ОТОБРАЖЕНИЕ ТИПОВ ДАННЫХ XML SCHEMA В ТИПЫ ЯЗЫКА СИНТЕЗ

Брюхов Д.О., Тюрин И.Н.
Институт Проблем Информатики Российской Академии Наук
117333, Российская Федерация, Москва, ул. Вавилова 44-2
e-mail: {brd, turin}@ipi.ac.ru

Статья[1] посвящена построению отображения встроенных типов данных XML Schema в типы языка СИНТЕЗ.

MAPPING XML SCHEMA DATA TYPES INTO SYNTHESIS TYPES

Briukhov D.O., Tyurin I.N.
Institute for Problems of Informatics RAS, Moscow, Russia
e-mail: {brd,turin}@ipi.ac.ru

In this paper the issues of XML Schema data types mapping to SYNTHESIS data types are considered. The mapping is used for loading the XML Schema specifications into the mediator's metainformation repository in the collection registration process.

1 Введение

В лаборатории композиционных методов проектирования информационных систем ИПИ РАН разрабатывается архитектура посредника неоднородных информационных коллекций [3], который позволяет работать с распределенными неоднородными коллекциями данных как с интегрированной коллекцией информации.

Посредник поддерживает процесс систематической регистрации и классификации коллекций, содержит унифицированные онтологические данные и метаинформацию для улучшения обнаружения и композиции существующих коллекций [1].

Регистрация коллекции - это процесс взаимодействия провайдера цифровой коллекции с предметным посредником во время операционной фазы посредника, заключающийся в контекстуализации коллекции в посреднике (согласовании ее понятийного и терминологического контекста с контекстом посредника), представления классов коллекции как материализованных взглядов над классами посредника, генерации адаптеров.

[1] Данная работа выполняется в рамках гранта РФФИ №01-07-90084.

В рамках посредника определяется метаинформация предметной области, которую представляет данный посредник. В качестве канонической модели данных посредника используется язык СИНТЕЗ [2].

Последние несколько лет консорциум W3C развивает стандарт языка разметки XML (eXtended Markup Language) в качестве основного носителя информации в Сети. С помощью данного стандарта строятся словари (DTD или XML Schemas) для передачи более специализированной информации.

XML Schema [4] определяет конкретный словарь XML - конкретный набор элементов разметки (тэгов) и ограничения, связанные с ними. Например, существует XML Schema для определения тэгов HTML, MathML (язык разметки математических выражений), DAML+OIL (язык описания онтологий) и других языков разметки.

В данной статье рассматривается отображение типов данных XML Schema в типы данных языка СИНТЕЗ. Данное отображение используется при регистрации коллекций для загрузки их спецификаций, представленных в XML Schema, в базу метаинформации посредника.

2 Система типов XML Schema

В языке XML Schema определены сложные (complex) и простые (simple) типы данных.

2.1 Простые типы данных

Простые типы данных XML Schema делятся на две группы: примитивные (primitive) и выводимые (derived). К примитивным типам данных относятся такие типы, как string, boolean, double, decimal и др. Выводимые типы строятся на основе примитивных типов. Ряд выводимых типов является предопределенным (например, integer выводится из decimal)

Любой простой тип в XML Schema определяется множеством своих значений и множеством лексических представлений (lexical representation). Операции над типами не определяются. Построение новых простых типов осуществляется с помощью одного из следующих методов:

- выведение по ограничению (deriving by restriction);
- выведение по списку (deriving by list);
- выведение по объединению (deriving by union).

Выведение по ограничению строится с помощью ограничивающих фасетов (constraining facets). Также отметим тот факт, что в XML Schema нет встроенных типов множеств, вместо этого вводятся понятия выведения по списку и объединению.

Приведем примеры построения новых типов:

- по ограничению

```
<xsd:simpleType name="over12">
  <xsd:restriction base="xsd:decimal">
    <xsd:minInclusive value="13" />
  </xsd:restriction>
</xsd:simpleType>
```

тип over12 выводится на основе примитивного типа decimal заданием ограничения снизу

- по списку

```
<xsd:simpleType name='listOfInteger'>
  <xsd:list itemType='integer'/>
</xsd:simpleType>
```

тип listOfInteger является списком элементов типа integer

- по объединению

```
<xsd:simpleType name="clothingsize">
  <xsd:union>
    <xsd:simpleType>
      <xsd:restriction base="integer" />
    </xsd:simpleType>
    <xsd:simpleType>
      <xsd:restriction base="string" />
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
```

значением элемента типа clothingsize может быть как целое число, так и строка

2.2 Сложные типы данных

В отличие от простых типов сложные типы позволяют указывать дополнительные атрибуты и внутренние элементы. Ниже приведен пример определения сложного типа на XML Schema:

```
<complexType name=”Employee”>
  <xsd:element name=”name” type=”xsd:string” />
  <xsd:element name=”age” type=”over17” />
  <xsd:element name=”salary” type=”xsd:integer” />
  <xsd:attribute name=”department” type=”xsd:string” />
 </complexType>
```

Атрибуты (xsd:attribute) введены в XML Schema для упрощения записи, типом атрибута может быть только простой тип данных. Всегда возможна замена спецификации атрибута на спецификацию элемента.

Для набора элементов можно задавать модель группы (model group), которая накладывает ограничение на форму записи данных элементов в типе:

- строгая последовательность (sequence) – элементы должны следовать в порядке, заданном в спецификации;
- произвольная последовательность – элементы могут следовать в произвольном порядке;
- выбор (choice) – элемент должен соответствовать только одной спецификации из группы.

3 Система типов языка СИНТЕЗ

В основе объектной модели лежит понятие абстрактного типа данных (АТД), служащее для описания неизменчивых типов данных любой природы. Описание абстрактного типа данных инкапсулирует спецификации атрибутов, ассоциаций и операций типа. Операции типов описываются типом функции. Наряду с АТД в СИНТЕЗе определён набор базовых типов, таких как integer, string, boolean, list, enum и др.

4 Отображение простых типов данных XML Schema в типы данных языка СИНТЕЗ

4.1 Отображение предопределенных типов

Среди предопределенных типов XML Schema мы выделяем типы, для которых существуют эквивалентные типы СИНТЕЗа, и для которых не существует эквивалентных типов. Для первой группы ниже приведена таблица соответствия типов:

Типы XML Schema	**Типы СИНТЕЗа**
anySimpleType	Tbuilt_in
string	string
boolean	boolean
float	float
double	double
duration	interval
dateTime	time
integer	integer
int	long
unsignedInt	unsigned long
short	short
unsignedShort	unsigned short

Для каждого типа из второй группы строится соответствующий ему АТД СИНТЕЗа. Приведем пример спецификации АТД соответствующего типу anyURI XML Schema.

```
{ xmlAnyURI;
  in: type;
  value: string;
}
```

4.2 Отображение типов, выводимых по ограничению

Данные типы XML Schema отображаются в АТД СИНТЕЗа. Поскольку в СИНТЕЗе АТД не могут являться подтипами встроенных типов данных, при отображении мы строим АТД, в атрибуте value которого хранится множество значений отображаемого типа.

Ограничение (restriction) отображается в тип атрибута value, а фасеты отображаются в инварианты для атрибута value. Например:

XML Schema:

```
<xsd:simpleType name="over17">
  <xsd:restriction base="xsd:integer">
  <xsd:minInclusive value="18"/>
  </xsd:restriction>
</xsd:simpleType>
```

СИНТЕЗ:

```
{Over17;
  in: type;
  value: integer;
    metaslot
      inv: {in: predicate, invariant;
           {{all o/Over17 ( o.value>=18 ) }}
    end
}
```

Исключением из этого правила является отображение фасета enumeration, который удобнее отображать с помощью встроенного типа СИНТЕЗа enum:

XML Schema:

```
<xsd:simpleType name="enumerationHeight">
  <xsd:restriction base="string">
    <xsd:enumeration value="short"/>
    <xsd:enumeration value="medium"/>
    <xsd:enumeration value="tall"/>
  </xsd:restriction>
</xsd:simpleType>
```

СИНТЕЗ:

```
{EnumerationHeight;
      in: type;
      value: {enum; enumlist: {‘short’, ‘medium’, ‘tall’
}}
}
```

4.3 Отображение типов, выводимых по списку

Данные типы XML Schema отображаются в АТД СИНТЕЗа, в атрибуте value которого хранится множество значений отображаемого типа. Типом атрибута value является тип СИНТЕЗа list. Тип элементов списка (type_of_element) соответствует ограничению по списку в XML Schema (itemType).

XML Schema:

```
<xsd:simpleType name=”listOfString”>
  <xsd:list itemType='xsd:string'/>
</xsd:simpleType>
```

СИНТЕЗ:

```
{ListOfString;
  in: type;
  value: { list; type_of_element: string }
}
```

4.4 Отображение механизма ограничения по объединению

Данные типы XML Schema отображаются в АТД СИНТЕЗа, в атрибуте value которого хранится множество значений отображаемого типа. Типом атрибута value является тип СИНТЕЗа union.

XML Schema:

```
<xsd:simpleType name="clothingsize">
  <xsd:union>
     <xsd:simpleType>
       <xsd:restriction base='integer'/>
     </xsd:simpleType>
     <xsd:simpleType>
       <xsd:restriction base='string'/>
     </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
```

СИНТЕЗ:

```
{Clothingsize;
  in: type;
  value: {union;
     type_of_label: string;
     label1: integer;
     label2: string;
  }
}
```

5 Отображение сложных типов данных XML Schema в типы данных СИНТЕЗ

Мы будем отображать сложные типы данных XML Schema в АТД языка СИНТЕЗ. Рассмотрим отображение на примере:

```
<complexType name=”Employee”>
  <xsd:attribute name=”company” type=”xsd:string” />
  <xsd:element name=”name” type=”xsd:string” />
 <xsd:sequence>
  <xsd:element name=”age” type=”over17” />
  <xsd:element name=”salary” type=”xsd:integer” />
  <xsd:element name=”department” type=”xsd:string” />
 </xsd:sequence>
 <xsd:choice>
  <xsd:element name=”addressRU” type=”AddressRU” />
  <xsd:element name=”addressOther” type=”xsd:string” />
 </xsd:choice>
</complexType>
```

В данном примере мы определили тип Employee, содержащий атрибут company, элемент name, строгую последовательность элементов и выбор. Элементы age, salary и department должны идти в том порядке, в котором определены в типе Employee. Выбор означает, что только один из элементов addressRU и addressOther может быть использован в описании экземпляра типа.

5.1 Отображение атрибутов

Атрибуты сложных типов отображаются в атрибуты АТД.

```
{ Employee;
   in: type;
   company: string;
   ...
}
```

5.2 Отображение элементов произвольной последовательности

Элементы сложных типов отображаются в атрибуты АТД.

```
{ Employee;
   in: type;
   name: string;
   ...
 }
```

5.3 Отображение элементов строгой последовательности

Элементы строгой последовательности сложных типов отображаются в атрибуты АТД. Для отображения порядка элементов в АТД вводится дополнительный атрибут _sequenceOrder, в котором указываются идентификаторы элементов в том порядке, в котором они описаны в спецификации на XML Schema:

```
{ Employee;
    in: type;
    ...
    age: Over17;
    salary: integer;
    department: string;
    _sequenceOrder: { age, salary, department };
    ...
}
```

5.4 Отображение элементов выбора

Для выбора элементов вводится дополнительный атрибут, типом которого является union атрибутов, представленных в соответствующем xsd:choice:

```
{ Employee;
    in: type;
    ...
    addressRU_otherAddress:
      { union;
        type_of_label: string;
        addressRU: AddressRU;
        addressOther: string
      }
}
```

6 Заключение

В данной статье представлены правила отображения типов данных XML Schema в СИНТЕЗ.

Литература

[1] Briukhov, D.O., Kalinichenko, L.A., Skvortsov, N.A. Information sources registration at a subject mediator as compositional development. In Proceedings of the Fifth East European Symposium on Advances in Databases and Information Systems (ADBIS'01), Springer-Verlag, 2001, pp. 70-83.

[2] Kalinichenko L.A. SYNTHESIS: the language for description, design and programming of the heterogeneous interoperable information resource environment, Moscow, 1995.
[3] 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
[4] XML Schema Language Part 2: Datatypes. http://www.w3.org/TR/2001/PR-xmlschema-2-20010330/. W3C Proposed Recommendation. March, 2001.