*УДК 004.652.8+004.91*

*Черкашин Евгений Александрович,  
к. т. н., старший научный сотрудник,  
Институт динамики систем и теории управления СО РАН им. В.М. Матросова,  
 тел. +7(914)870-67-54, e-mail: eugeneai@irnok.net,  
Орлова Ирина Витальевна,  
к. ф.-м. н., доцент,  
Иркутский национальный исследовательский технический университет  
тел. +7(914)921-07-47 e-mail: soobshenie\_1@mail.ru*

ИНСТРУМЕНТАРИЙ СОЗДАНИЯ ЦИФРОВЫХ АРХИВОВ ДОКУМЕНТОВ НА ОСНОВЕ СВЯЗАННЫХ ДАННЫХ

*E.A.Cherkashin, I.V.Orlova*

INSTRUMENTAL TOOLS FOR CONSTRUCTION OF DIGITAL ARCHIVES OF DOCUMENTS BASED ON LINKED DATA

**Аннотация.** Рассматривается задача разработки инструментария создания цифровых архивов документов. Особенностью данного инструментария является использование форматов связанных данных (семантического веба) для представления и хранения логической структуры документа и его метаинформации, системы логического вывода, встроенной в сервер, и средств создания документов из отдельных частей. Данные средства позволяют расширить функции хранения и доступа к документам функциями их интеграции с другими сайтами интернет. Рассмотрен пример приложения разработанного программного инструментария при подготовке документации о кусах, преподаваемых в университете.

Ключевые слова: *открытые связанные данные, компонентное проектирование, цифровой архив, создание документов.*

**Abstract.** *We consider the problem of development of software tools for construction of digital archives of text documents. The main feature of the tools is the usage of Linked Open Data (Semantic Web) for representation and storing the logical structure of a document and its metainformation, logical inference system integrated in the server, and the tools of document composition from other parts. Linked Open Data allows us to extend the storage, indexing and retrieval functions of the documents with functions of data integrations with other web documents. An example of the instrumentation tools application for study course documentation authoring is considered.*

**Keywords:** *Linked Open Data, component architecture, digital archive, document authoring.*

Введение

Технология открытых связанных данных (Linked Open Data, LOD) [1] предложена W3C-консорциумом для представления семантической информации в публикуемых данных таким образом, чтобы обеспечить не только возможность ее обработки при помощи программных агентов (Семантичекий Веб), но и связать всю имеющуюся информацию в единый семантический граф при помощи отношений и универсальных глобальных идентификаторов ресурсов (URI). Дескриптивные возможности технологий семантического веба, средства публикации документов HTML5, а также технологии LOD создают инфраструктурный базис средств публикации документов с обеспечением логических связей между документами. При этом содержательная информация документа связывается с содержательной информацией другого при помощи отношений – ссылок на соответствующие ресурсы. Ресурсы представляют собой как статическое содержимое, так и результаты запуска процедур публикации (конвертации) данных, порождающими размеченное текстовое содержимое.

Важным преимуществом использования LOD при создании информационных сред является ослабление требований к хранилищам публикуемой информации: сам документ является хранилищем данных в формализованном виде. В некоторой степени это позволяет время, затрачиваемое на проектирование структуры базы данных для хранилищ частичноструктурированных документов, перенаправить на процесс решения предметной задачи: пользователь (разработчик) обрамляет текстовые данные семантической разметкой.

Целью исследования является создание средств разработки цифровых архивов, позволяющих проектировать и реализовать предметно-ориентированные хранилища информационных объектов.

На основе созданных средств редактирования разрабатываются варианты цифровых архивов документов, предназначенных для решения задач формирования документации по учебным курсам. Использование форматов данных LOD, средств обработки HTML веб-браузера и разработанных технологий позволяет решать широкий класс задач формирования документации, начиная от формирования содержательной части текстов документов, заканчивая представлением стилевых характеристик текстов, а также интегрирования логической разметки в глобальные сервисы доступа к данным.

Данная работа продолжает исследования, обозначенные в [2] в направлении разработки программного инструментария сервисов хранения, обработки и обеспечения доступа к семантической информации.

Технологии LOD

Технологии LOD базируются на представлении публикуемой информации в виде семантически размеченного документа, веб-страницы, результата запроса к базе данных (контента). При использовании HTML для представления контента в роли основного формата разметки выступает RDFa (Resource Description Framework in Attributes). Разметка представляет собой граф (подграф), формируемый отношениями (тройками) вида <Субъект, отношение, объект>, нанесенный на древовидную структуру HTML при помощи нескольких специальных атрибутов. Большинство таких атрибутов игнорируются веб-браузером и не влияют на отображение документа.

Отдельно от визуально контента для представления LOD используются стандартные форматы семантического веба RDF, OWL, TTL, N3 и 4, JSON-LD, RDF/JSON и др. Данные представления – это расширения стандартных форматов вида текст, XML и JSON. Существует несколько бинарных форматов, ориентированных на компактное хранение графов в файловой системе.

Представление графа в оперативной памяти вычислительного устройства в большинстве библиотек и программных каркасов (фреймворков) ориентировано на быстрое выполнение операций перебора и фильтрации троек по одному или нескольким параметрам, а также на поддержку запросов стандарта SPARQL. Некоторые хранилища дополнительно поддерживают интерпретацию отношений rdf:subClassOf, rdf:subPropertOf и производных от rdfs:label. Это позволяет создавать средства визуализации содержимого графа, например, строить интерфейсы администратора хранилищ.

Разработчиками LOD создан ряд ресурсов для описания информации, в том числе набор онтологий базового уровня для описания различных аспектов контента (scos, rdfs, XMLSchema, dc, prov, oa, schema.org и др.), графы, представляющие энциклопедические данные, например, Dbedia [3], Wikidata [4], Goole Knowledge Graph, сервисы автоматизации частичной семантической разметки текста, например, Dbpedia Spotlight [5], средств распределенного исполнения SPARQL-запросов. Созданные ресурсы позволяют разрабатывать механизмы интеграции данных графов, распределенных в сети интернет, в единую информационную среду.

Средства цифрового архива

Инструментарий разработки цифровых архивов, поддерживающий технологии LOD предназначен для создания средств публикации накопленной в учреждении информации в виде, позволяющем ее интегрирование в единую информационную среду семантического веба, а также средств поддержки подготовки и публикации семантически размеченных документов, разработке веб-приложений публикации и обработке данных научных исследований.

Инструментарий цифрового архива строится на компонентной архитектуре. В качестве инструмента реализации взята компонентная архитектура Zope (Zope Component Architecture, ZCA) [6], созданная для среды программирования Python [7]. Основным форматом внутреннего представления данных LOD использован граф библиотеки rdflib. Для передачи данных во внешние процессы граф (подграф) сериализуется в один из подходящих текстовых или двоичных форматов.

Инструментарий содержит следующие основные компоненты:

* хранилища контента и метаданных с доступом SPARQL, полнотекстовым поиском и возможностью обработки LOD, основанной на формализованных знаниях;
* сервис анализа хранимого контента и комплектации архива семантической информацией, описывающей контент;
* средства разработки приложений LOD и их пользовательских интерфейсов.

Хранение документов LOD

Документы в цифровом архиве представляются а) своим содержимым, которое храниться либо в базе данных библиотеки kyotocabinet, либо в виде файла в файловой системе. Метаинформация, полученная обработкой и анализом содержимого, представляет собой б) аннотацию содержимого, описывается согласно LOD-стандарту oa и в) храниться в виде графов.

Каждая аннотация (субъект) представляет собой дерево с как минимум двумя узлами, формируемыми отношениями oa:hasTarget, указывающей на исходный контент, и oa:hasBody, указывающей на аннотацию. В качестве аннотации выступает текстовое содержимое документа, если его удалось изъять, а также структура типа Dublin Core (dc). Содержимое (байты, текст), хранимое в kyotocabinet идентифицируется его 128-битным murmur3-хэшем, что позволяет создавать отображения ключ (хэш) на содержимое и обратно. Хэш является идентификатором содержимого в LOD. Использованная система отображения позволяет достаточно просто и вычислительно эффективно реализовывать отображения при достаточно малой вероятности наложения хэш-значений.

Рассмотрим основные свойства использованных для описания унаследованного контента онтологий.

**Open Annotation** (oa). Стандарт этой отологии принят в 2017 году, ее основная задача представлять содержимое (аннотацию), описывающее другое содержимое. Закладка браузера является примером такого содержимого.

**Friend-of-a-friend** (foaf) позволяет представлять информацию об агентах: физических и юридических лицах, а также программных агентов.

**Provenence** (prov). Онтология prov – основа описания информационных потоков в документах и их взаимосвязи. В цифровом архиве prov используется для ассоциации документа с цифровым архивом и организацией владельцем.

**Dublin Core** (dc) представляет в аннотации ментаинформацию о творческом произведении: авторов, формат содержимого, его описание и др.

**DBPedia resource** (dbr) – пространство имен объектов (ресурсов) Wikipedia. Онтология dbr используется для обозначения конкретных сущностей, географических названий и т.п.

**Schema.org** (schema) представляет объекты, распознаваемые поисковыми агентами Google, Yandex, Yahoo и др.

Кроме перечисленных онтологий используются стандартные онтологии (RDF, RDFs, RDFa, XSD) и онтологии из проекта NEPOMUK (<https://userbase.kde.org/Nepomuk>), предназначенные для описания объектов, хранимых в полнотекстовых индексных цифровых архивах.

Компонента анализа контента

Метаинформация о контенте комплектуется в результате анализа так называемыми экстракторами. Экстракторы анализируют содержимое и пытаются получить информацию о текстовом содержимом, авторе, времени создания контента и др. Экстраторы реализованы на основе утилит file, extract (libextract), tracker и recoll. Все эти утилиты относятся к свободному программному обеспечению и интегрированы с компонентой как внешние процессы.

Результаты анализа экстракторов представляются в виде подграфа oa, связываются логически с исходным содержимым и передаются в хранилища метаданных LOD.

Хранилище метаданных

Реализация стандарта хранения графа LOD выполнена с использованием программной системы ClioPatria [8]. Система ClioPatria реализована в среде программирования SWI-Prolog. Проект находится в стадии активной разработки и нацелен на создание сервиса хранения графов LOD, реализующих как все стандарты хранения и доступа, так и тесную интеграцию с Prolog.

Системный уровень хранилища реализован на языке C. Хранилище использует компактные структуры данных для представления троек в оперативной памяти, а также на диске. Особенностью хранилища является тот факт, что все тройки хранятся непосредственно в оперативной памяти, т.к. у хранилища нет специальных допущений о структуре возможных запросов. Таким образом, оперативная память сервера – это критический ресурс хранилища.

Разработчиками ClioPatria создан инструментарий доступа к хранилищу несколькими способами: осуществление SPARQL-запросов по стандартному протоколу, а также запуск запросов к Prolog-системе при помощи протокола Pengines [9]. Для протокола Pengines создам ряд реализаций в виде классов языков программирования JavsScript, Java, Python. При помощи этих классов разработчики приложений и программных агентов могут взаимодействовать с машиной логического вывода севера ClioPatria.

Средства логического вывода

Наличие в ClioPatria встроенной системы программирования SWI-Prolog [10] и средств интеграции с RDFa отрывает широкие возможности по обработке размеченной цифровой информации, хранимой в архиве, на основе формализованных знаний. Пролог позволяет осуществлять запросы в форматах Datalog и SPARQL, а также строить систему знаний для обработки полученных данных.

В качестве эксперимента средствами Пролога в очень короткий срок удалось провести интеграцию известного примера естественно-языкового интерфейса к базе данных по географии США, поставляемого совместно с системой программирования TurboProlog-2.0 (geobase), с встроенной в SWI-Prolog системой поддержки RDFa. В результате разработан прототип аналогичного интерфейса к базе данных активных разломов Сибири, разработанной в Институте земной коры СО РАН.

Другим применением логического вывода выступает поддержка разработки интегрированных рекомендательных систем [11,12], основанных на онтологическом описании набора признаков рекомендуемого объекта и характеристик пользователя. Основной проблемой, решаемой в таких системах, является обобщение хранимой информации, что позволяет, как расширять набор получаемых рекомендаций за счет смежных терминов в одной общей категории, так и сужать объем выдачи при помощи замены общих терминов узкоспециализированными, релевантными поисковому запросу.

Полнотекстовый поиск

Система ClioPatria обладает некоторыми средствами поиска текстовой информации по ключевым словам, однако, в архиве целесообразно использовать более мощные системы полнотекстового индекса, например, Elasticsearch [13]. Реализация и интеграция этого сервиса достаточно проста, так как любой RDF-граф представим в формате JSON (JSON-LD), при этом JSON – это основной формат хранения индексируемой информации в системе ElasticSearch, необходимо только на сервере выделить особым образом тройки, которые отвечают за представление ресурса пользователю в результатах поиска. Elasticsearch обладает средствами нечеткого сравнения термов, что позволяет развивать систему в направлении реализации систем поиска релевантной информации. Основные аспекты функционирования модулей рассматривается далее по тексту статьи в разделе приложений.

Приложения технологий

Разработанные средства использованы в решении ряда прикладных задач. Приведем один пример использования средств LOD и компонент архива.

Министерство образования и науки Российской Федерации после ряда экспериментов перешло на широкомасштабное внедрение Болонского процесса в образовательную среду Российской Федерации. Одной из задач, решаемых в рамках этого внедрения – переход к компетентностно-ориентированному представлению требований к педагогическому процессу. Проводимая реформа затрагивает все аспекты процесса, включая систему классификации специальностей (внедрение программ м направлений, специализация по уровням квалификации (бакалавриат, магистратура и др.), введение прикладного бакалавриата, перечень преподаваемых курсов, целей и задач курсов (согласование с компетенциями, заданными в Федеральном государственном стандарте), формы ведения занятий в вузе (введение интерактивных форм обучения), распределение лекционных и практических занятий и др. Существующая документация курса дополнена новыми формами обязательных документов – фондом оценочных средств (ФОС), аннотациями курсов. В дополнение к этому, руководство вуза с целью повышения качества предоставляемых образовательных услуг вводит собственные дополнения к форме, содержанию и требованиям к оформлению документации, в частности, к качеству конвертации в HTML для публикации на сайте вуза.

Для минимального исполнения требований руководства вузов для каждого курса преподавателю требуется оформить как минимум три документа – рабочую программу курса, аннотацию к ней, а также ФОС. Этот набор составляется для каждой возможной комбинации: вуз, кафедра, специальность, направление (профиль), программа, квалификация (бакалавр, специалист, магистрант, аспирант), академический или прикладной вариант квалификации, форма обучения (очная, заочная, вечерняя, очно-заочная и др.). Каждая комбинация отражается в учебном плане вуза, данные из которого ежегодно или два раза в год должны уточняться в рабочих программах. Последние пять лет показали уровень продуманности принимаемых менеджерами министерства решений в части требований к представлению курса – разработано четыре поколения стандартов (1,2,3 и 3+) –, для которых требовалось представить документы в соответствующей обновленной форме. На этом фоне задача разработки нового учебного пособия теперь кажется более простой, чем раньше. Опыт показывает, что преподаватели в большей части не способны справиться с качественным оформлением документов в предоставленные им временные ограничения, что приводит к необходимости руководству кафедр нанимать секретаря-специалиста, функцией которого является доведение предоставленной документации до необходимого уровня качества.

Решение данной проблемы видится в разработке программной системы, позволяющей собирать тексты рабочих программ, аннотаций и ФОС из отдельных частей: перечня компетенций и распределения нагрузки (учебный план); содержания (преподаваемых тем модуля/курса) и ФОСа, тексты которых разделяется между разными версиями документов. Титульные листы формируются также из данных учебного плана и подготовленных шаблонов. Средств Microsoft Word и Excel, стандартно используемых для решения этой задачи, со встроенным VBA явно недостаточно.

Рассмотрим схему представления размеченной рабочей программы в разработанной системе. Формат представления базируется на результатах из [14,15].

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
 <html lang="ru" xmlns=<http://www.w3.org/1999/xhtml>  
**xmlns:taa=**[**http://irnok.net/engine/rdfa-manipulation**](http://irnok.net/engine/rdfa-manipulation) xml:lang="ru" metal:define-macro="page">  
 <head> <!-- Подключение стилевых таблиц и модулей -->  
 </head>  
 <body prefix="rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns# ... foaf: <http://xmlns.com/foaf/0.1/> imei: imei.html#   
course: https://irnok.net/college/plan/01.03.02-16-1234-2461\_1%D0%BA\_PB-SM.plm.xml.xlsx-%D0%911.%D0%92.%D0%94%D0%92.3.1.html#"  
 resource="#post" typeof="schema:CreativeWork sioc:Post prov:Entity">  
<!-- Панель управления приложением -->  
 <main lang="ru" **resource="#annotation**" **typeof="oa:Annotation"** id="main-document-container"><div **property="oa:hasTarget"   
 resource="#course-work-program"**></div>  
 <article **property="oa:hasBody"**   
typeof="**schema:Article foaf:Document curr:WorkingProgram**"  
 resource="**#course-work-program**" id="main-document">  
 <div **taa:content="imei:title-page"**></div> <!--Титульный лист.. -->  
 <div **taa:content="imei:neg-UMK"**></div> <!--Лист согласования..-->  
 <section id="contents" class="break-after"> <h2 class="nocount c">Содержание</h2>  
 <div id="tableOfContents"></div>  
 </section>  
 <section id="course-description" resource="#description"  
**property="schema:hasPart" typeof="schema:CreativeWork"**>  
 <div property="schema:hasPart" resource="#purpose"  
**typeof="dc:Text cnt:ContentAsText"** >  
 <div **property="cnt:chars" datatype="xsd:string"**>  
 <h2 **property="dc:title" datatype="xsd:string"**>Цели и задачи дисциплины (модуля)</h2>  
 <p>Целю преподавания дисциплины "Технологии программирования" является освоение ...</p>  
 </div>  
 </div>  
. . . . . . . .  
 <div property="schema:hasPart" typeof="dc:Text  
cnt:ContentAsText" resource="#volume">  
 <div property="cnt:chars" datatype="xsd:string">  
 <h2 property="dc:title" datatype="xsd:string"> Объем дисциплины (модуля) и виды учебной работы (разделяется по формам обучения)</h2>  
 <div **taa:content="course:time-distrib"**></div>  
 </div>  
 </div>  
. . . . . . . .

В приведенном примере ключевые структуры выделены жирным шрифтом, прокомментируем эти структуры.

Страница, отображаемая пользователю, – это аннотация документа "#annotation", причем и аннотируемое содержимое, и текст аннотации – это одно и то же на этапе формирования документа: ресурс "#course-work-program". В LOD все ресурсы являются глобальными. В данном случае это достигается подстановкой пространства имен по умолчанию, полного URI-адреса текущей страницы, с левой стороны от названия ресурса.

Титульный лист и лист согласования вставляются со страницы шаблонов “imei.html” (сведения об Институте математики, экономики и информатики Иркутского государственного университета). Данные курса и название специальности вставляются в шаблон из контекста документа. Все ключевые статические шаблоны курсов можно поместить на одну страницу шаблонов.

Текст разбивается на разделы, обрамляется тегами <div> и <span> со соответствующими структурами RDFa. Анализ опыта разработчиков LOD ресурсов показал, что для формирования отношения достаточно использовать RDFa-теги property, typeof и datatype. От использования rel и about следует отказаться. Это делает структуру семантической разметки более строгой за счет уменьшения количества сущностей.

Команда taa:content добавляет в документ текст другой страницы, адрес которой формируется интерпретацией параметра, например, taa:content="course:time-distrib" включает в текст страницы сгенерированную таблицу распределения часовой нагрузки между видами занятия (лекции, практики, лабораторные, СРС, и т.д.). Для генерации таких таблиц разработан сервер веб-страниц данных курсов. Каждая страница сервера представляет какой-либо курс/модуль, представленный в учебном плане. Содержание страницы кодируется структурой ее URL и сверстывается в момент первого обращения. Шаблоны текста обозначаются атрибутом id.

В режиме редактирования рабочей программы (документа) все теги, имеющие комбинацию атрибутов property="cnt:chars" и datatype="xsd:string", преобразуются в редактируемый текст. Для ясности пользователю представляется документ без taa:-включений с указанием этих включений при помощи специальных тегов и стилей. В момент завершения редактирования текст сохраняется в файловой системе сервера. Далее пользователь может выполнить фиксацию (commit) изменений и синхронизацию текста сохраненного документа с сервером системы контроля версий текста.

В другом приложении реализована задача автоматизации формирования документов с согласованием грамматики при подстановке фраз в текст. Рассмотрим пример текста доверенности.

<p>Я, <span property="fibol:designatesSignatory bibo:owner"  
 typeof="fibol:Signatory foaf:Person dbr:Principal"  
 resource="#principal"><span property="foaf:name" id="signatory-name" datatype="xsd:string" class="edit">Иванова Елена Викторовна</span>,<span property="adoc:hasPassport" resource="#signatory-passport" typeof="acrt:Certification"> <span property="acrt:qualification" resource="dbr:Passport">паспорт</span>…

Данный пример выражает факт, что доверитель (fibol:designatesSignatory) удостоверяется (adoc:hasPassport) при помощи паспорта (acrt:qualification dbr:Passport).

В режиме редактирования документа в виде формы теги, содержащие набор атрибутов datatype="xsd:string", class="edit", преобразуются в поле ввода. Значение этого поля распространяется далее по тексту документа. Согласование грамматики реализовано при помощи атрибутов id, data- и class. Форма class=“disp” задает возможность подстановки фразы, идентифицированной id, в текст данного тега. Атрибуты data-m и data-f задают вариант слов, соответствующий роду соответствующего существительного. Атрибут data-case задает склонение (падеж и число), в которое надо поставить слова текста. Алгоритмы грамматических преобразований реализованы на сервере. Продолжим предыдущий пример: рассмотрим с грамматические конструкции.

...проживающ<span class="disp" id="signatory-name" data-m="ий" data-f="ая">\_\_\_</span> по адресу:...Подпись <span class="disp string" id="signatory-name" data-case="gent">\_\_\_</span> удостоверяю...

Проект доступен в виде модулей языка программирования Python по адресу: https://github.com/isu-enterprise.

Заключение

В статье обсуждается вопрос применения технологий Linked Open Data (LOD, открытых связанных данных) для решения задач создания цифровых архивов документов с возможностью создания новых документов на основе хранимых данных. Верстка документа осуществляется при помощи алгоритмов, интерпретирующих отношения между элементами дерева HTML. Алгоритмы верстки запускаются при наличии соответствующих условий в узлах дерева документа. Каждый алгоритм вносит в дерево изменения, формирующий окончательное содержание и вид документа. Создаваемые документы хранятся на серверах, поддерживающих протокол HTTP.

Приведен пример использования разработанных средств подготовки документов в образовательной среде и юридических документов. Дальнейшее развитие данного проекта осуществляется по нескольким направлениям: а) совершенствование средств верстки документов, б) автоматизация разметки на основе анализа изменения документов [2], в) реализация практических приложений, г) сбор дополнительной информации о потребностях пользователей, д) разработки средств обеспечения регламентированного доступа к информации. Созданные программные средства и технологии направлены на разработку программной среды глобального электронного документооборота, позволяющего физическим лицам и организациям обмениваться документами, логическая структура которых трудноформализуема в рамках традиционных средств автоматизации документооборота.

Благодарности

Результаты получены при частичной поддержке Совета по грантам Президента Российской Федерации, государственной поддержка ведущих научных школ Российской Федерации (НШ-8081.2016.9).

Результаты получены при активном использовании сетевой инфраструктуры Телекоммуникационного центра коллективного пользования «Интегрированная информационно-вычислительная сеть Иркутского научно-образовательного комплекса» (ЦКП ИИВС ИРНОК) (<http://net.icc.ru>).

БИБЛИОГРАФИЧЕСКИЙ СПСОК

1. Bizer Ch., Heath T., Berners-Lee T. Linked Data – The Story So Far //International Journal on Semantic Web and Information Systems. 2009. Vol. 5 (3). P. 1–22.

doi:10.4018/jswis.2009081901.

1. Черкашин Е.А., Белых П.В. и др. Подход к управлению содержанием сайта на основе технологий RDF // Материалы Всероссийской конференции с международным участием «Знания – Онтологии – Теории» (ЗОНТ-2013), 8 – 10 октября. 2013. Т. 2. Новосибирск. Издательство «РИЦ Прайс-курьер».

ISSN 0568-661X

1. Lehmann J., Isele R., Jakob M., Jentzsch A., Kontokostas D., et al. DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia // Semantic Web Journal. 2015. Vol. 6, No. 2, P. 167-195, IOS Press.
2. Krötzsch M. How to use Wikidata: Things to make and do with 40 million statements // In Keynote at the 10th Wikimania Conference. 2014.
3. Daiber J., Jakob M., Mendes P. Improving Efficiency and Accuracy in Multilingual Entity Extraction // Proceedings of the 9th International Conference on Semantic Systems (I-Semantics), 2013.

URL: <http://korrekt.org/talks/2014/wikimania-wikidata.svg> (access-date: 01.07.2017)

1. Baiju M. A Comprehensive Guide to Zope Component Architecture. [Электронный ресурс]

URL: <http://muthukadan.net/docs/zca.html> (access date - 01.07.2017)

1. Langtangen H. A Primer on Scientific Programming with Python (Texts in Computational Science and Engineering) 3rd ed. Springer. 2012. 798 P.
2. Wielemaker J., Beek W., Hildebrand M., Ossenbruggen J. ClioPatria: A SWI-Prolog infrastructure for the Semantic Web //  [Semantic Web](https://www.researchgate.net/journal/1570-0844_Semantic_Web). 2016. Vol. 7(5). P. 529-541, DOI: 10.3233/SW-150191
3. Lager T., Wielemaker J. Pengines: Web Logic Programming Made Easy // Theory and Practice of Logic Programming. 2014. Vol. 14(4-5),

DOI: 10.1017/S1471068414000192

1. Wielemaker J., Schreiber G., Wielinga B. Prolog-Based Infrastructure for RDF: Scalability and Performance // In: Fensel D., Sycara K., Mylopoulos J. (eds) The Semantic Web - ISWC 2003. ISWC 2003. Lecture Notes in Computer Science. 2003. Vol. 2870. Springer, Berlin, Heidelberg.
2. Нефедова Ю. С. Архитектура гибридной рекомендательной системы GEFEST (Generation–Expansion–Filtering–Sorting–Truncation) // Системы и средства информатики. 2012, Т.22, вып.2, с.176–196.
3. Beel J., Gripp B., Langer S., Breitinger C. Research-paper recommender systems: a literature survey // International Journal on Digital Libraries. 2016. Vol. 17. P. 305. doi:10.1007/s00799-015-0156-0. (access date - 12.12.2016)
4. Kuć R., Rogoziński M. Mastering Elasticsearch - Second Edition. Packt Publishing. 2015. 372 p. ISBN-9781783553792
5. Capadisli S., Guy A., Verborgh R., Lange C., Auer S., Berners-Lee T. Decentralised Authoring, Annotations and Notifications for a Read-Write Web with dokieli// Procs of ICWE international conference, 5-8 June, 2017, Rome, Italy. (to appear) [Электронный ресурс] Preprint URL:<http://csarven.ca/dokieli-rww>
6. Heino N, Tramp S, Auer S, et al. Managing Web Content using Linked Data Principles – Combining semantic structure with dynamic content syndication // Computer Software and Applications Conference (COMPSAC), IEEE 35th Annual. 2011. P. 245-250. [Электронный ресурс] URL:<http://svn.aksw.org/papers/2011/COMPSAC_lod2.eu/public.pdf> (access date - 30.05.2013).