiki.org/	C#	
FlexWiki	http://www.flexwiki.com/	C#	Система от Microsoft
CLiki	http://www.cliki.net/	CommonLisp	
FreeWiki	http://sourceforge.net/projects/freewiki/	Java	
JavaWiki	http://c2.com/cgi-bin/wiki?JavaWiki	Java	
JspWiki	http://www.jspwiki.org	Java	
WebMacroWiki	http://www.webmacro.org/WebMacroWiki	Java	
TiddlyWiki	http://www.tiddlywiki.com	JavaScript	Wiki-система, состоящая из одного html-файла и не требует Web
CitiWiki	http://wiki.cs.cityu.edu.hk/citiwiki	PHP	
PmWiki	http://www.pmichaud.com/pmwiki	PHP	
WackoWiki	http://wackowiki.org/WackoWiki	PHP	Удобная и быстрая Wiki-система
MediaWiki	http://www.mediawiki.org/	PHP	Наиболее известная Wiki-система (на ней базируется сайт Wikipedia.org)

Semantic Media Wiki (SMW) одна из наиболее распространенных Wiki-систем [53]. Она базируется на использовании семантических аннотаций и фактически является расширением движка (плагина) MediaWiki (MW), на котором работает Википедия. Система имеет средства добавления новых элементов Wiki-разметки, которые позволяют размечать страницы типизированными свойствами и обеспечивают доступ к данным с помощью структурированных запросов. MW – это хорошо проверенная, надежная и удобная платформа, позволяющая превратить обычную Wiki-систему в семантическую без серьезных усилий. Существует много инструментов для работы с семантикой в виде расширений для MW, обеспечивающие удобство настройки и управления функционалом Wiki (комплекты SMW + Semantic Bundle). Архитектура SMW показана на рис. 4, который взят из [53].

Основной принцип работы – это задание Wiki-ссылкой семантических аннотаций. SMW отображает эти аннотации с помощью языка онтологий OWL DL в формат OWL/RDF. Разметка ведется с помощью Web-интерфейса, приспособленного к внешнему использованию. Каждая Wiki-статья отвечает одному онтологическому элементу, а каждая аннотация в статье формирует утверждения только об одном элементе. Такое ограничение имеет ключевое значение для эксплуатации, поскольку знания используются повторно, и конечные потребители должны знать, откуда эта информация была получена. Таким образом, SMW создает невидимый семантический слой, что позволяет системе выглядеть как Wiki, а функционировать как Linked Data.

Уточним, как основные онтологические понятия реализованы в SMW:

- категория – это аннотация, позволяющая пользователям классифицировать Web-страницы. Категории были уже возможны в MW, расширение SMW только предоставило им формальную интерпретацию в виде классов OWL;
- отношение – позволяют пользователям определить связи между статьями с помощью имен ссылки;
- атрибуты – используются для определения взаимосвязи статей и сущностей (дата, количество и т. д.). Для реализации атрибутов существует несколько

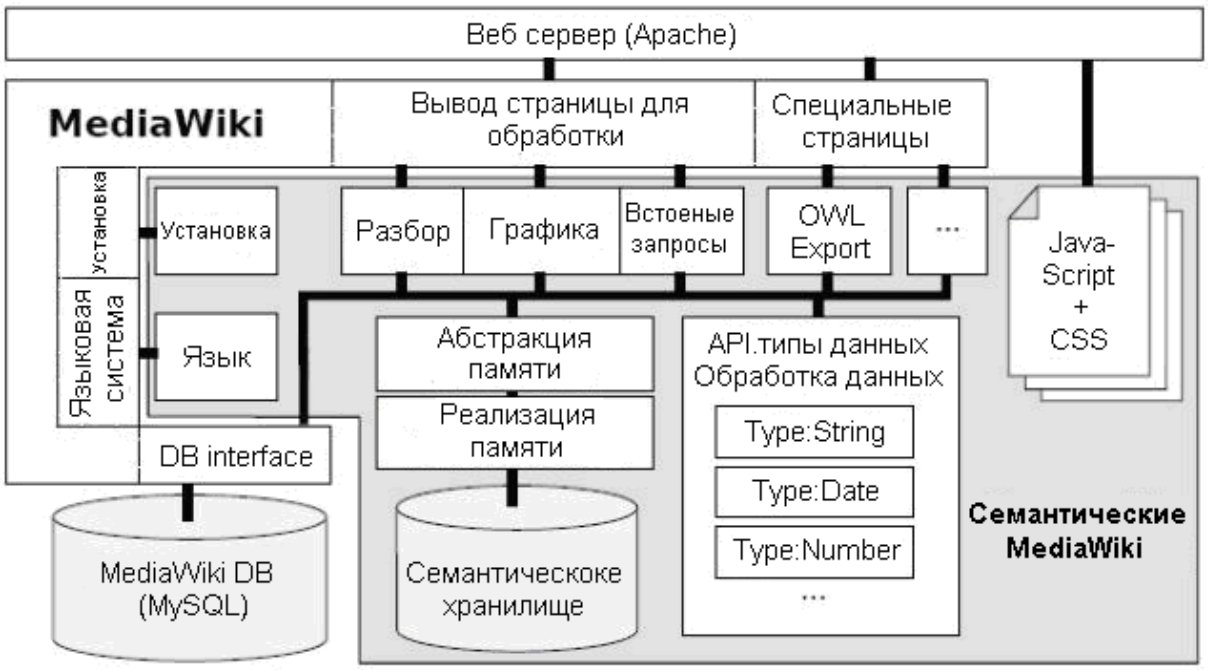

Рис. 4. Архитектура SMV

типов данных: целое число, дата, температура, географические координаты, электронная почта и т. п.

Тип элементов для большинства видов аннотаций фиксированный. Статьи публикуются как OWL-экземпляр, категория – как OWL-класс, а отношение становятся OWL-отношениями (object properties). Атрибуты, в зависимости от собственного типа у Wiki, могут иметь свойства типа данных, свойство аннотации или объекта.

SMW имеет специальный инструмент Import Vocabulary для импорта и повторного использования словарей, принадлежащих документам и стандартам Semantic Web. Это дает возможность соотносить аннотации SMW в общепринятые словари (например, FOAF, SIOC) путем их ассоциации с терминами Wiki.

Вместе с тем SMW имеет инструмент запросов, обеспечивающий нетривиальный поиск внутри системы с применением диапазонов значений, знаков подстановки и подзапросов. Функционал формирования запросов может использоваться и для создания страниц с динамическим контентом. Например, простой запрос [[Category: Event]] [[Start date :: ≥ Jan 01 2012]] [[Start date :: ≤ Dec 31 2012]], встроенный в страницу создает список мероприятий в сфере Semantic Web, которые состоятся 2012 году.

Система представления данных SMW позволяет избавиться как от некоторых проблем несовершенства текстового поиска, так и нарушения целостности данных в традиционных Wiki-системах. Wiki-статьи содержат много информации, однако зачастую пользователь желает получить конкретный ответ на поставленный вопрос. Для преодоления этой проблемы в MW были введены статьи-списки. Они состоят из ссылок на другие статьи, объединенные одной тематикой. Однако и этот подход имеет много недостатков, поскольку пользователь при создании статьи должен собственноручно вносить изменения в списки, которых она касается. Аналогично, при изменении информации в статье из списка, нужно вручную править статью-список. SMW благодаря механизму динамического создания страниц и большим количеством форматов вывода контента помогает решить эту проблему.

Одним из главных недостатков SMW является отсутствие встроенной поддержки моделирования правил. Поэтому, она не дает возможности из явных утверждений Wiki получать метаданные. Однако, по Жи Бао [54], использование шаблонов и расширений позволяет смоделировать несколько наиболее употребительных и полезных типов правил в SMW: моделирование логического следствия OWL, логических программ, проверка на ограничение целостности.

Роль расширений в системе SMW очень важна.

Приложение Triple Store Connector позволяет соединить Wiki с RDF-хранилищем и использовать SPARQL для запросов. Версия 1.6 работает с любым RDF-хранилищем, поддерживающим SPARQL и SPARQL Update.

Данные, созданные в SMW, могут легко передаваться в форматах CSV, JSON и RDF наружу. Это дает возможность Wiki быть источником данных для внешних приложений, исполняя роль, которую обычно выполняют реляционные базы данных. А с использованием расширений External Data и Semantic Result Formats, несколько семантических Wiki могут использовать данные друг друга, устраняя необходимость в дублировании и ручной синхронизации. Дополнения Data Import, Data Transfer и External Data позволяют использовать данные извне: с Web-сервисов, ресурсов Linked Data, уже существующих систем. Таким образом, система, построенная на SMW, может выполнять роль информационного хаба, который собирает и синхронизирует данные отовсюду.

Расширение Semantic Forms предлагает не опытным Wiki-пользователям интерфейс для ввода данных с помощью форм с поддержкой автозаполнения полей. Приложение Halo Extension имеет целью сделать работу людей еще удобнее,

обеспечив WYSIWYG редактором, обозревателем онтологий и семантической панелью инструментов. Расширение Ontology Editor является платформой для совместной разработки легких онтологий. Оно обогащает систему мощными интерфейсами редактирования и создания онтологий, статистике, импортом и экспортом фолксономий и онтологий, алгоритмом восстановления знания.

Учитывая широкое распространение и сообщество активных пользователей, SMW можно считать наиболее успешной семантической Wiki-системой. Она нашла применение во многих направлениях, в частности в крупных биовики: SNPedia, Neurolex. Среди других удачных примеров применения следует привести Shortipedia-Wiki, сайт для сбора фактов [55], который завоевал третье место на Semantic Web Challenge 2010.

IkeWiki и KiWi (IW) олицетворяет направление построения семантических Wiki [55], в котором используются онтологии для улучшения самой системы. В IW реализован функционал построения логического вывода для поддержки пользователя в выполнении задач. Система предлагает и WYSIWYG редактор, который пригодится пользователям, которые не имеют опыта работы с Wiki-редактором.

IkeWiki платформа написана на языке Java. Данные хранятся в СУБД Postgres, однако существует разграничение текста и структуры документов. При необходимости, они возвращаются пользователю в удобном формате XML (для текста) или RDF (для структуры).

База знаний в системе представлена RDF фреймворком Jena. Часть RDF-хранилища является SPARQL-движком, обеспечивающим поиск в системе и базе.

Для аннотаций существует три вида редакторов. Редактор метаданных позволяет заполнять текстуальные метаданные, касающиеся страницы (данные Dublin Core или RDF комментарии). Редактор типа позволяет ассоциирование страницы с одним или несколькими типами, внедренными в систему. Редактор ссылок обеспечивает управление аннотациями ссылок. В IkeWiki доступны аннотации, определяющиеся логическим выводом. Например, если ссылка из статьи «Kyiv» до «Ukraine» именовать как «capitalOf», то система автоматически ассоциирует тип «Capital» страницы, описывающий Киев, и этот тип не может быть удален пользователем.

IkeWiki больше не развивается как самостоятельная система. Ее продолжением стала система KiWi [56], которая унаследовала большинство характеристик. Онтологическая единица этой системы определяется с помощью URI и включает текст понятный человеку в формате XHTML. Можно сделать KiWi-систему частью Linked Open Data, что в свою очередь, делает возможной интеграцию контента с другими сервисами Semantic Web.

В системе KiWi реализован фасетный поиск, способный совмещать поиск метаданных (типы, теги, люди), текстовый поиск и поиск в базе данных (название, дата). Более сложный поиск находится в состоянии разработки.

Среди особенностей системы KiWi следует отметить настраиваемую страницу пользователя под названием Dashboard. Она позволяет удобно отслеживать и применять в работе: поток деятельности (например, обновление к элементам контента), рекомендации (предлагает пользователю другой контент с помощью различных рекомендательных алгоритмов), историю (список элементов, которые просматривал или редактировал пользователь), теги (список тегов используемых пользователем). Кроме этого, Dashboard это и профиль пользователя, и список его друзей. Так, в системе KiWi делают упор на функционале социальных сетей, поскольку он помогает определить контекст, над которым работает пользователь.

OntoWiki [51] предназначена для разработки баз знаний. Платформа ставит на первое место представления данных в формате RDF. Для машинной

обработки система поддерживает различные RDF-форматы и RDFa, Linked Data и SPARQL-интерфейсы. Знания в системе представлены так называемой "информационной картой", обогащенной удобными интерфейсами для визуализации и редактирования контента (WYSIWYG редактор RDF, контроль версий, статистика, поддержка сообщества и т.д.). Каждый узел (представлен страницей системы) в информационной карте связан с соответствующим цифровым источником.

OntoWiki спроектирована для работы с онтологиями любого размера и пытается поддерживать создание онтологий "с нуля". Эта платформа не только позволяет применять частично определенные шаблоны из репозитария шаблонов, но и создавать собственные. Она поддерживает коллаборативную разработку путем отслеживания изменений, возможностью комментировать и обсуждать каждую часть базы знаний, позволяя оценивать и ограничивать количество контента и поощрять пользовательскую активность.

Семантический поиск системы представлен как поиск в локальном RDF-хранилище с помощью SPARQL-запросов и предлагает несколько способов навигации: таксономический и иерархический поиск, фасетный поиск и текстовый поиск. В сочетании с поисковым роботом, который ищет, загружает и сохраняет любые RDF-документы с Weba, OntoWiki можно легко превратить в семантическую поисковую систему.

Freebase. Система Freebase (FB) заинтересовала Google и приобретена компанией летом 2010 года. Эта платформа позволяет пользователям определять собственные схемы для моделирования различных типов данных и управлять связанной структурированной информацией. FB – это большая колаборативная база знаний. Она содержит значительно меньше триплетов чем DBpedia (в 10 раз), однако качество данных значительно лучше.

Платформа FB пытается интегрировать различные подходы к формированию базы знаний. Кроме Wiki-стиля, она имеет возможность собирать информацию автоматически с различных ресурсов, таких как Wikipedia и MusicBrainz. Некоторые сущности могут быть смоделированы и добавлены к системе людьми. Однако в ней не просто создавать собственные типы. Она имеет и сложный интерфейс. Все атрибуты должны иметь типы и границы определенные самой системой. FB имеет строгие ограничения схемы, что может вызвать неудобства в опубликовании данных.

Web-интерфейс предоставляет пользователям удобную возможность искать, принимать, редактировать и организовывать информацию. Основной способ доступа к FB – через API на основе протокола HTTP. Все операции по запросам происходят на специально разработанном языке MQL (Metaweb Query Language).

Недостатком платформы является сложность присоединения к внешнему ресурсу изнутри FB. Система связывает данные друг с другом с помощью значений атрибутов, однако она не может подключиться к другим источникам на уровне данных.

Заключение

Информационные ресурсы Сети составляют свыше десятка миллиардов документов (Web-страниц) и их количество существенно увеличивается с каждым днем. Для поиска информации в этой распределенной, полнотекстовой базе данных необходимо использовать самые мощные ИПС. Необходимость анализа больших объемов неструктурированных или слабо структурированных данных очень часто усложняют процесс принятия решений. Для подобного анализа нужны технологии другого типа, представленные системами добычи знаний. Примером такой системы может быть система mSpace. Она представляет собой набор мощных инструментов, позволяющих собирать данные из различных источников, организовывать информацию по категориям и дающих воз-

можность пользователю свободно ориентироваться в ней.

СП уже вполне готова к широкому внедрению в корпоративном секторе. Все ее основополагающие технологии становятся стандартами, а крупные участники рынка высоких технологий внедряют их в прикладные программы корпоративного уровня.

Для совершенствования процессов создания, копирования и доступа к знаниям предлагается использовать системы управления знаниями, основанные на технологии Wiki. Множество реализаций Wiki-систем позволяют выбрать наиболее полезную систему, а условия распространения позволяют создать персональную или корпоративную базу знаний, используя наиболее подходящие семантические Wiki. Это значительно упростит создание развитого, более функционального информационного пространства.

Представленный в работе обзор может использоваться разработчиками прикладных интеллектуальных систем для выбора адекватных средств построения распределенных баз знаний.

1. *Андон Ф.И., Гришанова И.Ю., Резниченко В.А.* Semantic Web как новая модель информационного пространства Интернет // Проблеми програмування. – 2008. – № 2/3 (спец. вип.). – С. 417–430.
2. *Андон Ф.И., Яшунин А.Е., Резниченко В.А.* Логические модели интеллектуальных информационных систем. – К.: Наук. думка, 1999. – 396 с.
3. *Глибовец Н.Н., Глибовец А.Н., Шабинский А.С.* Применение онтологий и методов текстового анализа при создании интеллектуальных поисковых систем // Проблемы управления и информатики. – 2011. – № 6. – С. 95–103.
4. *Berners-Lee T., Hendler J., Lassila O.* The Semantic Web. Scientific American, 2001, 284, С. 34–42.
5. *World* Wide Web Consortium(W3C) – Режим доступа: http://www.w3.org – Название с экрана. (Загружено 21.04.2011)
6. *Андон П.І., Дерецький В.О.* Засоби координації агентів у пошукових архітектурах Web // Проблеми програмування. – 2004. – № 1. – С. 60–70.
7. *Пелещишин А.М., Березко О.Л.* Аналіз сучасних концепцій розвитку середовища WWW // Вісник Національного університету "Львівська політехніка": Комп'ютерні науки та інформаційні технології. – 2006. – № 565. – С. 57–64.
8. *Segaran T., Evans C., Taylor J.* Programming the Semantic Web. Build Flexible Applications with Graph Data. – O'Reilly Media, 2009. – P. 302.
9. *Mahola E., Miller E., eds* RDF Primer W3C Recommendation (2004) – Режим доступа: http://www.w3.org/TR/xhtml-rdfa-primer/
10. *SPARQL* Query Language for RDF. W3C Recommendation 15 January 2008 – Режим доступа: http://www.w3.org/TR/rdf-sparql-query/
11. *Bratt S.* Semantic Web, and Other Technologies to Watch, World Wide Web Consortium (2007). – Режим доступа: http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/#(24) – Название с экрана. (Загружено 21.04.2011)
12. *Berners-Lee T.* Linked data. World Wide Web design issues. (2000). – Режим доступа: <http://www.w3.org/DesignIssues/LinkedData.html> – Название с экрана. (Загружено 03.02.2011)
13. *Андон П., Дерецький В.* Проблеми побудови сервіс-орієнтованих прикладних інформаційних систем в semantic Web середовищі на основі агентного підходу // Проблеми програмування. – 2006. – № 2-3. – (спец. вип.). – С. 493–502.
14. *SWRL*: A Semantic Web Rule Language Combining OWL and RuleML: W3C Member Submission 21 May 2004. – Режим доступа: http://www.w3.org/Submission/SWRL/
15. *Microformats* – Режим доступа: http://microformats.org/
16. *RDFa* 1.1 Primer. Rich Structured Data Markup for Web Documents: W3C Working Group Note 07 June 2012. – Режим доступа: http://www.w3.org/TR/xhtml-rdfa-primer/
17. *RSS*: Really Simple Syndication: 21 Mar 2007. – Режим доступа: – technet.microsoft.com/en-us/.../secrssinfo
18. *BigTorrent* – Режим доступа: ru.wikipedia.org/wiki/BitTorrent
19. *Allsopp J.* Developing with Web Standards. – New Riders. – 2010. – 432 p.
20. *What* is CSS? – Режим доступа: http://www.w3.org/Style/CSS/
21. The Species of OWL.– Режим доступа: http://www.w3.org/TR/owl-guide/#OwlVarieties
22. *OWL* Full, OWL DL and OWL Lite. –

Режим доступа: http://www.w3.org/TR/owl-ref/#Sublanguage-def
23. *Semantic* Web Services. – Режим доступа: http://www.ai.sri.com/daml/services/
24. *Dublin* Core Metadata Element Set, Version 1.1 – Режим доступа: http://dublincore.org/documents/dces/
25. DBpedia. – Режим доступа: http://en.wikipedia.org/wiki/DBpedia
26. *Хатян О.А.* Онтології як інструментальна складова дослідження інформаційного простору // Інформаційна безпека людини, суспільства, держави. – 2010. –№ 1(3). – С. 71–77.
27. *Gruber T.R.* A translation approach to portable ontologies // Knowledge Acquisition. – 1993. – V. 5(2). – P. 199–220.
28. *Corcho O., Fernandez-Lopez M., Gomez-Perez A.* Methodologies, tools and languages for building ontologies. Where is their meeting point? Data & Knowledge Engineering. – 2003. – 46(1). – P. 41–64.
29. Hovy E. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. Proceedings of the 1st I nternational Conference on Language Resources and Evaluation (LREC): Granada, Spain. 1998) – Режим доступа: http://www.isi.edu/natural-language/people/hovy/papers/98LREC-ontol-align.pdf
30. *Hendler J.* A little semantics goes a long way. – Режим доступа: http://www.cs.rpi.edu/~hendler/LittleSemanticsWeb.html
31. *Глибовець А.М., Шабінський А.С.* Один підхід до побудови інтелектуальної пошукової системи // Наукові записки НАУКМА. Комп'ютерні науки. – 2010. – Том 112. – С. 26–30.
32. *Анісімов А.В., Глибовець А.М., Жаб'юк В.Я.* Основні архітектурні принципи побудови програмних систем реалізації мобільних агентів // Вісник Київського національного університету імені Тараса Шевченка. – серія: фізико-математичні науки. – Випуск № 3. – 2008. – С. 125–131.
33. *Глибовець А.М., Гороховський С.С., Жаб'юк В.Я.* Вирішення проблем побудови платформи мобільних агентів за допомогою технологій Java, Jini // Вісник Київського національного університету імені Тараса Шевченка. – серія: фізико-математичні науки. – Випуск № 4. – 2008. – С. 109–115.
34. *Linking* Open Data. W3C SWEO Community Project (22.10.2010). – Режим доступа: http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData – (Загружено 21.05.2011.)
35. *Глибовець М.М., Глибовець М.М., Бондар Є.О., Гороховський С.С.* Хмарні обчислення. Проблеми і перспективи // Вісник Київського університету. – 2011. – № 1. – С. 74–81.
36. *Protégé.* Режим доступа: http://protege.stanford.edu/
37. *TopBraid* Composer. – Режим доступа: http://www.topquadrant.com/products/TB_Composer.html.
38. *GooglDocs*. – Режим доступа: docs.google.com/
39. *WhatIsWiki.* – Режим доступа: http://wiki.org/wiki.cgi?WhatIsWiki
40. *Portland* Pattern Repository. – Режим доступа: http://c2.com/ppr/
41. *TiddyWiki* a reusable non-linear personal веб notebook – Режим доступа: http://www.tiddlywiki.com
42. *Bo Leuf*, Ward Cunningham, The Wiki Way: Quick Collaboration on the Web, Addison-Wesley. – 2001. – 436 p.
43. *Schaffert S.* IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. – 2006. – P. 388–396.
44. *Sauermann L.* The Gnowsis. Using Semantic Web Technologies to build aSemantic Desktophttp – Режим доступа: www.dfki.uni-kl.de/~sauermann/papers/sauermann2003.pdf
45. *English J., Hearst M., Sinha R., Swearingen K., Yee K.P.* Hierarchical faceted metadata in site search interfaces // CHI'02 extended abstracts on Human factors in computing systems. ACM New York, NY, USA. – 2002. – P. 628–639.
46. *P Castagna*, SE Campanini. Towards a semantic wiki wiki web. – Режим доступа: http://www.tecweb.inf.puc-rio.br/semweb/space/Platypus+Wiki/platypuswiki.pdf
47. *Souzis A.* Building a Semantic Wiki // IEEE Intelligent Systems. – 2005. – Vol. 20. – N 5. – P. 87–91.
48. *Shakya A., Takeda H. and Wuwongse V.* Community-driven Consolidated Linked Data, in M. Sheth ed., Semantic Services, Interoperability and Web Applications: Emerging Concepts. – 2011. – P. 228–258.
49. *Buffa* M et al. SweetWiki: A semantic wiki // Journal of Web Semantics. – 2008. – 6(1). – P. 84–97.
50. *Völkel M.* et al. Semantic Wikipedia // J. of Web Semantics. – 2007. – N 5. – P. 251–261.

51. *Auer S., Dietzold S., Riechert T.* OntoWiki–A tool for social, semantic collaboration // ISWC 2006, 5th International Semantic Web Conference, Athens, GA, USA. – 2007. – P. 736–749.
52. *SAR* – Режим доступа: http://en.wikipedia.org/wiki/SAR.
53. *Krötzsch M., Vrandečić D., Völkel M.* Semantic MediaWiki // Lecture Notes in Computer Science. – 2006. – P. 935–942.
54. *Bao J.* et al. – Rule Modeling Using Semantic MediaWiki // 3rd Annual Conference of the International Technology Alliance (ACITA'09), Maryland, USA, 2009.
55. *Vrandecic D.* et al. – Shortipedia: Aggregating and Curating Semantic Web Data. Proceedings of the ISWC, Shanghai. – 2010.
56. *Schaffert S.* et al. KiWi – A Platform for Semantic Social Software. 4th Semantic Wiki Workshop at ESWC09 Heraklion, 2009.

Получено 19.07.2012

Об авторах:

Глибовец Андрей Николаевич,
кандидат физико-математических наук,
доцент кафедры сетевых технологий
НаУКМА,

Глибовец Николай Николаевич,
доктор физико-математических наук,
профессор,
декан факультета информатики
НаУКМА,

Покопцев Дмитрий Евгеньевич,
магистр,

Сидоренко Марина Олеговна,
ассистент кафедры информатики
НаУКМА.

Место работы авторов:

Национальный университет
“Киево-Могилянская Академия”,
254070, Киев-70,
ул. Григория Сковороды, 2,
Тел.: (044) 463 6985,
факс.: (044) 416 4515,
e-mail: glib@ukma.kiev.ua