Подход к проектированию персонализованных электронных библиотек над Web-сайтами со слабоструктурированными данными

Л. А. Калиниченко, Н. А. Скворцов, Д. О. Брюхов, Д. В. Кравченко, И. А. Чабан
Институт проблем информатики Российской Академии наук
Москва, В-334, 117900, ул. Вавилова, 30/6
E-mail: [leonidk,scvora,brd,dmitry,chb]@synth.ipi.ac.ru

Аннотация

В настоящей статье рассматриваются вопросы проектирования персонализованных электронных библиотек над существующими электронными коллекциями слабоструктурированных данных в Web. Описываемая методика проектирования позволяет создавать библиотеки, направленные на персонализованное обслуживание потребителей информации, предъявляющих свои требования к составу и представлению необходимой им информации. Электронная библиотека разрабатывается, как композиция фрагментов Web-сайтов. Данная работа демострирует применение разработанных в ИПИ РАН методов композиционного проектирования информационных систем [2] к слабоструктурированным данным и Web.

1 Введение

Представленный подход[1] направлен на создание виртуальных электронных библиотек, соответствующих специфическим информационным потребностям пользователей. Такие потребности предлагается задавать в виде спецификаций создаваемых для их нужд коллекций. Для уточнения семантической нагрузки описаний спецификации ее элементы привязываются к понятиям онтологии предметной области, к которой относится требуемая персонализованная библиотека. Для использования информации из Web-сайтов как электронных коллекций необходимо иметь спецификации их схем, также привязанные к онтологиям предметных областей этих сайтов. Используемые методы проектирования базируются на принципах повторного использования и композиционного проектирования информационных систем. На основе отображения онтологий предметных областей сайтов и требуемой электронной библиотеки в общий онтологический контекст (shared ontology), близкий к данным областям знаний, и дальнейшего анализа спецификаций схем выявляются фрагменты описаний сайтов, соответствующие фрагментам спецификаций требований, так чтобы части схем сайтов уточняли [7] спецификации схемы разрабатываемой библиотеки. Реализация библиотеки будет состоять в композиции таких фрагментов.

Программная поддержка рассматриваемого подхода основана на прототипе CASE-системы СИНТЕЗ, разработанном в ИПИ РАН для композиционного проектирования информационных систем [2].

Следующие подразделы "Введения" (1.1–1.4) посвящены базовым понятиям предлагаемой методики проектирования. В разделе 1.1 Web-сайты, являющиеся коллекциями-источниками для персонализованной электронной библиотеки, рассматриваются как базы данных со слабоструктурированной информацией. Основные принципы персонализации электронных библиотек представлены в разделе 1.2. Применение композиционных методов проектирования информационных систем для создания электронных библиотек над информационными ресурсами в Web обсуждается в разделе 1.3. Для описания гетерогенных данных и работы с ними однородным образом используется модель данных СИНТЕЗ, кратко охарактеризованная в разделе 1.4.

Далее, в разделе 2, представлен пример, основанный на реальных источниках информации, на котором будут демонстрироваться используемые методы проектирования электронных библиотек. В качестве коллекций выбраны сайты патентных служб США и Канады. Описываются их схемы и схема спецификаций требований к персонализованной библиотеке.

Разделы 3–4 посвящены собственно проектированию: подходам к отображению онтологий, онтологической интеграции схем сайтов и спецификаций требований, выявлению релевантных фрагментов схем и их композиции в требуемую коллекцию.

1.1 Электронные коллекции со слабоструктурированной информацией в Web

Web является богатейшим источником всевозможной информации в различных предметных областях для создания электронных библиотек. Однако установление семантики информации в Web для ее дальнейшего использования затруднено в условиях неструктурированности или слабоструктурированности и разнородности ее представления. Это одна из причин актуальности исследований задач структуризации информации в Web.

В настоящее время интенсивные исследования направлены на выявление структуры данных в сайтах с целью их трактовки как баз данных. Так, структуриро-

[1] Данная работа проведена при поддержке Российского фонда фундаментальных исследований грант 98-07-91061 и INTAS-OPEN грант 97-1109

Первая Всероссийская научная конференция
ЭЛЕКТРОННЫЕ БИБЛИОТЕКИ:
ПЕРСПЕКТИВНЫЕ МЕТОДЫ И ТЕХНОЛОГИИ,
ЭЛЕКТРОННЫЕ КОЛЛЕКЦИИ
19 - 21 октября 1999 г., Санкт-Петербург

ванная составляющая HTML-сайтов может быть выявлена на основе анализа тегов гипертекстовых документов и обнаружения их структурных закономерностей. W3C вводит стандарт XML, в котором структура документов задается в разделе описаний типов документов (DTD) [13]. Структура страниц может быть общей для некоторого множества документов или уникальной для определенных документов. Таким образом, неструктурированная информация, представленная в Web-страницах, подчиняется определенной схеме, соответствующей сайту как коллекции документов.

В данной работе вопрос о методах выявления схем HTML-документов не затрагивается. Примером исследования в этом направлении является проект Araneus [9], использующий модель ADM для представления схем HTML-сайтов и специальные подходы и инструменты для выявления схем и манипулирования данными сайтов [5, 1]. Методы Araneus удобно использовать совместно с развиваемым здесь подходом создания электронных библиотек: модели сайтов на языке Araneus достаточно отобразить в каноническую модель СИНТЕЗ (раздел 1.4). Предлагаемый метод предназначен для работы с произвольными представлениями Web-сайтов (не только тем, который обеспечивается ADM). В частности, отображение в каноническую модель СИНТЕЗ модели данных стандарта XML рассматривается в [8].

1.2 Проектирование персонализованных электронных библиотек

Персонализованный подход к обслуживанию потребителей информации обычно предполагает наличие процедуры регистрации конкретных групп для определения области их информационных интересов. Спецификация персонализации в разных электронных библиотеках может быть различной. Например, пользователь может задавать свои требования, расширяя тезаурус и рубрикатор библиотеки или указывая, какие термины тезауруса относятся к его полю деятельности или каким рубрикам библиотеки должны принадлежать интересующие его данные. Наконец, интерфейс конкретного пользователя может включать специфические сервисы, отличающиеся от поисковых интерфейсов обычных электронных библиотек или поисковых процессоров.

Данная работа ограничивается рассмотрением спецификации требований в виде схемы проектируемой электронной библиотеки над Web-ресурсами. Тем самым пользователь определяет состав и форму требуемой информации. Другими словами, в данном случае процедура его регистрации и состоит в задании такой схемы. В качестве исходных информационных коллекций будут выступать те Web-сайты, которые способны привнести полезные фрагменты в проектируемую коллекцию, конкретизирующую спецификации требований.

1.3 Композиционное проектирование

Для пользователя электронной библиотеки важен факт получения информации по запросам, и гораздо реже — знание реального источника этой информации, если в запросах явно не задан интересующий источник. Планирование обработки запросов и выбор реальных источников данных происходит прозрачно для человека. Электронную библиотеку над информацией в Web можно рассматривать как информационную систему, в состав которой входит некоторое множество удаленных источников данных, каждый из которых представлен в глобальной схеме репозитория данной системы.

В рассматриваемом подходе электронная библиотека включает зарегистрированные в ней электронные коллекции в Web в качестве своих информационных компонент. Она проектируется посредством композиции фрагментов схем существующих компонент-источников информации и функционирует, как единая распределенная система, в состав которой входят эти компоненты. Композиционное проектирование электронных библиотек основано на уточнении спецификаций [7] требований соответствующими им частями спецификаций схем существующих источников данных. Согласно рассматриваемому подходу проектирование включает следующие этапы [2]:

1. Приведение спецификаций схем коллекций в Web к описаниям в единой канонической модели [6], связывание элементов спецификаций с понятиями онтологических контекстов соответствующих предметных областей, отображение онтологических описаний предметных областей коллекций и разрабатываемой системы в контекст общей онтологии. Этот этап проводится для любых сайтов однократно для подготовки их к использованию в проектировании электронных библиотек вне зависимости от спецификаций требований к конкретной библиотеке.

2. Отображение онтологического контекста проблемной области спецификаций требований в контекст общей онтологии и онтологическая интеграция схем коллекций со спецификациями требований. В результате онтологического анализа сопоставляются элементы описания схем сайтов и спецификаций требований, описывающие совместимые по смыслу понятия.

3. Выбор среди онтологически релевантных элементов описаний схем сайтов тех, которые могут быть использованы для конкретизации определенных фрагментов спецификации требований. Разрешение структурных конфликтов и рассогласований между схемами.

4. Реализация схемы требуемой электронной библиотеки как композиции выбранных фрагментов схем электронных коллекций. Проектирование адаптеров (wrappers) над электронными коллекциями, посредством которых электронная библиотека сможет их использовать.

1.4 Каноническая модель представления метаданных электронных библиотек

Язык СИНТЕЗ используется для унифицированного представления и манипулирования неоднородными информационными ресурсами. Важно отметить, что СИНТЕЗ предоставляет модели метаданных фактически любого вида и является расширяемым для отражения дальнейшего развития моделей данных и технологий. Он создает богатые возможности для описания гетерогенных информационных ресурсов: структурированных (информационные системы, базы данных), слабоструктурированных (гипертекстовые документы) и неструктурированных (текстовые коллекции) данных, элементов баз знаний, онтологических спецификаций, деятельностей,

потоков работ. Полное описание языка СИНТЕЗ можно найти в [6].

Единицей описания в СИНТЕЗе является фрейм. На базе языка фреймов создана объектная модель языка. В основе объектной модели лежит понятие абстрактного типа данных (АТД), служащее для описания типов данных любой природы. Описание абстрактного типа данных включает спецификации атрибутов, ассоциаций, инвариантов и операций типа. Операции типов описываются типом функции. Ассоциации могут задаваться с помощью метаклассов ассоциаций, где описывается вид ассоциации любой сложности. Иерархия типов задается отношением подтипа. В СИНТЕЗе определен набор базовых типов.

Классы в языке СИНТЕЗ представляют совокупности однородных объектов предметной области. Каждый объект из совокупности является экземпляром данного класса. С классом связан экстенсионал, содержащий множество экземпляров класса. Сами классы как объекты могут быть типизированы, могут определяться их интерфейсы. Классы принадлежат иерархии, основанной на отношении обобщения/специализации. В языке СИНТЕЗ существуют предопределенные классы. Например, `class`, `type`, `concept` — метаклассы, экземплярами которых являются все определенные в данном ресурсе классы, типы и типы онтологических понятий соответственно. Принадлежность к этим метаклассам определяет вид конкретного объекта.

В СИНТЕЗе вводятся следующие операции над типами [7]. Унарная операция взятия редукта типа (`reduct`) определяет подмножество спецификаций этого типа как его супертип. Операция пересечения двух типов (`meet`) имеет результатом тип, включающий семантически общую часть спецификаций типов-операндов. Операция соединения двух типов (`join`) производит тип, включающий объединение спецификаций типов-операндов. Обозначение приведенных операций следующее:

- редукт типа T (`reduct`): $\sim T$;
- пересечение типов T_1 и T_2 (`meet`): $T_1 \sqcap T_2$;
- объединение типов T_1 и T_2 (`join`): $T_1 \sqcup T_2$.

Неструктурированные данные, как и любой другой вид информации, представляются в языке в виде фреймов. Именно неструктурированную информацию могут содержать значения слотов фреймов без указания имен слотов. Слабоструктурированные данные состоят из структурированной и неструктурированной частей. Для описания структурированной части определяются типы. Описание неструктурированной части информации будет представлено слотами, типы значений которых определяются динамически.

Единицей описания ресурса в СИНТЕЗе является схема. В схеме могут быть описаны один или несколько модулей. В частности, для сайтов или разрабатываемой библиотеки в схеме СИНТЕЗа описывается модуль, содержащий типы и классы спецификаций схемы сайта или спецификаций требований, и его подмодуль с онтологическими определениями соответствующей предметной области.

2 Пример

В статье рассматривается пример проектирования персонализованной электронной библиотеки, предоставляющей пользователю информацию о патентах, зарегистрированных на территории Соединенных Штатов или Канады. Спецификация требований к библиотеке задана объектной моделью СИНТЕЗа посредством схемы `patentLibrary`. В схеме описывается тип `Patent` и соответствующий ему класс `patent`, который будет содержать объекты данного типа.

```
{ patentLibrary;
  in : schema;

  type:
  { Patent;
    in : type;
    title : string;
      metaslot
      obl : invariant,
      {{obligatory;}}
      end
    inventors : string;
      metaslot
      obl : invariant,
      {{obligatory;}}
      end
    category : string;
    country : string;
    regDate : string;
    abstract : string;
    claims :
      { union; type_of_label:integer;
        1: string;
        2: {sequence;
          type_of_element:string}
      };
    descr : string
  };

  class_specification:
  { patent;
    in : class;
    instance_section: Patent
  }

  { comment;
  title - заголовок патента,
  inventors - авторы изобретения,
  category - категория патентов
    в международной классификации,
  country - страна с преимущественным
    правом на изобретение,
  regDate - дата регистрации патента,
  abstract - аннотация к изобретению,
  claims - список элементов новизны
    или описание предмета изобретения,
  descr - описание изобретения.
  }
}
```

Библиотека организуется над двумя реальными электронными коллекциями — Web-сайтами, содержащими информацию о патентах, зарегистрированных в Соединенных Штатах Америки [11] и Канаде [12]. В них представлена текстовая информация, содержащая аннотацию, патентуемые нововведения, сведения об авторах, графические изображения с фотокопиями оригиналов

патента и другие данные. Схемы Web-сайтов весьма отличны друг от друга. Они выявляются на основе анализа документов коллекций и представляются в модели ADM на языке определения данных Araneus [9].

```
SCHEME CanadaScheme
  PAGE-SCHEME CanadaPatentPage
    PatentNumber : TEXT ;
    Title        : TEXT ;
    ToImages     : LINK-TO ImgPage;
    Inventors    : TEXT ;
    Owners       : TEXT ;
    FilingDate   : TEXT ;
    CanadClass   : TEXT ;
    InterClass   : TEXT ;
    PriorCountry : TEXT ;
    Abstract     : TEXT ;
    ToClaims     : LINK-TO ClaimsPage;
    END
  PAGE-SCHEME ClaimsPage
    PatentNumber : TEXT ;
    Title        : TEXT ;
    Claims       :
      TEXT UNION LIST-OF(Claim : TEXT;)
      OPTIONAL;
    END
  PAGE-SCHEME ImgPage
    ImgList : LIST-OF (Img : IMAGE;);
    END
END-SCHEME

SCHEME USAScheme
  PAGE-SCHEME USApatentPage
    PubNumber : TEXT ;
    Abstract  : TEXT OPTIONAL;
    Title     : TEXT ;
    Inventors : TEXT ;
    Assignee  : TEXT OPTIONAL;
    Filed     : TEXT ;
    USClass   : TEXT ;
    INClass   : TEXT ;
    Claims    : TEXT OPTIONAL;
    Descr     : TEXT OPTIONAL;
    END
END-SCHEME
```

Для сайта в Araneus определяется схема, состоящая из описаний типов страниц, встречаемых в нем. Структура страниц специфицируется атрибутами. В них могут быть описаны неструктурированные данные (TEXT), изображения (IMAGE), ссылки (LINK-TO), списки (LIST-OF), альтернативы в структуре (UNION) и др. Необязательные атрибуты помечаются как OPTIONAL. Приведенные схемы несколько сокращены, реально страницы этих сайтов содержат больше информации о патентах. После обнаружения схем сайтов для возможности вовлечения их в проектирование необходимо отобразить их из ADM в каноническую модель СИНТЕЗ.

Спецификации информационных ресурсов описаны в виде схем canadaScheme и USAScheme, задающих типы, соответствующие спецификациям страниц сайтов, и классы, соответствующие множествам таких страниц. Атрибуты схем, описывающие ссылки на другие страницы задаются в виде ассоциаций. В данном случае, достаточно одного вида ассоциации — один к одному,— который определен метаклассом ассоциаций one_to_one.

```
{ one_to_one;
  in: metaclass, association;
  inverse: i_oneone;
  instance_section:
  { association_type:
    {{0,1},{1,1}}
  }
}
```

Далее приведены спецификации схем сайтов на языке СИНТЕЗ.

```
{ canadaScheme;
  in : schema;

  type:
  { CPatPage;
    in : type;
    PatentNumber : string;
      metaslot
      obl : invariant,
      {{obligatory;}}
      end
    Title : string;
    ToImages : ImgPage;
      metaslot
      in: one_to_one;
      end
    Inventors : string;
    Owners : string;
    FilingDate : string;
    CanadClass : string;
    InterClass : string;
    PriorCountry : string;
    Abstract : string;
    ToClaims : ClaimsPage;
      metaslot
      in: one_to_one;
      end
  },

  { ClaimsPage;
    in : type;
    PatentNumber : string;
    Title : string;
    Claims :
      { union; type_of_label:integer;
        1: string;
        2: {sequence;
          type_of_element:string}
      };

  },

  { ImgPage;
    in : type;
    ImgList : {sequence;
      type_of_element:Image}
  };

  class_specification:
  { cPatPage;
```

```
    in : class;
    instance_section: CPatPage
  },
  { claimsPage;
    in : class;
    instance_section: ClaimsPage
  },
  { imgPage;
    in : class;
    instance_section: ImgPage
  }
}

{ USAScheme;
  in : schema;

  type:
  { USPatPage;
    in : type;
    PubNumber : string;
    Abstract : string;
    Title : string;
    Inventors : string;
    Assignee : string;
    Filed : string;
    USClass : string;
    INClass : string;
    Claims : string;
    Descr : string
  };

  class_specification:
  { usPatPage;
    in : class;
    instance_section: USPatPage
  }
}
```

В описании схемы `USAScheme` присутствует один список (`ImgList`). Этот список графических изображений представлен атрибутом, типом значений которого является последовательность изображений. `Image` — абстрактный тип данных, требующий определения, которое в примере не приводится. В приведенных спецификациях также должны быть определены метаслоты с инвариантами определенности для всех обязательных атрибутов в описаниях на Araneus, аналогично тому, как это сделано для атрибута `PatentNumber` типа `CPatPage`. Такие инварианты говорят о том, что атрибуты, для которых они определены, не могут принимать пустого значения (`none`). Для частей страниц, допускающих альтернативы в структуре, используется тип объединения (атрибут `Claims` в типе `ClaimsPage`).

Локальные онтологические модули, описывающие контексты предметных областей двух данных коллекций и электронной библиотеки, должны также специфицироваться внутри приведенных схем как подмодули модулей описания их типов и классов.

Полученные спецификации становятся частью большого репозитория, содержащего описания схем многих сайтов. Этот репозиторий используется для поиска Web-коллекций, которые могли бы использоваться как компоненты при проектировании конкретных электронных библиотек.

3 Спецификации онтологических понятий предметных областей сайтов

Со спецификациями локальных электронных коллекций и требований в разрабатываемой системе должны быть связаны онтологические контексты, содержащие понятия предметных областей, которым принадлежит информация, представленная в коллекциях. Описание онтологических понятий приводится в канонической модели того же языка СИНТЕЗ, что используется для спецификации сайтов и требований к электронной библиотеке. Онтологический контекст представляется модулем онтологических спецификаций, являющимся подмодулем модуля спецификаций соответствующей схемы или требований к системе. Стоит отметить, что и контекст общей онтологии должен быть представлен спецификациями в канонической модели. Для этого разрабатываются отображения наиболее известных моделей представления онтологий в каноническую модель СИНТЕЗ.

С понятиями как общей онтологии, так и локальных онтологических контекстов связаны их вербальные определения и списки дескрипторов. Вербальные определения напоминают определение слов в толковом словаре. Списки дескрипторов в спецификациях понятий онтологии строятся на основе значащих слов из вербального определения понятия. При формировании списков дескрипторов используются средства лексического и морфологического анализа. В роли дескрипторов могут участвовать нормализованные слова или основы слов. Дескрипторы понятий используются для предварительного установления связей с другими понятиями вне данного онтологического контекста.

Между понятиями онтологии могут задаваться связи обобщения/специализации (понятия/подпонятия) и позитивные связи (синонимии). Эти связи могут быть нечеткими, то есть иметь силу от 0.0 до 1.0. Значение силы связи, не заданное явно, устанавливается равным 1.0. Понятия также могут характеризоваться атрибутами, ассоциациями, наложенными на них логическими ограничениями. Элементы спецификаций схем коллекций и требований к создаваемой библиотеке становятся экземплярами классов, соответствующих понятиям связанных с данными схемами онтологических контекстов, если по смыслу они соответствуют этим понятиям.

Для сопоставления схем сайтов и библиотеки на основе имеющейся онтологической информации необходимо привести локальные онтологические контексты коллекций и создаваемой электронной библиотеки к одному онтологическому контексту. С этой целью производится отображение понятий из онтологических контекстов электронных коллекций и библиотеки в понятия общей онтологии [3]. На первом этапе производится установление связей между понятиями из разных контекстов с помощью вычисления коэффициентов корреляции между понятиями на основе анализа их вербальных определений. Затем при необходимости может проводиться более глубокая и точная интеграция с учетом внутренней структуры понятий.

3.1 Отображение локальных контекстов в общую онтологию

Интеграция локальных контекстов онтологических описаний сайтов и спецификаций требований основывается на отображении их в общую онтологию. Интеграция на уровне вербальных описаний производится на осно-

ве анализа списков дескрипторов понятий с целью отображения понятий онтологии одного контекста в другой. Предварительно вычисляется степень связи понятий двух онтологических контекстов при помощи векторного подхода [10]. Пусть X и Y — понятия разных онтологических контекстов, локального и общей онтологии. Пусть V_X и V_Y — векторы, состоящие из дескрипторов, определяющих соответствующие понятия X и Y. Векторы C_X и C_Y генерируются для векторов V_X и V_Y и содержат списки весов W_{Xk} и W_{Yk} соответственно для каждого возможного дескриптора k. Веса вычисляются по следующим формулам [10]:

$$W_{Xk} = \frac{(1 + \frac{f_k}{f_{max}}) \cdot \log \frac{N}{n_k}}{\sqrt{\sum_{V_X}((1 + \frac{f_i}{f_{max}}) \cdot \log \frac{N}{n_i})^2}}$$

$$W_{Yk} = \frac{f_k \log \frac{N}{n_k}}{\sqrt{\sum_{V_Y}(f_i \log \frac{N}{n_i})^2}}$$

где f_{Xk} и f_{Yk} — частота дескриптора k в векторах V_X и V_Y соответственно, f_{max} — максимальная частота дескриптора в векторе V_X или V_Y, N — количество понятий общей онтологии, n_k — количество понятий общей онтологии, вектор V_Y которых содержит дескриптор k. Первый множитель произведения увеличивает значимость часто упоминаемых в определении дескрипторов. Второй множитель увеличивает значимость тех дескрипторов, которые встречаются в меньшем количестве векторов V_Y, соответствующих понятиям общей онтологии. Частоты в векторах V_X сведены к интервалу от 0.5 до 1.0, так как каждый дескриптор понятия локальной онтологии важен при нахождении соответствующих понятий. Веса W_{Yk} и W_{Yk} нормализованы для того, чтобы избежать зависимости от разницы длин векторов дескрипторов для разных X и Y. В случае существования словарей или тезаурусов с весовыми значениями слов могут использоваться другие подходы определения весов дескрипторов.

Функции оценки корреляции онтологических понятий определяются следующим образом [10, 3]:

$$sim(X,Y) = \frac{\sum_{k=1}^{t}(W_{Xk} \cdot W_{Yk})}{\sqrt{\sum_{k=1}^{t}(W_{Xk})^2 \cdot \sum_{k=1}^{t}(W_{Yk})^2}}$$

$$r(X,Y) = \frac{\sum_{k=1}^{t} \min(W_{Xk}, W_{Yk})}{\sqrt{\sum_{k=1}^{t}(W_{Xk})^2}}$$

$$r(Y,X) = \frac{\sum_{k=1}^{t} \min(W_{Xk}, W_{Yk})}{\sqrt{\sum_{k=1}^{t}(W_{Yk})^2}}$$

Область значений функции $sim(X,Y)$ — вещественные числа в интервале от 0.0 до 1.0. Значение 0.0 означает, что понятия не связаны друг с другом. Значение 1.0 свидетельствует об идентичности понятий. Такие понятия имеют одинаковые списки дескрипторов. Связанными положительными связями с данным понятием считаются те понятия другого контекста, для которых значение функции $sim(X,Y)$ больше некоторого порогового значения ℓ, само же значение функции в этом случае выступает в качестве силы этой связи.

Функции $r(X,Y)$ и $r(Y,X)$ служат для нахождения вероятных связей понятия/подпонятия между понятиями различных контекстов. Для корректной работы этих функций необходимо, чтобы веса дескрипторов были нормализованы. При использовании приведенного выше метода вычисления весов это условие выполняется. Если $r(X,Y)$ и $r(Y,X)$ меньше некоторого заданного порогового значения ℓ, то понятия X и Y не связаны друг с другом. Если $r(X,Y)$ и $r(Y,X)$ больше ℓ, то они связаны позитивной ассоциацией с коэффициентом корреляции, равным минимальному из этих значений. Если $r(X,Y)$ больше ℓ, а $r(Y,X)$ меньше ℓ, тогда понятие X является суперпонятием понятия Y, другими словами между понятиями X и Y устанавливается ассоциация обобщения. Коэффициент корреляции при этом равен значению $r(X,Y)$. И наоборот, если $r(X,Y)$ меньше ℓ, а $r(Y,X)$ больше ℓ, тогда понятие X является подпонятием понятия Y, другими словами между понятиями X и Y устанавливается ассоциация специализации. Коэффициент корреляции при этом равен значению $r(Y,X)$. Результаты автоматического отображения понятий из контекста в контекст могут корректироваться при надобности.

На данном этапе не затрагивается внутренняя структура понятий и связанные с ними логические ограничения. Если процесс интеграции онтологических контекстов реализуется только на основе вербальных описаний, то при дальнейшей интеграции данных, релевантных понятиям контекстов, могут использоваться только сами понятия без их внутренней структуры, положительные связи и связи иерархии обобщения между понятиями. Интеграция внутренней структуры понятий онтологических контекстов предполагает манипулирование понятиями как спецификациями типов. Особенности интеграции внутренней структуры онтологических понятий выходят за рамки вопросов, рассматриваемых в статье.

3.2 Онтологическая интеграция схем

После отображения локальных онтологических контекстов в контекст общей онтологии необходимо перейти к онтологической интеграции схем. Основная цель онтологической интеграции схем — обнаружение среди описаний информационных коллекций типов, классов и их фрагментов, онтологически релевантных спецификациям требований к разрабатываемой электронной библиотеке. Задача состоит в связывании онтологически релевантных элементов описания спецификаций библиотеки с элементами спецификаций Web-сайтов.

В интеграции участвуют три вида онтологических контекстов (модулей):

- онтологический модуль приложения (application ontology module, AOM) содержит спецификации онтологического контекста персонализованной электронной библиотеки и связан с модулем спецификации требований к библиотеке;
- онтологический модуль ресурса (resource ontology module, ROM) содержит онтологические спецификации предметной области конкретной информационной коллекции и связан с модулем объектной модели спецификации схемы соответствующего Web-сайта;

- общий онтологический модуль (common ontology module, COM) содержит общую онтологию конкретной предметной области, близкой к той, в которой разрабатывается библиотека. COM служит для связывания понятий спецификаций требований и понятий схем сайтов.

В результате работы этапа интеграции локальных контекстов (AOM и ROM) с общей онтологией (COM) по вербальным описаниям появляются позитивные связи и связи понятия/подпонятия между локальными контекстами приложения и ресурсов и общим контекстом онтологии предметной области. Для предварительной, онтологической интеграции элементов описания спецификаций требований со спецификациями существующих ресурсов необходимо найти связи между контекстом AOM и контекстами ROM. Их можно установить на основании внутренних связей обобщения и внутренних позитивных связей в COM в сочетании с установленными межконтекстными связями, объединяющими понятия локальных онтологий с понятиями COM.

Связанными с понятием из AOM могут быть понятия ROM, имеющие с ним положительную связь или связь специализации, то есть синонимичные ему или являющиеся его подпонятиями. Для выявления таких связей необходимо анализировать пути соответствия понятий AOM понятиям ROM и дополнять граф понятий недостающими связями.

Поиск новых связей производится с помощью алгоритма, использующего свойства транзитивности связей. Две последовательные положительные связи дают новую положительную связь, сила которой равна произведению сил данных положительных связей. Если в последовательности из двух связей есть связь специализации, то сила результирующей связи равна минимальной из сил этих связей, а вид этой связи определяется видом связей пути. Если в пути присутствует положительная связь, то результирующая связь будет положительной связью. Если же в пути только связи специализации, то и результирующая связь будет специализацией. Для того, чтобы вычислить силу пути, включающего связи разных видов, необходимо в первую очередь находить силы его подпутей, состоящих только из связей синонимии, затем вычислять силы фрагментов, включающих и связи обобщения/специализации. Задача алгоритма — найти максимальное значение силы связи между двумя данными понятиями из онтологических контекстов. Связи, силы которых не преодолевают порогового значения ℓ, отбрасываются. В качестве может быть выбрано число, использовавшееся как пороговое значение на этапе отображения онтологических контекстов на вербальном уровне.

На основе найденных в процессе работы описанного алгоритма позитивных связей и связей специализации между понятиями локальных онтологических контекстов необходимо найти соответствие элементов спецификаций. Для этого вводится концепция слабой онтологической релевантности элементов описания спецификаций:

Определение 1. Элемент I_r спецификаций информационного ресурса онтологически слабо релевантен элементу I_s того же вида (тип, класс, функция, атрибут или др.) спецификаций требований, если I_r связан с понятием C_r, I_s связан с понятием C_s и C_r имеет позитивную ассоциацию с C_s или C_r есть подпонятие C_s:

$$weak\_relevance(I_r,I_s) \iff (\exists C_r,C_s : inst\_of(I_r,C_r) \wedge inst\_of(I_s,C_s) \wedge (positive(C_r,C_s) \vee subconcept(C_r,C_s)))$$

При необходимости более точной онтологической интеграции спецификаций схем Web-сайтов после проведения структурной интеграции контекстов можно использовать концепцию сильной онтологической релевантности элементов спецификаций. Она основана на принадлежности элементов общим классам понятий общей онтологии, в которую интегрированы локальные контексты или классам их подпонятий.

Определение 2. Элемент I_r спецификаций информационного ресурса сильно онтологически релевантен элементу I_s того же вида (тип, класс, функция, атрибут или др.) спецификаций требований, если I_r слабо онтологически релевантен I_s, и I_r является экземпляром по меньшей мере одного онтологического понятия C_r, которые является специализацией (подпонятием) онтологического понятия C_s, имеющего экземпляром I_s (для такой специализации, если типы экземпляров понятий указаны, тип экземпляров C_r должен быть подтипом типа экземпляров C_s), или I_s и I_r принадлежат одному и тому же онтологическому понятию C:

$$tight\_relevance(I_r,I_s) \iff weak\_relevance(I_r,I_s) \wedge (\overline{(\exists C : (inst\_of(I_r,C) \wedge inst\_of(I_s,C) \vee \exists C_r,C_s : (subconcept(C_r,C_s) \wedge inst\_of(I_r,C_r) \wedge inst\_of(I_s,C_s)))))$$

При дальнейшем проектировании электронной библиотеки, перед процедурой композиции ее единой схемы, результаты онтологической интеграции схем используются для предварительного связывания элементов спецификаций схем коллекций и спецификаций требований, являющихся экземплярами классов определенных онтологических понятий.

4 Проектирование конкретизаций и композиция

Метод идентификации фрагментов локальных спецификаций, уточняющих фрагменты спецификаций требований основан на принципах уточнения [7]. Процесс идентификации фрагментов и их композиции состоит из нескольких шагов:

1. На основе информации, выявленной при интеграции онтологических контекстов, производится выбор среди онтологически релевантных типов и классов схем коллекций тех фрагментов, которые могут быть использованы для уточнения соответствующих фрагментов разрабатываемой библиотеки. Фрагменты спецификаций типов генерируются с помощью операции взятия редукта типа. Редукты рассматриваются как образцы для уточнения спецификаций. При этом происходит разрешение различных рассогласований и конфликтов в фрагментах спецификаций коллекций и разрабатываемой персонализованной библиотеки.

2. После идентификации используемых редуктов типов, на основе операций над типами `meet` и `join`, создаются композиции редуктов вовлекаемых типов ресурсов, уточняющие спецификации требований.

Они призваны покрыть типы спецификаций требований такими редуктами.

3. Над классами, соответствующими типам спецификаций требований, строятся взгляды, являющиеся композицией классов, соответствующих вовлеченным типам спецификаций электронных коллекций.

4. Проверка правильности построенной конкретизации может осуществляться с использованием формальных методов доказательства. В случае правильности факта уточнения конкретизирующие типы и взгляды становятся соответственно подтипами типов и подклассами классов спецификаций требований к проектируемой электронной библиотеке.

Возвращаясь к примеру, предположим, что в результате онтологической интеграции схем коллекций патентов в США и Канаде со спецификациями требований к проектируемой библиотеке мы получили некоторое множество онтологически релевантных элементов этих схем элементам того же вида из спецификации требований. Среди них отбираются те классы, типы и их элементы, которые мы будем использовать для реализации спецификаций требований.

Тип `Patent` и класс `patent` спецификаций требований соответствуют типу `CPatPage` и классу `cPatPage` спецификаций канадского сайта. Почти все атрибуты типа `Patent` нашли соответствие в атрибутах типа `CPatPage`, исключая атрибуты `descr` и `claims`. Атрибуту `claims` соответствует атрибут `Claims` из другого типа страниц (`ClaimsPage`) данной схемы, ссылка на который присутствует в типе `CPatPage`. Атрибут `descr` не может иметь реализации на основе элементов схемы данного сайта. Выпишем пары соответствующих друг другу элементов в схемах `PatentLibrary` и `CanadaScheme`:

`PatentLibrary`	`CanadaScheme`
`Patent`	`CPatPage`
`Patent.title`	`CPatPage.Title`
`Patent.inventors`	`CPatPage.Inventors`
`Patent.category`	`CPatPage.InterClass`
`Patent.country`	`CPatPage.PriorCountry`
`Patent.regDate`	`CPatPage.FilingDate`
`Patent.abstract`	`CPatPage.Abstract`
`Patent.descr`	-
`Patent.claims`	`ClaimsPage.Claims`
`patent`	`cPatPage`

Для сайта с американскими патентами тип `Patent` и класс `patent` спецификаций требований соответствуют типу `USPatPage` и классу `usPatPage` спецификаций сайта. Не может быть реализован лишь один атрибут `country`. Пары соответствующих выбранных элементов в схемах `PatentLibrary` и `USAScheme` таковы:

`PatentLibrary`	`USAScheme`
`Patent`	`USPatPage`
`Patent.title`	`USPatPage.Title`
`Patent.inventors`	`USPatPage.Inventors`
`Patent.category`	`USPatPage.INClass`
`Patent.country`	-
`Patent.regDate`	`USPatPage.Filed`
`Patent.abstract`	`USPatPage.Abstract`
`Patent.claims`	`USPatPage.Claims`
`Patent.descr`	`USPatPage.Descr`
`patent`	`usPatPage`

Разрешение структурных конфликтов между спецификациями ресурсов и требований производится в процессе сопоставления онтологически релевантных путей. Пути должны выбираться минимальными, то есть внутри них не должно существовать релевантных подпутей. Поиск релевантных путей ведется с помощью применения специальных правил. Одно из таких правил используется в нашем примере:

Правило 1. Пути релевантны, если типы, являющиеся конечными узлами пути онтологически релевантны, путь со стороны спецификаций требований состоит из одного однозначного атрибута, путь со стороны ресурса состоит из однозначных атрибутов и связей обобщения и заканчивается однозначным атрибутом.

На основании отношения онтологической релевантности и разрешения структурных конфликтов идентифицируются общие редукты спецификации типа требований и типа коллекции. Они и определяют фрагмент типа коллекции, который может быть использован для конкретизации типа требований. Для общих редуктов строятся конкретизирующие редукты, в которых элементы спецификаций требований приводятся в соответствие с элементами спецификаций информационных ресурсов. Тем самым удается избавиться от структурных конфликтов и других рассогласований между спецификациями интегрируемых типов. В примере большинство атрибутов типа `Patent` входят в общий редукт: они получают свои значения напрямую от соответствующих атрибутов типа `CPatPage` или `USPatPage` в зависимости от того, из какой коллекции почерпнуты данные. Исключение составляют атрибуты `descr` и `country`, не имеющие реализаций для разных схем, и атрибут `claims` из-за структурного конфликта. Для него в конкретизирующем редукте может быть построена функция, разрешающая структурный конфликт между схемами `PatentLibrary` и `CanadaScheme` в соответствии релевантным путем, обнаруженным по вышеупомянутому правилу:

```
f_claims: {in: function;
  params: {+c/CPatPage,
           -return/Patent.claims}
{{ return = c.ToClaims.Claims}} } }
```

Атрибут `descr` не реализован на канадском сайте, его следует инициализировать значением `none` для объектов из этого ресурса. Атрибут `country` для объектов из сайта американских патентов можно реализовать значением `"United States"`, так как он не реализован на американском сайте. Заметим также, что тип атрибута `Claims` из спецификаций схемы `USAScheme` уточняет тип объединения атрибута `claims` спецификации требований, поэтому рассогласования в типах атрибутов в данном случае не возникает.

Проектирование конкретизирующих типов, реализующих соответствующие типы спецификаций требований, основано на композиции спецификаций типов коллекций

с помощью операций meet и join. С учетом проведенного выше анализа релевантных путей и редуктов составляется формула, которая позволит конструировать тип, конкретизирующий тип в спецификации требований, избегая обнаруженные конфликты. В нашем примере композиция CTPatent для типа Patent будет состоять из объединения типов CPatPage и ClaimsPage, для наибольшего покрытия типа Patent, и пересечения результата с типом USPatPage:

$$CTPatent \approx (CPatPage \sqcup ClaimsPage) \sqcap USPatPage$$

После выполнения этих операций в типе CTPatent не будет хватать атрибутов descr и country, необходимых для конкретизации. Однако уже известен путь их компенсации значениями по умолчанию для тех коллекций, в которых они не реализуются. Поэтому конкретизирующий тип дополняется этими атрибутами. Результирующий тип CTPatent становится подтипом типа спецификации требований Patent, так как покрывает тип спецификации требований.

Процесс проектирования завершается созданием над классами ресурсных коллекций взглядов, которые реализуют классы спецификаций требований и становятся подклассами соответствующих классов, определенных в спецификации требований. Для случая композиции классов двух сайтов в электронную библиотеку патентов определяется взгляд vPatent над ресурсными классами cPatPage, claimsPage и usPatPage. Операции join типов соответствует пересечение соответствующих классов объектов, операции meet — объединение. Поэтому для получения класса, соответствующего объединению типов CPatPage и ClaimsPage, обозначенному во взгляде как тип CanPage, берется пересечение классов cPatPage и claimsPage. Взгляд, соответствующий пересечению типов CanPage и USPatPage, есть объединение полученного класса с классом usPatPage. Во взгляде vPatent определяется функция, описывающая правила формирования класса библиотеки и отображения состояний объектов ресурсных типов в тип библиотеки:

```
{ vPatent;
  in: class;
    metaslot
    prop_view: { in: function;
      params: -returns/vPatent as_class;
      enforcement: on_access;

    {{
      ( { cp/CanPage
          [ title/Title,
            inventors/Inventors,
            regDate/FilingDate,
            category/PriorCountry,
            abstracxt/Abstract,
            claims/Claims ] ||
          cPatPage (cp/CanPage[CPatPage]) &
          claimsPage (cp/CanPage[ClaimsPage])} &
        descr(cp) == none ) |

      ( ( usPatPage(u/USPatPage
          [ title/Title,
            inventors/Inventors,
            category/INClass ]) &^
          cPatPage(u/CPatPage
          [ title/Title,
            inventors/Inventors,
            category/InterClass ]) ) &
        ( usPatPage(u/USPatPage
          [ title/Title,
            inventors/Inventors,
            category/INClass,
            regDate/Filed,
            abstract/Abstract,
            claims/Claims,
            descr/Descr ]) &
          country(u) == 'United States') )
    }}
    end

  superclass: patent;
  instance_senction: CTPatent
}
```

Классы объектов различных коллекций могут содержать своими экземплярами некоторые объекты, которым соответствуют одни и те же сущности реального мира. Такие соответствия должны обнаруживаться в процессе формирования классов библиотеки. Это возможно реализовать с помощью дополнений к предикату, формирующему взгляд. Для объектов, являющихся экземплярами классов разных коллекций, определяются правила их идентификации объектам реального мира в зависимости от внутреннего состояния. В нашем примере таким правилом служит выполнение условия совпадения заголовков патентов, их авторов и категорий в международной классификации. Во взгляде задано условие, определяющее, что в случае обнаружения идентичных объектов в классах cPatPage и usPatPage в класс vPatent войдет только объект из класса cPatPage.

Заключение

В статье изложен метод проектирования персонализованных электронных библиотек над слабоструктурированными источниками информации в Web по спецификациям требований пользователя. Действие метода демонстрируется на примере создания библиотеки над реальными электронными коллекциями, расположенными на Web-сайтах. Показаны этапы разработки библиотеки от составления спецификаций требований до композиции реальных источников информации в электронную библиотеку. Подобные методы и техника проектирования, а точнее, их развитие, станут частью принципов построения персонализованных электронных библиотек в проекте, созхдаваемом по гранту РФФИ.

Библиография

[1] P. Atzeni, G. Mecca , P. Merialdo. *Semistructured and Structured Data in the Web: Going Back and Forth.* In Sigmod Record, Special Issue on the Workshop on the Management of Semistructured Data, 1997

[2] D. O. Briukhov, L. A. Kalinichenko. *Component-Based Information Systems Development Tool Supporting the SYNTHESIS Design Method.* Proceedings of the East European Symposium on "Advances in Databases and Information Systems", Poland, Springer, LNCS No.1475, 1998

[3] D. O. Briukhov, S. S. Shumilov. *Ontology Specification and Integration Facilities in a Semantic Interoperation Framework.* Proc. of the International Workshop ADBIS'95, Springer, 1995

[4] P. Fankhauser, E. J. Neuhold. *Knowledge Based Integration of Heterogeneous Databases.* Integrated Publication and Information Systems Institute (GMD-IPSI), Darmstadt, 1993.

[5] S. Grumbach, G. Mecca. *In Search of the Lost Schema.* In Proceedings of Intern. Conference on Database Theory (ICDT'99), 1999

[6] L. A. Kalinichenko. *SYNTHESIS: the Language for Desription, Design and Programming of the Heterogeneous Interoperable Information Resource Environment.* Institute for Problems of Informatics, Russian Academy of Sciences, Moscow, 1995

[7] L. A. Kalinichenko. *Compositional Specification Calculus for Information Systems Development.* In Proceedings of the East-West Symposium on Advances in Databases and Information Systems (ADBIS'99), Maribor, Slovenia, September 1999, Springer Verlag, LNCS, 1999

[8] О. Мачульский, М. Осипов, Л. А. Калиниченко. *Отображение модели данных XML в объектную модель языка СИНТЕЗ.* Первая национальная конференция "Электронные библиотеки: перспективные методы и технологии, электронные коллекции", 1999

[9] G. Mecca, P. Merialdo, P. Atzeni, V. Crescenzi. *The Araneus Guide to Web-Site Development.* Araneus Project Working Report, AWR-1-99 (version 1.0 - March, 9, 1999)

[10] G. Salton, C. Buckley. *Term-Weighting Approaches in Automatic Text Retrieval.* Readings in Information Retrieval under edition of K. S. Jones and P. Willett, Morgan Kaufmann Publishers Inc., 1997.

[11] *United States Patent.* [http://164.195.100.11/netahtml/search-bool.html]

[12] *Canadian Patent Database.* [http://Patents1.ic.gc.ca/intro-e.html]

[13] A. Hopmann, et. al., *Web Collections using XML.* [http://www.w3.org/pub/WWW/Member/9703/XMLsubmit.html]