Знания-Онтологии-Теории (ЗОНТ-17)

Использование технологий Linked Open Data при подготовке и публикации текстовых документов

Черкашин Е.А.[1,2,3], Шигаров А.О.[1,3], Орлова И.В.[2], Михайлов И.С.[4]

[1]Институт динамики систем и теории управления СО РАН им. В.М. Матросова, ул. Лермонтова, д. 134, г. Иркутск, 664033, Россия
[2]Иркутский национальный исследовательский технический, ул. Лермонтова, д. 83, г. Иркутск, 664074, Россия
[3]Иркутский научный центр СО РАН, ул. Лермонтова, д. 134, г. Иркутск, 664033, Россия
[4]Лимнологический институт СО РАН, ул. Улан-Баторская, д.3, г. Иркутск, 664033, Россия

eugeneai@icc.ru, shigarov@icc.ru

Аннотация. *Рассматривается проблематика автоматизации создания текстовых документов при активном использовании декларативных средств и технологий открытых связных данных. Разработан программный инструментарий, позволяющий сверстывать HTML5-документы из различных источников: веб-страниц и результатов обработки текста на стороне клиента (веб-браузера), а также представлять документ в заданном виде, например, как шаблон или форму. Технологии базируются на активном использовании RDFa-разметки. Рассмотрено несколько примеров приложения разработанных программ.*

Ключевые слова: открытые связанные данные, автоматизация верстки документа, семантическая разметка документа

1 Введение

Технология Linked Open Data (LOD, открытые связанные данные) [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 (URI) ресурсов. Дескриптивные возможности технологий семантического веба, средства публикации документов HTML5, а также технологии LOD создают инфраструктурный базис средств создания (верстки) и публикации документов. При этом документ формируется из отдельных частей (текстов и изображений), загружаемых с других серверов при помощи ссылок на соответствующие ресурсы. Ресурсы представляю собой как статическое содержимое, так и результаты запуска алгоритмов конвертации данных, порождающими текстовое содержимое. Средства LOD обеспечивают информации, представленной в документе, логическую разметку, информативный базис для различных вариантов визуального представления и интерпретации, установку логических связей с другими документами, экспорт информации в другие документы, процедурную обработку и т.д. Одним из важных 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.

В работе рассматривается подход к построению программного обеспечения, реализующего функции подготовки и верстки документов, размеченных согласно принципам LOD. Документ создается при помощи набора скриптов JavaScript, функционирующего на стороне веб-браузера (клиентский JavaScript). Фрагменты текста загружаются из других документов или же порождаются серверными и клиентскими скриптами. Результирующий документ помещается в хранилище данных, распечатывается или загружается на рабочую станцию пользователя в виде HTML-файла. Скрипты JavaScript реализуют различные представления документа (View), например, в виде формы ввода; производить преобразования частей текста, например, склонять существительные и подставлять местоимения, генерировать содержание и т.п. На основе созданных средств редактирования разрабатываются варианты цифровых архивов документов, предназначенных для решения задач формирования документации по учебным курсам, оформление результатов исследований высокопроизводительно секвенирования ДНК/РНК. 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.

Развитие средств представления документов на основе LOD направлено на формирование прототипа глобальной среды, поддерживающей автоматизацию передачи информации между документами нескольких предприятий, и создание, в идеале, распределенного документооборота по принципам социальной сети. Задача актуальна, прежде всего, в среде малого бизнеса, где нет устоявшихся информационных потоков и связей между документами, что, в основном, обусловлено динамичностью самого бизнеса. Использование средств крупномасштабной автоматизации документооборота в такой среде связано с большими материальными затратами на приобретение и/или постоянное усовершенствование программного продукта. Более того, на предприятиях, где функционируют такие среды автоматизации, практически всегда существует большой класс документов, не интегрированных в основной документооборот (например, докладные записки, заявления), документов с трудноформализуемой информацией (например, должностные инструкции), а также документов, получаемых организацией извне и не содержащих логическую разметку (например, стандарты отрасли).

Данная работа продолжает исследования, обозначенные в [2] в части реализации клиентских сервисов, функционирующих на вычислительной платформе веб-браузера.

2 Технологии верстки документов на основе LOD

Разрабатываемые программные средства верстки текста документов базируются на технологиях, реализующих стандарт HTML5, существующих в современных веб-браузерах. Документ – это веб-страница, которая создается из отдельных частей при помощи алгоритмов, реализованных в системе программирования клиентского JavaScript. Для внесения изменений в содержимое страницы использованы встроенные средства редактирования веб-страниц браузеров (атрибут `contentEditable`), управляемые специальными библиотеками (например, Medium Editor, https://yabwe.github.io/medium-editor/), позволяющими расширять базовые функции встроенного редактора функциями преобразования структуры документа и его LOD-разметки.

Текст документа размечен при помощи стандарта RDFa. Семантическая разметка представляет логическую структуру документа, предметные отношения между сущностями, представленными в документе, например, выделяются стороны, заключающие некоторое соглашение, предмет соглашения; формальные данные; отношения “часть-целое”, при помощи которых осуществляется секционирование текста документа. В дополнение к семантической разметке в текст документа встраиваются командные структуры, позволяющие активировать алгоритмы преобразования содержимого, например, инструкции включения частей одних документов в другие, грамматические преобразования фраз. Командные структуры распознаются проще, чем структуры семантической разметки. Обработка командных структур направлена на преимущественно алгоритмические преобразования текста.

В качестве основы семантической разметки документа использованы результаты открытого проекта Dokieli [3]. Проект ориентирован на создание средств редактирования текстов HTML5, представляющих научные публикации, несколькими соавторами. Dokieli позволяют размечать текст тегами RDFa, копировать документ в собственное хранилище и производить аннотацию элементов текста комментариями читателей и соавторов. В разметке документов используются следующие онтологии.

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. Ресурс `oa` (закладка) находится в двух основных отношениях с другими ресурсами: `oa:hasTarget`, `oa:hasBody`. Первое отношение ссылается на аннотируемое содержимое через URI, а второе, которое в общем случае является множественным отношением, - на ресурс-аннотацию, например, на текст-описание ресурса или комментарий, связанный с ресурсом. Редактируемый документ представляет собой аннотацию порождаемого документа, т.е. сначала он сам себя аннотирует, а после завершения подготовки окончательной версии документа – на его неизменяемую копию, например, на свой PDF.

Friend-of-a-friend (`foaf`) позволяет представлять информацию об агентах: физических и юридических лицах, а также программных агентов. Данная онтология широко применяется для представления связей между лицами (агентами) в социальной сети.

Provenence (`prov`) обеспечивает словарь терминов и отношений, характеризующих происхождение информации. Например, при помощи данной онтологии удобно ссылаться на ресурсы (документы), из которых была получена информация, использованная в документе. Онтология `prov` – основа описания информационных потоков в документах и их взаимосвязи.

Dublin Core (`dc`) используется для обозначения элементов редактируемой аннотации, например, заголовков разделов, заголовка документа, типов документов, различных описаний, если они содержатся в документе. В отличие от `oa` эта онтология представляет, в основном, ментаинформацию о творческом произведении.

DBPedia resource (`dbr`) – пространство имен объектов (ресурсов) Wikipedia. При помощи онтологии `dbr` удобно ссылаться на конкретные экземпляры объектов и конкретные классы, например, паспорт физического лица (`dbr:Passport`), конкретный город и т.п.

Schema.org (`schema`) представляет объекты, распознаваемые поисковыми агентами Google, Yandex, Yahoo и др. Эта онтология используется в случае, если в перечисленных выше и специализированных по предметным областям документа онтологиях не находится подходящего отношения. В этой онтологии есть полезные отношения типа “часть-целое” (`schema:hasPart`), “результат творческой деятельности” (`schema:CreativeWork`), при помощи них описываются разделы документа и строится их иерархическая структура.

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. Они позволяют более детально описывать метаинформацию о творческом произведении типа файл, изображение, документ, раздел документа, а также задавать тип документа, текстовое содержимое, идентификацию произведений и т.д. На трудоемком этапе поиска онтологий для представления документов по предметным областям оказался весьма полезным ресурс Linked Open Vocabularies (`http://lov.okfn.org/dataset/lov/`), а также помощь сообщества исследователей, поддерживающих этот ресурс.

Архитектура программной системы представлена на Рис. 1 (все взаимодействия, изображенные на рисунке, двунаправленные). Отображаемая пользователю страница загружается с сервера, где она конструируется из содержимого, хранимого в базе данных или файловой системе, результатов преобразований данных других источников алгоритмами конвертации, и элементов шаблона, который включает в отображаемую страницу интерфейс пользователя для управления представлениями и модулями интерпретации. В процессе формирования окончательного вида документа модули интерпретации запрашивают необходимые данные и ресурсы на исходном сервере или других интернет-серверах.

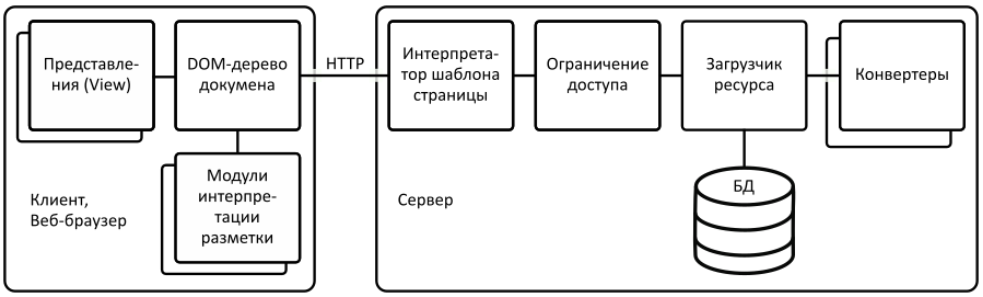

Рис. 1. Общая архитектура системы верстки документов

Модули интерпретации семантической разметки реализованы в клиентском JavaScript. Они запускаются, как только основное содержимое загружается в браузер. Модули сканируют структуру дерева документа, распознавая условия их активации. Если поиск оказался успешным, запускается тело модуля, которое в общем случае вносит изменения в документ. Формирование документа заканчивается, как только все условия не были активированы, и не осталось необработанных команд.

База данных системы представляет собой набор средств хранения и обеспечения доступа к разноформатным данным. Основной текст документа, загружаемый в первую очередь, храниться в виде файлов XML в файловой системе на сервере. Над файловой системой надстроен уровень, позволяющий отслеживать версии документов и развитие документа во времени, создавать интегральные резервные копии. Такой уровень достаточно легко реализуется при помощи современных систем контроля версий, например, GIT или Subversion. Формат XML позволяет представлять логическую разметку основной части документа, а также хранить ее в самом документе.

Для решения задач обеспечения стандартного для хранилищ данных сервиса доступа SPARQL и полнотекстового поиска разработаны компоненты хранения логического слоя и сопутствующих данных в виде графов троек. В качестве таких компонент выступают системы ClioPatria [4] или Jena. Первая система интересна тем, что полностью реализована на языках Prolog и C, она предоставляет несколько форматов компактного хранения данных, а также тесную интеграцию данных троек со средой программирования языка Prolog. К программам (предикатам) обеспечивается регламентированный доступ при помощи протокола Pengines [5], для которого реализованы библиотеки для популярных языков программирования, включая клиентский JavaScript, Java, Python. Если по каким-либо причинам не удается реализовать сервис хранения троек средствами ClioPatria, то можно использовать сервисы на основе Java-библиотеки Jena.

Полнотекстовый индекс обеспечивается системой Elasticsearch. Реализация этого сервиса достаточно проста, так как любой RDF-граф представим в формате JSON (JSON-LD). JSON – 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 работы в разделе приложений.

3 Программные системы - аналоги

Среди существующих проектов управления семантически-размеченным содержимым, выделяется проект Semantic MediaWiki, где реализуется подход, базирующийся на обычном механизме Wiki для представления основного текстового содержимого сайта. В текст добавляются аннотации при помощи специальной разметки. Семантические аннотации позволяют создавать механизмы поиска страниц Wiki, учитывающие семантику размеченного текста [6].

В отличие от Semantic MediaWiki в проекте OntoWiki аналогичные результаты получены, начиная с другой отправной точки. OntoWiki базируется на преимущественном использовании логического представления информации в виде семантической сети. Эта логическая структура редактируется при

помощи генерируемых автоматически форм ввода по известным терминам в словаре. Пользователю доступно изменение лишь одного текстового свойства LOD `lod:content`, которое, в общем случае, содержит текст с разметкой HTML. Разметка HTML не привязывается к логической структуре отображаемого объекта (субъекта). Текст можно редактировать при помощи встроенного в OntoWiki WYSIWYG-редактора. Проект OntoWiki направлен на поддержку технологий социальных сетей на базе технологий Linked Data [6].

Как было сказано выше, использованный формат разметки документов является развитием идей проекта Dokieli (https://github.com/linkeddata/dokieli) [2]. Проект представляет собой WYSIWIG-редактор размеченных LOD HTML-страниц, стилей (CSS) отображения и встроенного подграфа (его содержимое не видно в документе), представленного в различных форматах (TTL, N3, JSON-LD, TriG). Редактор позволяет добавлять в текст новую RDFa-разметку, при этом эта интересующая нас функция реализована в самом общем виде. Кроме того, средствами клиентского JavaScript реализованы функции авторизации пользователя согласно стандарта WebID, аннотацию текста другими пользователями в формате Open Annotation в виде комментариев к тексту и закладок. Механизм аннотаций интегрирован с сервисом gitter.im, который показывает комментарии читателей в виде диалога. Вся информация сохраняется либо на серверах, поддерживающих протокол Solid (https://solid.mit.edu/), либо локально в базе данных браузера. Редактируемый документ очень просто может быть перемещен (скопирован) из одного solid-хранилища в другое. Получаемые тексты можно загружать на рабочую станцию пользователя. Таким образом, dokieli представляет собой систему поддержки распределенного редактирования текстовых документов, который в полной мере реализует взаимодействие с другими ресурсами LOD, и все функции взаимодействия реализованы исключительно при помощи подпрограмм клиентского JavaScript.

Существует также огромный класс редакторов документов, ориентированных на создание научных публикаций (см., например, http://substance.io/). В данном исследовании мы стремимся развить средства автоматизации разметки документа на основе анализа изменений документа, предложенной в [2], и конкретно в этой работе преследуется цель реализации одной из инфраструктурных задач – разработки средств создания документов на платформе веб-браузера.

4 Приложения созданных технологий

Разработанные средства верстки документов использованы в решении ряда прикладных задач.

4.1 Тексты учебных программ вузов

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.

```
<!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"

      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-differ"></div>
                            </div>
                        </div>
                        . . . . . .
```

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

1. 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.
2. 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.
3. Текст разбивается на разделы, обрамляется тегами и 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.
4. Команда taa:content добавляет в документ текст другой страницы, адрес которой формируется интерпретацией параметра, например, taa:content="course:time-differ" 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.

Исходный код проекта доступен в виде модулей языка программирования Python по следующему адресу: https://github.com/isu-enterprise. Основной модуль – isu.college.

4.2 Подготовка юридических документов

Студентами Института математики, экономики и информатики Иркутского государственного университета разработано приложение для создания юридических документов – нестандартного вида постановлений судебных исполнителей и нотариальных документов. Разметка документов в этом проекте аналогична разметке, используемой в предыдущем примере, добавлено несколько онтологий для представления юридически-значимых терминов: `fibo` (The Financial Industry Business Ontology) и формальных спецификаций удостоверяющих документов `acrt` (A Certification Ontology).

Онтологии `fibo` позволяют задавать роли лиц в документе, например в генеральной доверенности – кто является доверителем, а кто доверенным:

```
<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 приложении добавлен режим редактирования документа в виде формы. Модуль JavaScript ищет теги содержащие набор атрибутов `datatype="xsd:string" class="edit"` и преобразует текст тега в поле ввода, при этом основной текст недоступен для редактирования. Основную грамматическую форму предложений текста в соответствии с форматом RDFa можно записать в атрибут `content`, и, например, хранить имена людей в именительном падеже, а в отображаемый текст вставлять подходящее склонение.

Согласование грамматики реализовано при помощи атрибутов `id`, `data-` и `class`. Значение атрибута `class`, равное “`disp`” говорит о возможности подстановки отредактированной в режиме формы строки, идентифицированной `id`, в текст тега. Атрибуты `data-m` и `data-f` задают вариант слов, соответствующий роду основного существительного в тексте. Атрибут `data-case` задает склонение (падеж и число), в которое надо поставить слова текста. Алгоритмы грамматических преобразований реализованы на сервере. Приведем пример представления текста доверенности с грамматическими конструкциями.

```
...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...
```

Реализованы функции копирования документа, загрузки данных одного документа в другой. Дальнейшее совершенствование данной программной системы ведётся в направлении построения графа документооборота на основе анализа передачи данных между документами. Адрес модуля приложения - https://github.com/isu-enterprise/isu.aquarium.

5 Дальнейшее развитие программных технологий

Сегодня, большое количество информации публикуется в табличном виде в неструктурированных источниках (например, государственная статистика, финансовые и бизнес отчеты, наборы открытых научных данных, социальные медиа). Основными форматами представления данных таких источников являются PDF, растровые сканы, DJVU, а также форматы Word и Excel. Данные, хранимые в этих источниках, являются ценным ресурсом при формировании документов. Например, большинство постановлений и образовательных стандартов Минобрнауки РФ публикуется в формате PDF, причем в сканированном виде. Чтобы иметь возможность использовать данные из этих источников, прежде всего, необходимо извлечь эти данные и трансформировать их в размеченный текст или в таблицы реляционной базы данных. Для решения этой важной задачи коллективом авторов ведется разработка технологий преобразования, анализа, интерпретации, очистки и

отслеживания происхождения табличных данных из неструктурированных источников, концептуализации их естественно-языкового содержания. Создана организация Cells Research Group (https://github.com/cellsrg), в которой реализованы два проекта: система распознавания структуры таблиц в PDF-документах (https://github.com/cellsrg/TabbyPDF); анализ семантической структуры таблиц Excel, основанный на применении правил (https://github.com/cellsrg/cells-ssdc). Результатом распознавания таблицы является выгрузка данных в реляционную таблицу [7].

Разметка LOD для табличных данных неструктурированных источников описывается при помощи онтологии qb (Qube ontology), которая позволяет представлять эти данные в виде гиперкуба. Для каждой ячейки таблицы в самом общем случае задается координата ячейки в кубе и свойства хранимого значения, например, тип данных и единицы измерения. Разметка таблицы в qb позволяет алгоритмически обрабатывать ее непосредственно, т.е. без необходимости предварительного анализа структуры.

Результаты предполагается внедрить в задаче публикации данных результатов исследований в области высокопроизводительного секвенирования ДНК/РНК, провидимых в Лимнологическом институте СО РАН. Для этого необходимо дополнительно размечать текстовые и табличные данные RDFa-конструкциями, поддерживаемыми системами обеспечения доступа к данным LOD в биологии, например, BIO2RDF (http://bio2rdf.org/). В результате реализации данного сервиса ЛИН СО РАН получит возможность интегрировать данные о микробиоме озера Байкал в мировую исследовательскую среду видов растений, животных и микроорганизмов аналогично [8].

Другой важной задачей анализа неструктурированной информации является автоматизация разметки однотипных документов по образцу. Задача актуальна, например, в вузах, где накоплено большое количество материала в виде учебных программ и методического обеспечения, который требует преобразования в различные формы представления: учебные программы по стандарту ФОС-3+, электронные курсы moodle и др. Кроме того, в вузах РФ активно ведутся разработки интеллектуальных систем автоматизации обучения, где материал студенту выдаётся в зависимости от предопределенного общего сценария и результатов оценки его знаний. Размеченный методический и документальный материал позволит частично автоматизировать процесс заполнения базы модулей такой системы электронного обучения.

6 Заключение

В докладе обсуждается вопрос применения технологий Linked Open Data (LOD, открытых связанных данных) для решения задач создания текстовых документов средствами, предоставляемыми современными веб-браузерами. Благодаря нотации LOD реализуется возможность интеграции данных документов и веб-приложений в один результирующий текстовый документ.

Верстка документа осуществляется при помощи алгоритмов, интерпретирующих отношения между элементами дерева HTML. Отношения описываются средствами семантического веба и LOD, в основном, конструкциями RDFa и командами, задаваемыми в атрибутах тегов 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 (запросы GET и POST). Благодаря этому технология очень просто масштабируема даже на сервера хранения данных, функционирующих в режиме разделяемого хостинга и поддерживающих среду программирования PHP. Разработаны тестовые варианты хранилищ на разных языках программирования.

Приведены несколько примеров использования разработанных средств верстки для подготовки документов в образовательной среде, подготовки юридических документов и др. 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 подходов к автоматизации документооборота.

7 Благодарности

Результаты получены при частичной поддержке следующих проектов:

- Иркутского научного центра СО РАН № 4.1.2;
- 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).

Авторы благодарны сообществу Linked Opev Vocabularies (http://lov.okfn.org/dataset/lov/) за помощь, оказанную при поиске формализаций предметных областей (онтологий).

Литература

[1] Bizer, Ch., Heath, T., Berners-Lee, T. Linked Data – The Story So Far. *International Journal on Semantic Web and Information Systems.* 5 (3): (2009) 1–22. doi:10.4018/jswis.2009081901. ISSN 1552-6283.

[2] 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 г., Новосибирск. Издательство «РИЦ Прайс-курьер», 2013. 2 т., 500 с., ISSN 0568-661X

[3] 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

[4] Wielemaker, J., Beek, W., Hildebrand, M., Ossenbruggen, J. ClioPatria: A SWI-Prolog infrastructure for the Semantic Web. *Semantic Web* 7(5):529-541, 2016, DOI: 10.3233/SW-150191

[5] Lager, T., Wielemaker, J. Pengines: Web Logic Programming Made Easy. *Theory and Practice of Logic Programming* 14(4-5), 2014, DOI: 10.1017/S1471068414000192

[6] N.Heino, S.Tramp, N.Heino, S.Auer. Managing Web Content using Linked Data Principles – Combining semantic structure with dynamic content syndication. *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual*. pp. 245 - 250. [Электронный ресурс] URL:http://svn.aksw.org/papers/2011/COMPSAC_lod2.eu/public.pdf (дата обращения: 30.05.2013).

[7] Шигаров А.О. Восстановление логической структуры таблиц из неструктурированных текстов на основе логического вывода // Вычислительные технологии. 2014. Т. 19, № 1, С. 87-99.

[8] Dalamagas, T., Bikakis, N., Papastefanatos, G., Stavrakas, Y., Hatzigeorgiou A. Publishing Life Science Data as Linked Open Data: the Case Study of miRBase *Proceedings of the first International Workshop On Open Data*, WOD-2012. [Электронный ресурс] URL: https://arxiv.org/abs/1205.2320