Платформа реализации электронных архивов данных и документов

© А. Г. Марчук © П. А. Марчук
ИСИ СО РАН, НГУ,
Новосибирск
mag@iis.nsk.su peter@iis.nsk.su

Аннотация

В статье описывается разработанная в Институте систем информатики им. А. П. Ершова СО РАН платформенное решение, предназначенное для реализации электронных архивов данных и документов. Система основывается на фактографическом подходе, в качестве концептуальной базы используются идеи и стандарты группы Semantic Web и собственные наработки. Платформа использована и используется для ряда прикладных, исследовательских и экспериментальных информационных систем.

Работа поддержана программой РАН Р-15/10, грантом РФФИ 11-07-00388а, интеграционным проектом СО РАН М-48.

1 Введение

В работах [1-4] был предложен подход к созданию архивных фактографических систем. Особенностями подхода являются:

- хранение электронных образов документов разной природы и предоставление к ним доступа;

- наличие базы данных о документах и сопряженных с ними сущностях и использование базы данных для структурирования документного массива;

- определение семантики данных через задание онтологии, использование формализмов направления Semantic Web, таких как RDF, OWL;

- распределенное хранение документов и базы данных.

В Институте систем информатики уже более 10 лет создаются архивные системы и совершенствуются технологии электронной фактографии. За последние 2 года технологии приобрели «платформенный» характер, т.е. стали набором взаимоувязанных соглашений, средств и компонентов, в совокупности позволяющих достаточно экономными усилиями создавать конкретную Web-ориентированную информационную систему формирования, наполнения, редактирования публикации и анализа архивных документов и данных.

Платформенное решение представляет собой компромисс между фиксированностью базиса понятий и соглашений и гибкостью комбинирования средств при адаптации создаваемой системы к предметной области. В основном, компромисс обеспечивается использованием формальных спецификаций в виде онтологии данных. Эта онтология может быть сменной и соответствовать видению разработчиками предметной области и стандартов фиксации информации.

Гибкость платформы проявляется и в том, что базовые компоненты способны функционировать в разных операционных системах и с разными системами управления базами данных. Платформа разбита на функциональные слои. Слои соответствуют этапам обработки информации и обеспечению пользовательского доступа к этой информации.

2 Ввод и первичная обработка электронных образов документов

2.1 Основные решения и функциональность

Единицами архивного хранения являются электронные образы документов, представляющие собой файлы разных форматов или файловые сборки. В основном, это мультимедийная информация: электронные образы фотографий, кино, видео, аудио материалов, файлы различных публикационных и сохраняющих форматов (doc, pdf, rtf, html, txt и др.). Файловые сборки являются или технологическими сборками (DVD, zip) или группирующими сборками, такими как папка, тетрадь, содержащими вложенные документы.

По предложенным методикам исходные документы переводятся в цифровой формат и группируются в иерархию документных сборок. Массив полученной информации, соответствующий разделу архива, преобразуется в собранную форму, называемую кассетой. По аналогии с «реальным» архивом, кассета играет роль шкафа для хранения архивных папок. При преобразовании также осуществляется полезная обработка. Дело в том, что

Труды 14-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» — RCDL-2012, Переславль-Залесский, Россия, 15-18 октября 2012 г.

ряд форматов электронного хранения документов имеют либо излишнюю громоздкость для передачи данных в среде Интернет, либо специфичны или архаичны, поэтому, наряду с оригиналом документа в исходном формате, часто формируется его «интернетовская» копия, обеспечивающая оперативный доступ к содержимому документа. Например, для полных оригиналов видео в качественном исполнении, дополнительно вычисляется вариант этого видео в одном из потоковых форматов и со значительным усилением сжатия данных. То же самое касается качественных фото.

Поскольку с информационной точки зрения, архив является совокупностью базы данных и содержимого (контента) документов, важной частью подхода является сочетание базы данных и хранилища документов. Часто, в известных системах, например класса CMS, контент документов «втягивается» в СУБД. Реже, главным в паре база данных-хранилище, выступает хранилище. Так устроены многие репозитории. Недостатком первого подходя является отсутствие регламента на перенос всего архива в другую СУБД. Недостатком второго подхода является примитивность базы данных. Использование RDF в качестве носителя информации базы данных, решает эту проблему. Мы просто помещаем RDF-документы в кассету (хранилище) как и другие документы.

2.2 Структура кассет и прикладной программный интерфейс (API)

Кассета организована в виде директории специального формата. В директории имеется метаинформация о кассете, включая версию и некоторые параметры. Имеется 3 поддиректории: meta, originals и documents. Директория meta предназначена для хранения базы данных самой кассеты, т.е. набора записей о помещенных в кассету документов и о «технической» структуре кассеты. Последнее означает, что документы выстраиваются в иерархию. Иерархия реализуется через применение коллекций документов и подколлекций. Эта иерархия, как правило, создается на этапе подготовки материалов для ввода и архив и может отражать пользовательскую идею о группировании документов. Эта сохраненная иерархия существенно используется в дальнейших этапах обработки введенного документного массива. База данных выполнена в виде RDF-файла и участвует, наряду с другими RDF-документами в построении единой информационной базы конкретных информациионных систем.

Директория originals предназначена для хранения оригиналов сохраняемых в архиве документов. В нее документный файл или файловая сборка помещаются без изменений, замещается лишь название, это замещение требуется для того, чтобы ликвидировать возможное совпадение имен файлов и использование в именах нежелательных символов, включая национальные.

Директория documents предназначена для хранения и организации специально преобразованных документных файлов. Для больших фотографий вычисляются имиджи разных, меньших размеров, видео преобразуется в используемые потоковые варианты, возможно с изменением размеров. Есть и более «экзотические» вычисления. Например, предподготовка для использования технологии DeepZoom [5] при просмотре фотографий большого разрешения или панорамных сборок фотографий.

Прикладной программный интерфейс состоит из двух модулей (проектов): CassetteKernel и CassetteExtension. Первый используется при любых видах работ с кассетами, второй необходим лишь при формировании и изменении кассет. Определяются классы Cassette, CassetteInfo, RDFDocumentInfo. Ключевым является класс Cassette. Этот класс определяет внутреннее устройство кассет, чтение, генерацию и запись. В класс встроены генератор уникальных идентификаторов и таблица определения вида файла по расширению его имени. Набор методов позволяет находить оригиналы или специальные файлы по их URI, порождать поток RDF-файлов, хранимых в кассете, порождать поток записей из метаинформационного документа, добавлять файлы и коллекции и т.д. Классы CassetteInfo и RDFDocumentInfo задают внешнее использование кассет и RDF-документов для случая множественного их использования.

В CassetteExtension собраны статические методы предобработки документных файлов, включая геометрические преобразования, выделение встроенной в файл метаинформации, напр. Exif, использования внешних программ.

2.3 Приложение для формирования и редактирования кассет

Поскольку структура и программный интерфейс кассеты формально описаны, можно создавать различные приложения и Web-приложения для порождения кассет и их редактирования. В предыдущей практике проектировались специальные программы, особенно для преобразования в кассетную форму больших массивов файлов и Web-приложения для пополнения архивов документами силами пользователей. Наиболее функционально развитым приложением по работе с кассетами, является CManager, выполненным в среде .NET-WPF и включенным в состав платформенного решения.

Приложение имеет оконный интерфейс с интуитивно понятным набором действий по навигации и редактированию. Можно создать кассету или подключиться к уже созданной, можно пополнять кассету документами простым «перетягиванием» (технология dag-and-drop) файлов или директорий в логическое место размещения. Можно просматривать документы архива или запускать оригиналы через системные средства.

Также можно выполнять некоторые действия над отдельными документами. Например, можно

поворачивать фотографии. Это действие выполняется с учетом специфики хранения качественных изображений. Поворот и отражение фотографий можно производить и до помещения в архив, но такое действие иногда забывают осуществить, иногда оно не рационально, поскольку может привести к (небольшой) потери качества оригинала. Это будет если мы уже сжатую, напр. jpeg-фотографию будем поворачивать и снова делать jpeg, соответственно внося новые искажения. При архивном повороте, оригинал остается без изменения, а трансформируются только предподготовленные копии.

Для выполнения специфичных или сложных действий, приложение использует свободно распространяемы программы ffmpeg – для преобразования видео и MediaInfo для получения метаинформации о видео.

3 Онтология

3.1 Принципы формирования онтологии

Онтология фиксирует набор сущностных классов и набор отношений между ними. Это фиксируется в основном подходе к определению наборов, так называемом методе ER (Entities - Relationships). Кроме того, сущностные классы, а иногда и отношения выстраиваются в иерархию по соответствию класс - подкласс и отношение - подотношение с наследованием свойств. Наиболее часто используемым формализмом для определения онтологий является язык Web Ontology Language (OWL) [6]. Главным критерием, который можно предъявить к системе структуризации, является адекватность описания объекту описания. При этом адекватность как правило вступает в противоречие с общностью описания. Например, территориальное разбиение страны на административные единицы может быть самым разным: области, штаты, графства, кантоны и др., но для общности, желательно иметь единое понятие, соответствующее такому делению. Первый использованный принцип - онтология должна вводить минимальное число понятий, достаточных для описания объектов мира с требуемой для работы детальностью. Система структуризации не должна иметь прямой зависимости от времени рассмотрения данных, географического положения, культурных и национальных особенностей.

Известные онтологии не всегда соответствуют этому критерию, причем в самых простых вопросах. В качестве примера можно привести довольно часто встречаемое поле “возраст”, которое явно зависит от времени прочтения информации.

3.2 Особенности базовой онтологии неспецифических сущностей (BONE)

Базовая онтология неспецифических сущностей определяет следующие основные классы: персоны, организационные системы, географические системы, документы и отношения между ними. Кроме того, есть ряд дополнительных сущностных классов таких, как коллекции, архивы, предметы и др. На рисунке 4 изображены основные сущностные классы и отношения между ними

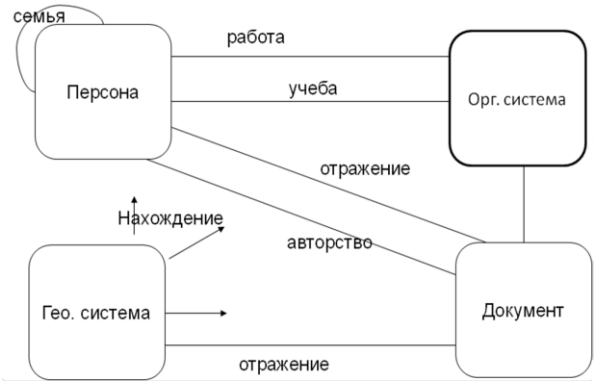

Рис. 1. Основные сущностные классы онтологии BONE

Как видно из рисунка, имеется 4 основных класса сущностей: персоны, организационные системы, документы и географические системы. Под организационными системами понимаются все формальные и неформальные объединения людей для достижения общих целей. В эту категорию попадают организации, туристические группы, клубы, конференции и т.д. На рисунке дугами также указаны основные отношения между классами. Между персонами задаются родственные и семейные отношения, между организационными системами и людьми есть отношения типа “работа” и тапа “учеба”. Документы могут иметь авторов, а это задает отношение авторства, кроме того, документы отражают внешний мир. Например, на фотодокументе может быть изображен конкретный человек или группа людей. Географические системы, через отношение “нахождение” являются местами расположения объектов других классов.

3.3 Прикладной интерфейс онтологии

Онтология описана стандартными средствами OWL и может быть использована соответствующими программами. Тем не менее, имеется два варианта программных компонентов, упрощающих работу с онтологией для тех случаев, когда такая работа необходима. Первый вариант базируется на XML (RDF) представлении онтологического описания. Имеются статические методы, позволяющие находить описания по идентификатору онтологического объекта, отслеживать иерархию классов, получать атрибуты описаний.

Другой вариант рассчитан на получение онтологической информации в условиях интенсивной обработки данных, например в случае выполнения запросов к RDF-данным. В этом случае, онтология представлена графом с дополнительными атрибутами и прямыми ссылками по всем связям. Это – так

называемая модель (проект) SGraph. В целом, модель предназначена для эффективной реализации RDF-графов для случаев, когда данные могут помещаться с оперативной памяти сервера. Для случаев, когда данные имеют объем больше нескольких миллионов высказываний, применяется либо реляционная база данных (см. далее), либо используется механизм кеширования модели.

4 Редактирование базы данных

4.1 Общие положения

После ввода документного массива, пользователи осуществляют описание документов этого массива добавляя информацию в базу данных. Причем техническую информацию о документе, система сама «старается» зафиксировать на стадии преобразования. Это касается характеристик носителя информации (разрешающая способность, формат, конверт и др.) и некоторых данных, зафиксированных в метаинформационных полях файла (время съемки, характеристики съемки и др.). Пользователь описывает информацию, имеющуюся в документе (поля «имя» и «описание»), а главное – «привязывает» документ к объектам базы данных через различные отношения. Через отношения определяются авторы документа, изображенные или отраженные персонажи, темы, связь документа с местом, организацией или событием. Естественно, в базе данных появляются описания сущностей различных видов (персоны, организационные системы, географические объекты, коллекции) и дополнительные связи сущностей между собой.

4.2 Система работы с RDF данными

Основой платформы является система работы с RDF-моделью. Подобные системы часто называют «движок». RDF-модель является некоторой специальной конфигурацией собранного воедино в граф множества RDF-документов. Собственно методика здесь известна и платформенное решение следует простому варианту ее реализации. RDF-документы «разбирается» на множество высказываний – триплетов. И это множество размещается в одной таблице реляционной базы данных – таблице утверждений. Для того, чтобы сделать более эффективной обработку, идентификаторы сущностей преобразуются в целые значения, то же самое делается с литеральными значениями. Соответственно, получаются еще две таблицы: таблица идентификаторов сущностей и таблица литералов. В нашем случае, мы также разбиваем таблицу утверждений на две таблицы, группируя в них DatatypeProperty утверждения и ObjectProperty утверждения. Естественно, таблицы нужным образом индексируются.

Движок сделан как самостоятельный пакет Sema2012, который возможно использовать и в других приложениях. Имеется настройка на два варианта использования СУБД: MS SQL Server и MySQL.

Стандартная схема использования системы работы с RDF-данными следующая. При конфигурировании архивной информационной системы определяются кассеты, хранящие нужные в проекте документы и RDF-файлы базы данных. Затем формируется рабочая реляционная база данных посредством импортирования RDF-документов. Этот процесс выполняется не слишком быстро из-за особенностей ряда конструкций, введенных для возможности осуществления целостной системы редактирования базы данных.

После этого, рабочая база данных готова к оперативному использованию в режиме редактирования данных. При редактировании, производится параллельное изменение как рабочей базы данных, так и RDF-документов, закрепленных за пользователями. Поэтому всегда можно к этапу формирования рабочей базы данных. Такая схема изменения данных применена из-за того, что отображение RDF-документы –> рабочая база данных необратимо.

База данных выполняет запросы по предоставлению доступа к данным. Можно производить поиск по записям определенного класса, задав поисковую строку и указав в каких полях ее надо искать. Также можно производить выборку подграфа в виде древовидной структуры специальной организации. Это соответствует получению некоторой окрестности заданного узла графа.

4.3 Задание шаблонных деревьев

Для работы с RDF-данными в приложениях применяются шаблонные деревья. Это позволяет для большинства случаев минимизировать форму запроса. Так для запроса на получение окрестности сущностного узла достаточно задать уникальный идентификатор (URI) этого узла. По идентификатору, в данных через отношение rdf:type выявляется класс сущности, по классу сущности, среди шаблонных деревьев определяется подходящее, по выбранному шаблонному дереву, из RDF графа извлекается подграф в виде дерева, соответствующего шаблону.

Шаблонное дерево имеет простую структуру, пример дерева приведен следующим фрагментом:

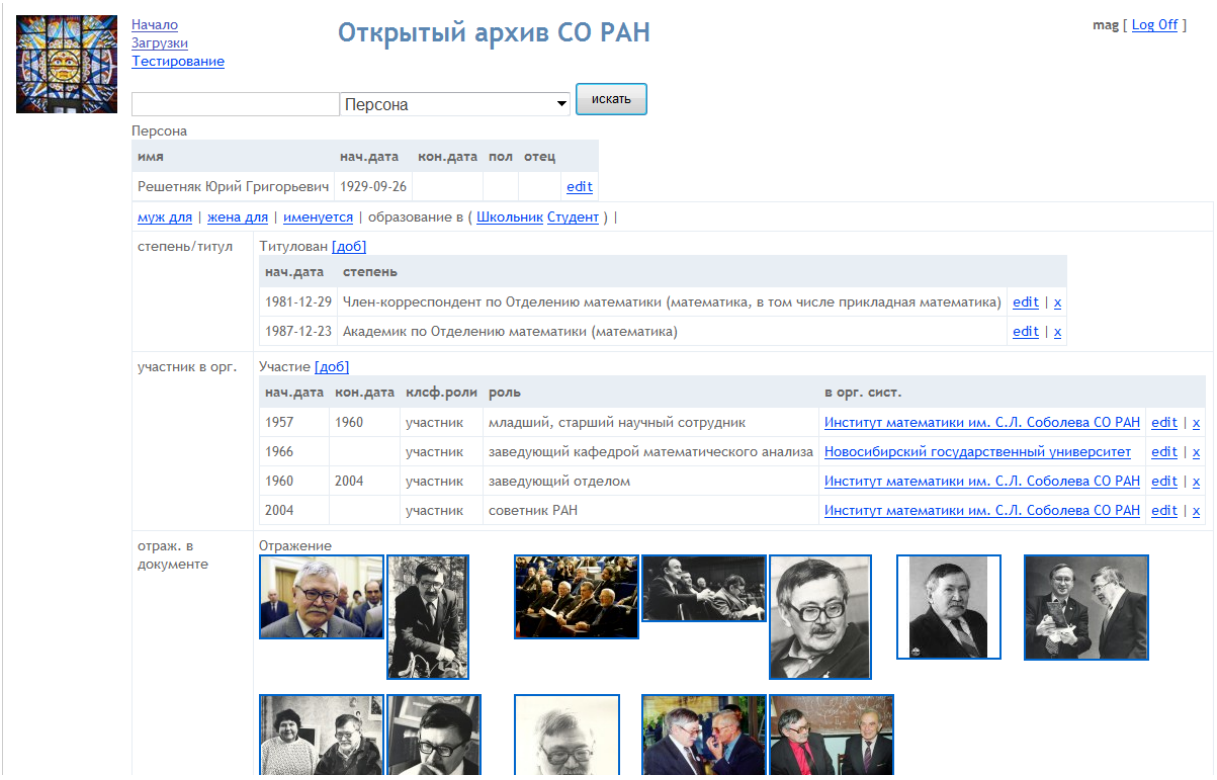

Рис. 2. Интерфейс приложения Ursul для Открытого архива СО РАН

```
<TemplateTree>
  <record type="Person">
    <label xml:lang="ru">Персона</label>
    <field prop="name"/>
    <field prop="startDate"/>
    <inverse prop="participant">
      <record type="Participation">
        <field prop="startDate"/>
        <field prop="endDate"/>
        <field prop="role"/>
        <direct prop="inOrg">
          <record type="orgSys">
            <field prop="name"/>
            <field prop="orgClassification"/>
          </record>
        </direct>
      </record>
    </inverse>
    …
  </record>
  …
</TemplateTree>
```

Здесь, в соответствии с онтологией BONE задан шаблон для выдачи окрестности узла, описывающего персону. Для персоны задаются поля имени и даты рождения, а также все отношения участия (Participation) в организационных системах, характеризующиеся начальной и конечной датой, ролью (напр. должностью), и информацией об организации участия – имени и классификатора.

В принципе, на основе методики, изложенной в [4], шаблонное дерево можно вычислить из онтологии. Но это задаст полный информационный портрет для каждого сущностного класса, что не всегда требуется и ограничит глубину графа. В реализующихся архивных информационных системах шаблонное дерево строится «вручную», в дальнейшем предполагается создать для этого подходящий инструмент.

4.4 Web-приложение редактирования базы данных

Платформенный характер описываемой системы предполагает создание приложений и интерфейсов из компонентов для проектируемых архивных информационных систем. Тем не менее, кроме CManager'а, в платформу был включен еще один готовый программный комплекс – Ursul. Это – Web-приложение, предназначенное для визуализации и редактирования базы данных, предоставления доступа к контенту документов, предоставления доступа к базе данных для внешних информационных систем. На рисунке 2 приведен пример интерфейса приложения.

Приложение Ursul позволяет:

- выполнять поиск в базе данных сущности заданного типа по поисковому образцу;

- получать информационный портрет сущности, включая содержимое документов;

- заполнять и изменять информационные поля объектов, добавлять отношения, редактировать атрибуты отношений, устанавливать через отношения прямые ссылки на объекты;

- производить навигацию по ссылкам.

Для системного администратора допускается изменять набор включенных в проект кассет, добавлять пользователей, изменять их полномочия, производить генерацию временной базы данных, выполнять проверки целостности и корректности базы данных.

Приложение также работает в двух технических режимах: как хранилище документного контента, предоставляющего (как правило – браузеру), файлы документов и как Web-сервис, выполняющий HTTP-запросы внешних агентов по выдаче фрагментов RDF-графа или фиксирующий редактирующее изменение. Информацию сервис выдает в виде XML, упакованный в конверт SOAP.

5 Системная организация

5.1 Публичные и специальные интерфейсы

После применения предыдущих этапов архивная база данных и документов уже сформирована. Ее в дальнейшем можно пополнять и редактировать, но уже можно использовать для решения различных задач. Это осуществляется через создание публичных или специализированных интерфейсов. Если проводить аналогию с музеем, то база данных и документов представляет собой фонд хранения, но требуется также и экспозиции, т.е. специально подобранные и оформленные множества экспонатов. Также требуются средства доступа к архиву, позволяющие проводить научные исследования.

Публичные интерфейсы для архивной фактографичекой системы подразделяются на три класса. Первый класс – поисково-просмотровый интерфейс общего (универсальная экспозиция) или специального назначения (тематические или специальные экспозиции). Использование онтологии позволяет иметь хотя бы один «готовый» публичный интерфейс, таким интерфейсом в платформенном решении является Web-приложение Publicuem. Для конкретных проектов создаются интерфейсы, учитывающие специфику проекта и его документного контента. Например, для проекта «Фотоархив СО РАН», сделано Web-приложение soran1957.ru, часть этих же данных была использована для юбилейного сайта ММФ НГУ.

Второй класс – интерфейсы, позволяющие архиву выступать источником данных для внешних систем. Здесь возможна интеграция как с системами, построенными на данной платформе, так и с системами, предназначенными для интеграции подобной семантически определенной информации. В настоящее время, в институте идет работа над интеграцией с проектом Linking Open Data (LOD) [7, 8]. Ядро системы, ее «движок» может работать как компонент приложения, так и как сервис (Web-сервис), предоставляющий информацию «наружу» по формализованным запросам. Также сервис может изменять содержимое базы данных и документов.

Третьим классом интерфейсов являются аналитические системы, позволяющие анализировать данные и документы по интересующим пользователя профилям. Это пока наименее разработанный в платформе слой. Созданные средства нацелены на анализ корректности и целостности данных. Например, есть анализ на предмет выделения записей о потенциально одних и тех же сущностях, производится поиск «накопившихся» ошибок в данных, анализируются цепочки переименований и др.

5.2 Эволюция системы

Архивные системы предназначены для фиксации данных на длительный период времени. При нынешних темпах изменений в информационных технологиях, это несет группу проблем, решение которых является принципиальным для использования той или иной системы. К таким проблемам относятся: возможные изменения системы структуризации и модели представления RDF-данных, изменения в онтологии, изменения в способах и форматах хранения данных и документов, изменения в структуре файлового представления документов и появление новых вариантов такого представления.

Главное, что позволяет с оптимизмом смотреть на будущие трудности в отслеживании изменений – это формальные спецификации основных моментов, связанных со структуризацией, применяемой онтологии, структурой хранения (кассеты). База данных формируется в стандартном XML-RDF, который не только «проживет» не одно десятилетие, но и удобен для выполнения преобразований в случае регулярных изменений. Специального инструментария для этих преобразований не создается, поскольку направления изменений неизвестны.

Собственно процесс эволюции данных идет уже не один год. Внесен ряд изменений и в базовую схему структуризации и в онтологию и в специфические документные форматы файловых представлений документов (DeepZoom, MPEG-4 и др.). Предполагается произвести большую переработку онтологии BONE: будут изменены идентификаторы классов и отношений, объединены некоторые классы, устранены неиспользуемые описания и т.д. Такие и подобные изменения не повлияют сейчас и в дальнейшем на сохранность документного массива и базы данных.

5.3 Состав и особенности реализации системы

Основой платформенного решения являются модули (в терминах MS Visual Studio – проекты), обеспечивающие реализацию базовых действий. К таким модулям относятся модули работы с кассетами, модули или программы работы с первичными документами, модуль работы с RDF-моделью (движок). Целостные и достаточно сложные действия оформляются в виде Windows или Web приложений. Как уже указывалось,

имеется WPF-приложение CManager, выполняющее основные действия по созданию кассет и наполнению их документным материалом. Имеется Web-приложение Ursul, представляющее собой редактор базы данных архивной системы, но предоставляющее также дополнительные сервисы по доступу к документам и доступу к движку.

В дальнейшем предполагается дооснастить платформу типовым решением публичного интерфейса. Предположительно, в платформенном варианте система будет поставляться в виде проекта для Microsoft WebMatrix или Microsoft Visual Studio.

5.4 Использованные технологии

Система написана на языке C# с использованием Linq, технологий .NET, ASP.NET MVC, программа CManager написана в WPF, в качестве СУБД возможно применение MS SQL Server и MySQL. Решение эксплуатируется в ОС Windows, совместно с IIS. Проводились успешные эксперименты по погружению системы в Linux под платформой Mono. Также проводились эксперименты по интеграции решения с сайтом, построенным на CMS Drupal.

Ряд программ или модулей взят из внешних источников. К таким программам относятся программы обработки видео и аудио контента, получение метаинформации из документных файлов, создание и использования многостраничных сборок имиджей по технологии DeepZoom.

Литература

[1] Марчук А.Г. Распределенные электронные архивы, библиотеки и базы данных // Препринт 122, Институт систем информатики им. А.П. Ершова СО РАН, Новосибирск – 2004. — 25 с.

[2] Марчук А.Г., Марчук П.А. Платформа интеграции электронных архивов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции. Труды девятой всероссийской конференции. Переславль, 2007, с. 89-94.

[3] Marchuk A.G. Methods and Technologies of Digital Historical Factography // Knowledge Processing and Data Analysis. First International Conference, KONT 2007, Novosibirsk, Russia, September 14-16, 2007,and First International Conference, KPP 2007, Darmstadt, Germany, September 28-30, 2007. Revised Selected Papers. Series: Lecture Notes in Computer Science, Vol. 6581, Subseries: Lecture Notes in Artificial Intelligence, Wolff, K.E.; Palchunov, D.E.; Zagoruiko, N.G.; Andelfinger, U. (Eds.), 2011, ISBN 978-3-642-22139-2, pp 217-231

[4] Ануреев И.С., Батура Т.В., Боровикова О.И., Загорулько Ю.А., Кононенко И.С., Марчук А.Г., Марчук П.А., Мурзин Ф.А., Сидорова Е.А., Шилов Н.В. Модели и методы построения информационных систем, основанных на формальных, логических и лингвистических подходах / Отв. ред. А.Г. Марчук ; Рос. акад. наук, Сиб. отд-ние, Ин-т систем информатики им. А.П. Ершова. – Новосибирск: Изд-во СО РАН, 2009. ISBN 978–5–7692–1113–3. – 330 с.

[5] DeepZoom // http://msdn.microsoft.com/en-us/library/cc645050%28VS.95%29.aspx

[6] OWL Web Ontology Language Overview // http://www.w3.org/TR/2004/REC-owl-features-20040210/, 2004

[7] Tim Berners-Lee Linked Data / http://www.w3.org/DesignIssues/LinkedData.htm, 2006.

[8] Tom Heath and Christian Bizer Linked Data: Evolving the Web into a Global Data Space (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool, 2011, 1-136.

Platform for Digital Data and Documents Archive Implementation

Alexander G. Marchuk, Peter A. Marchuk

In the article, a new platform solution for digital data and documents archive information systems implementation is described. It was build in the A.P.Ershov Institute of Informatics Systems. System is based on factographic approach and concepts, recommendations and standards of Semantic Web. A new approach for digital archiving is presented. This approach consists from several principles of structuring documents and data which are implemented in specifications and modules. Platform was used and still in use in several research and applied digital archive information systems.