ОТОБРАЖЕНИЕ ЯЗЫКА СИНТЕЗ В RDFS

Тюрин И.Н., Брюхов Д.О., Калиниченко Л.А.

Институт Проблем Информатики Российской Академии Наук
44-2, ул. Вавилова, Москва, Российская Федерация, 117333
e-mail: {turin, brd, leonidk}@synth.ipi.ac.ru

Аннотация

В данной статье[1] рассматривается вопрос отображения языка СИНТЕЗ в RDFS . Определяются принципы отображения и расширения RDFS для описания языка СИНТЕЗ, модель данных которого используется в качестве канонической для посредника неоднородных информационных коллекций.

1 Введение

С развитием WWW встал вопрос стандартизации внешнего представления метаданных. Архивы статей, публикаций, каталоги веб-сайтов, каталоги программного обеспечения, поисковые серверы – это неполный перечень интернет-служб, нуждающихся в стандартном представлении метаданных для обмена, синхронизации и пополнения своих баз. Очевидно, что метаданные бывают различных уровней: есть метаданные, описывающие конкретные ресурсы, например, Dublin Core [1]; метаданные более высокого уровня описывают схему самого Dublin Core, его компоненты – и так далее. В дальнейшем, говоря о метаданных, мы будем подразумевать «данные о данных» в общем случае, безотносительно их уровня.

Последние несколько лет консорциум W3C развивает стандарт языка разметки XML (eXtended Markup Language) [3] в качестве основного носителя информации в Сети. С помощью данного стандарта строятся словари (DTD или XML Schemas [9]) для передачи более специализированной информации. Для спецификации метаданных ресурсов Интернета предлагаются стандарты RDF (Resource Description Framework) [7] и RDFS (Resource Description Framework Schema) [2].

В RDF метаданные представляются в виде ориентированного графа, в котором узлы ассоциируются с ресурсами, а дуги – с их свойствами. Под ресурсом понимается любой ресурс в Интернете, имеющий URI (Uniform Resource Identifier), под свойством понимается некоторая характеристика ресурса. RDFS является надстройкой над RDF и представляет собой язык для описания схем данных, предназначенный для моделирования иерархий классов, свойств и других примитивов. Наличие XML сериализации языка позволяет обмениваться RDFS документами как любыми другими Web данными.

В настоящее время RDF и RDFS используются при создании веб-каталогов, например, DMOZ (http://www.dmoz.org) и серверов приложений, в частности, Zope (http://www.zope.org).

В лаборатории композиционных методов проектирования информационных систем ИПИ РАН разрабатывается архитектура посредника неоднородных коллекций [5], который позволяет работать с распределенными неоднородными коллекциями данных как с интегрированной объектно-ориентированной базой данных. В рамках посредника определяется метаинформация предметной области, которую представляет данный посредник. В качестве канонической модели данных посредника используется язык СИНТЕЗ [6].

---

[1] Данная работа выполняется в рамках проектов, поддерживаемых РФФИ (гранты №00-07-90086, 01-07-90084).

Задачей данной статьи является отображение языка СИНТЕЗ в RDFS. Круг задач, для решения которых используется такое преобразование, включает:

- Получение спецификации базы метаинформации посредника в существующем веб-стандарте;
- Обмен метаинформацией между различными посредниками;
- Просмотр (browsing) базы метаинформации, используя существующие инструменты;
- Регистрация коллекций в посреднике.

2 СИНТЕЗ

Язык СИНТЕЗ является гибридным языком для описания объектных и слабо-структурированных данных. Важным понятием языка СИНТЕЗ является абстрактный тип данных (АТД). АТД определяет интерфейс объекта данного типа и его свойства. Определение типа содержит такие понятия как атрибуты, функции и инварианты. С каждым атрибутом может быть связан метаслот, задающий дополнительные характеристики данного атрибута, такие как: инварианты, начальные значения и др.

Ниже приводится пример спецификации АТД на языке СИНТЕЗ:

```
{ Organization;
   in: type;
   name: string;
   in_state: State;
   staff: { set; type_of_element: Person }
      metaslot
         in: i_total;
         inverse: Person.works_for
      end;
   staffinv: {
      in: invariant;
      {{ staff.cardinal < 1000 }}}
};
```

В данном примере определен тип Organization, атрибутами которого являются: name (тип string), in_state (тип State), staff (множество элементов типа Person). Определен инвариант staffinv, и с атрибутом staff связан метаслот, специфицирующий дополнительные свойства данного атрибута.

3 RDF/RDFS

RDF ориентирован на достижение интероперабельности приложений, которые обмениваются информацией в Интернете. RDF базируется на XML, его предназначением является описание веб-ресурсов для последующей автоматической обработки. Описание ресурса задается путем указания его свойств и их значений. Например:
Ресурс: http://www.ipi.ac.ru/synthesis/staff/turin/
Свойство: works_for
Значение: IPI RAS

Данный пример записывается в виде утверждения на RDF (RDF statement), как

```
<rdf:Description rdf:about="http://www.ipi.ac.ru/synthesis/staff/turin/">
   <s:works_for>IPI RAS</s:works_for>
</rdf:Description>
```

(в некоторых случаях мы будем опускать квалификатор "rdf:", подразумевая, что используем RDF namespace в качестве XML пространства имен по умолчанию)

Любой ресурс, описываемый в RDF, должен иметь URI (Uniform Resource Identifier). RDF не накладывает ограничений на род описываемых ресурсов, ресурсом может выступать любой объект, на который можно сослаться, например, им может быть веб-сайт, а может быть описание печатного издания.

Чтобы точнее описывать ресурсы, возникает необходимость указать некоторые факты относительно ресурсов, свойств и их значений. То есть определить то, как должны описываться ресурсы того или иного рода. RDF Schema предназначена для создания систем типов для последующего использования в RDF спецификациях. В ней вводятся такие ресурсы, как rdfs:Resource, rdfs:Class, rdfs:ConstraintProperty и свойства, как rdfs:subClassOf, rdfs:subPropertyOf, rdfs:domain, rdfs:range, которые используются для описания свойств других RDF описаний.

```
<rdfs:Class ID="Organization">
 <rdfs:subClassOf
  rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdf:Property ID="name">
 <rdfs:domain rdf:resource="#Organization"/>
 <rdfs:range rdf:resource="#String"/>
</rdf:Property>
```

Данный пример описывает тип Organization, как класс в терминах RDFS и связывает с этим классом свойство name. Значением этого свойства может быть экземпляр класса String. При описании показывается, что Organization является подклассом Resource – элемента RDFS.

4 Принципы отображения языка СИНТЕЗ в RDFS

RDF Schema предоставляет основную модель для определения классов и их свойств, на основе которой строится система типов для конкретной предметной области. Однако, не всегда механизмы RDFS в полной мере обеспечивают потребности решения той или иной задачи, и приходится строить некоторое расширение RDFS для работы со специфическими конструкциями. Заранее следует отметить, что расширение само описывается в терминах RDFS, и поэтому не портит синтаксическую и семантическую форму понятий RDFS. Примером расширения RDFS может служить расширение RDFS для OIL (Ontology Inference Layer), которое рассмотрено в [4]. При этом расширении RDFS был предложен ряд методов для обработки специфических ситуаций, аналоги которых с той или иной степенью сходства встречаются при расширении RDFS СИНТЕЗа.

При построении RDF Schema СИНТЕЗа (включая расширение) необходимо придерживаться следующих правил:

1. Если конструкция СИНТЕЗа совпадает с конструкцией RDFS или родственна ей, то при построении RDFS строится подкласс, дополняющий последнюю до соответствия конструкции СИНТЕЗа;

2. Для конструкций СИНТЕЗа, которые могут быть представлены в RDFS только путем расширения, строится соответствующее расширение RDFS;
3. Ряд конструкций СИНТЕЗа (например, логические формулы) нерационально представлять в RDFS, так как RDFS не имеет сходных понятий. Такие конструкции представляются в RDFS в синтаксисе СИНТЕЗа, используя механизм представления инородных данных XML (CDATA).

4.1 Отображение родственных конструкций

При отображении СИНТЕЗа в RDFS практически не происходит замены понятий СИНТЕЗа понятиями RDFS, так как элементы СИНТЕЗа определены строже. В ряде случаев для них можно построить подкласс элемента RDFS:

СИНТЕЗ:

```
{ string;
   [length: <целое без знака>]
}
```

RDFS:

```
<rdfs:Class rdf:ID="String">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdfs:Class>

<rdf:Property rdf:ID = "length">
   <rdfs:domain rdf:resource = "#String" />
   <rdfs:range rdf:resource = "#UInteger" />
</rdf:Property>
```

В примере выше для определения класса String, соответствующего встроенному типу данных СИНТЕЗа string (тип символьной строки), мы строим подкласс RDF класса http://www.w3.org/2000/01/rdf-schema#Literal, связывая с ним свойство length типа UInteger (определяет тип целого без знака).

Таким образом, мы можем перевести описание типа string из СИНТЕЗа в RDFS, например, для спецификации атрибута attr1, имеющего ограничение длины 50 символов:

СИНТЕЗ:

```
…
attr1: { string; length: 50 }
…
```

RDFS:

```
<syn:attribute>
   <syn:id>attr1</syn:id>
   <syn:attValue rdf:resource="#String">
      <syn:length>50</syn:length>
   </syn:attValue>
</syn:attribute>
```

4.2 Расширение RDFS

В ряде других случаев приходится расширять RDFS путем ввода элементов, не являющихся ни частью RDFS, ни частью СИНТЕЗа. Например:

СИНТЕЗ:

```
{ <идентификатор АТД>;
…
<идентификатор атрибута>: <значение>;
…
}
```

RDFS:

```
<rdfs:Class rdf:ID = "ADT">
   <rdfs:subClass rdf:resource = "#Aval" />
</rdfs:Class>

<rdf:Property rdf:ID = "_hasAttribute">
   <rdfs:range rdf:resource = "#_Attribute" />
   <rdfs:domain rdf:resource = "#ADT" />
</rdf:Property>

<rdfs:Class rdf:ID = "Attribute" />

<rdf:Property rdf:ID = "id">
   <rdfs:range rdf:resource = "#Literal" />
   <rdfs:domain rdf:resource = "#ADT" />
   <rdfs:domain rdf:resource = "#Attribute" />
</rdf:Property>

<rdf:Property rdf:ID = "attValue">
   <rdfs:range = "#Aval" />
   <rdfs:domain = "#Attribute" />
</rdf:Property>
```

Промежуточное свойство _hasAttribute вводится для того, чтобы не нарушать принцип RDF: ресурс и значение должно связывать некоторое свойство. В данном случае экземпляр класса Attribute должен быть привязан к экземпляру класса ADT. Аналогичный подход используется в [4] при отображении операций над классами онтологий.

4.3 Представление инородных для RDFS данных

Одним из примеров конструкции СИНТЕЗа, которую нельзя (неоправданно сложно) отобразить в RDFS, является запись формулы. RDFS не обладает механизмами для описания выражений. Поэтому расширение RDFS путем встраивания таких механизмов не имеет смысла, так как RDF-ориентированные приложения будут вынуждены игнорировать подобную информацию, а СИНТЕЗ-ориентированные (например, навигатор по базе метаинформации) должны будут проделать работу по обратному преобразованию RDFS конструкций в СИНТЕЗ конструкции. Мы предлагаем использовать XML тип CDATA для описания формул:

СИНТЕЗ:

```
{ staff.cardinal < 1000 }
```

RDFS:

```
<syn:formula>
   <![ CDATA[ staff.cardinal < 1000 ]]>
</syn:formula>
```

Данный метод рекомендуется в [8], как один из возможных методов отображения сложных записей.

4.4 Отображение фиксированных атрибутов

В 4.2 при описании метода расширения RDFS мы показали, как отображаются атрибуты АТД, идентификаторы которых определяются при спецификации АТД. Однако, АТД и другие элементы СИНТЕЗа обладают набором фиксированных атрибутов, идентификаторы которых предопределены. Такие атрибуты можно описать как свойства (Property) соответствующих классов без расширения RDFS. Рассмотрим этот метод на примере спецификации типа функции.

СИНТЕЗ:

```
{ <function id>
   in: function | predicate …
…
}
```

В данном примере мы имеем не только фиксированный атрибут in, но и фиксированный набор значений (точнее говоря, атрибут in может иметь дополнительные значения помимо function и predicate, но одно из последних должно быть указано обязательно).

RDFS:

```
<rdf:Property rdf:ID = "in">
   <rdfs:domain rdf:resource = "#Function" />
   <rdfs:range rdf:resource = "#FunctionState" />
</rdf:Property>

<rdfs:Class rdf:ID = "FunctionState" />
<FunctionState rdf:ID = "function" />
<FunctionState rdf:ID = "predicate" />
```

Мы опустили спецификацию дополнительных значений для атрибута in ради упрощения примера.

4.5 Пример спецификации АТД на RDFS

Пример АТД Organization на СИНТЕЗе мы взяли из раздела 2.

RDFS:

```
<syn:ADT>
   <syn:id>Organization</syn:id>
   <syn:_hasAttribute>
      <syn:Attribute>
         <syn:id>name</syn:id>
         <syn:attValue rdf:resource = "#String" />
      </syn:Attribute>
      <syn:Attribute>
         <syn:id>in_state</syn:id>
```

```
         <syn:attValue rdf:resource = "#State" />
      </syn:Attribute>
      <syn:Attribute>
         <syn:id>staff</syn:id>
         <syn:attValue rdf:resource = "#Set">
            <syn:typeofelement rdf:resource = "#Person" />
         </syn:attValue>
         <syn:_hasMetaslot>
            <syn:Metaslot>
               <syn:in>
                  <syn:CategoryAttribute rdf:resource ="#i_total" />
               </syn:in>
               <syn:inverse>
                  <syn:path>
                     <syn:prefix>Person</syn:prefix>
                     <syn:postfix>works_for</syn:postfix>
                  </syn:path>
               </syn:inverse>
            </syn:Metaslot>
         </syn:_hasMetaslot>
      </syn:Attribute>
      <syn:Attribute>
         <syn:id>staffinnv</syn:id>
         <syn:attValue rdf:resource = "#Invariant" />
         <syn:formula>
            <![ CDATA[ staff.cardinal < 1000 ]]>
         </syn:formula>
      </syn:Attribute>
   </syn:_hasAttribute>
</syn:ADT>
```

4.6 Проблемы отображения

При построении расширения RDFS для описания языка СИНТЕЗ ряд проблем, вызванных некоторыми ограничениями RDFS, остается нерешенным. Наиболее важным кажется ограничение RDF Schema на количество возможных элементов rdfs:range для свойства. Это можно обойти, каждый раз создавая мнимый суперкласс для объединения различных классов между собой (наличие множественного наследования позволяет это делать), но применение этого способа будет перегружать схему тяжелыми для понимания связями. Другой проблемой при построении отображения является нестандартность объектной модели RDF Schema: описание свойств в терминах того, какие классы они могут связывать, не согласуется с описанием классов в терминах свойств, как это принято в модели СИНТЕЗа. То есть семантика некоторых связей является не совсем ясной для понимания, например, как следует рассматривать набор из нескольких rdfs:domain для свойства, следует ли понимать, что свойство, определенное таким образом, должно обладать одинаковой семантикой для всех классов или же это просто метод сокращения записи.

Как отмечено в 4.1, мы использовали достаточно слабую интеграцию понятий СИНТЕЗа с понятиями RDFS. В [4] приводится пример отображения онтологий OIL в RDFS. В силу сравнительной простоты модели OIL авторам удалось построить более тесные взаимосвязи между элементами OIL и RDFS. При этом в качестве дополнительных проблем RDF Schema были выявлены:

- наличие запрета на циклы при указании отношений класс/подкласс (авторы считают, что подобное ограничение является лишним, так как уменьшает количество моделей, которые можно представить в RDFS);

- принципиально различные способы указания принадлежности ресурса некоторому классу, что делает запись труднопонимаемой:

```
<SomeClass rdf:ID="abc">…</SomeClass>
<rdf:Description about="abc"><rdf:type="#SomeClass">… </rdf:Description>
```

- способ наследования rdfs:range и rdfs:domain при использовании механизма наследования свойств (rdfs:subPropertyOf).

Заключение

Данная работа показала возможность использования существующих веб-стандартов для представления в них языка СИНТЕЗ. Было построено отображение языка СИНТЕЗ в RDFS. В процессе построения был выявлен ряд методов расширения RDFS.

Список литературы

[1] Beckett D., Miller E., Brickley D. An XML Encoding of Simple Dublin Core Metadata. http://dublincore.org/documents/2001/04/11/dcmes-xml/. Proposed Recommendation.

[2] Brickley D., Guha R.V. Resource description framework (RDF) schema specification. http://www.w3.org/TR/PR-rdf-schema. W3C Proposed Recommendation. Technical report, W3C, 1999.

[3] Extensible Markup Language (XML) 1.0. http://www.w3.org/TR/2000/REC-xml-20001006. W3C Recommendation. October, 2000.

[4] Broekstra J., Klein M., Decker S., Fensel D., Horrocks I. Adding formal semantics to the Web Building on top of RDF Schema. OIL papers (http://www.ontoknowledge.org/oil/papers.shtml), July 2000.

[5] Kalinichenko L.A., Briukhov D.O., Skvortsov N.A., Zakharov V.N. Infrastructure of the subject mediating environment aiming at semantic interoperability of heterogeneous digital library collections. 2nd Russian Conference "DIGITAL LIBRARIES: Advanced Methods and Technologies, Digital Collections, September 26-28, 2000, Protvino

[6] Kalinichenko L.A. SYNTHESIS: the language for description, design and programming of the heterogeneous interoperable information resource environment. Institute for Problems of Informatics of the Russian Academy of Sciences, Moscow, 1995

[7] Lassila O., Swick R. Resource description framework (RDF) model and syntax specification. http://www.w3.org/TR/REC-rdf-syntax . W3C Recommendation. Technical report, W3C, 1999.

[8] Staab S., Erdmann M., Maedche A., Decker S. An Extensible Approach for Modeling Ontologies in RDFS. In Proc. of the International Workshop on the Semantic Web 2000, Lisbon, Portugal, April 2000.

[9] XML Schema Language Part 0: Primer. http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/. W3C Recommendation. May, 2001.

[10] XML Schema Language Part 1: Structures. http://www.w3.org/TR/2001/PR-xmlschema-1-20010330/. W3C Proposed Recommendation. March, 2001.

[11] XML Schema Language Part 2: Datatypes. http://www.w3.org/TR/2001/PR-xmlschema-2-20010330/. W3C Proposed Recommendation. March, 2001.

MAPPING OF THE SYNTHESIS LANGUAGE INTRO RDFS

Turin I.N., Briukhov D.O., Kalinichenko L.A.

Institute for Problems of Informatics RAS,

Moscow

e-mail: {turin, brd, leonidk}@synth.ipi.ac.ru

In this paper the issues of the SYNTHESIS language mapping into RDFS is considered. The paper discusses principles of the language mapping and constructing RDFS extensions suitable for refining the SYNTHESIS language data model of which is used as a canonical one for the mediator of heterogeneous information sources.