



Описание структуры документа



Лекция

XML Schema

XML Schema — язык описания структуры XML документа - определения правил, которым должен подчиняться документ.

По сравнению с DTD, схемы обладают более мощными средствами для определения сложных структур данных, обеспечивают более понятный способ описания грамматики языка, способны легко модернизироваться и расширяться.

Файл, содержащий XML Schema, обычно имеет расширение «.xsd» (XML-Schema-Definition).

Структура файла схемы данных:

```
<?XML version='1.0' ?>
```

```
<schema id="OurSchema">
```

```
<!-- последовательность инструкций -->
```

```
</schema>
```

Словарь документа

Определение класса элемента:

```
<elementType id="issue">  
<descript>Элемент содержит информацию о выпуске журнала</descript>  
</elementType>
```

Определение атрибутов элемента:

```
<elementType id="photo">  
<attribute name="src"/>  
</elementType>
```

Для этого правила корректным будет являться следующий фрагмент XML-документа:

```
<photo src="0"/>  
<photo src="some text"> Текст </photo>
```

Словарь документа

Ограничения на значения и способ использования атрибутов:

```
<elementType id="photo">  
  <attribute name="src" atttype="ID"/>  
  <attribute name="flower" atttype="ENUMERATION" values="red green  
blue" default="red"/>  
</elementType>
```

Список возможных значений атрибута:

```
<attribute name="flower" atttype="ENUMERATION" values="red green blue"  
default="red"/>
```

Для приведенных примеров корректным будет являться следующий фрагмент XML-документа:

```
<photo src="0" flower="green"/>  
<photo src="1"></photo>
```

Модель содержимого элемента

Вложенные элементы описываются при помощи инструкции `element`, в которой параметром `type` указывается объект в виде ссылки на его определение:

```
<elementType id="article">  
  <element type="#title"/>  
  <element type="#author"/>  
</elementType>
```

Для этого правила корректным будет являться следующий фрагмент документа:

```
<article>  
  <title>Любая информация</title>  
  <author>Любая информация</author>  
</article>
```

Модель содержимого элемента

Задание режима использования вложенного элемента через параметр `occurs`. Возможные значения этого параметра:

- **REQUIRED** - элемент должен быть обязательно определен
- **OPTIONAL** - использование элемента не является обязательным
- **ZEROORMORE** - вложенный элемент может встречаться несколько раз или ни разу
- **ONEORMORE** - элемент должен встречаться хотя бы один раз

Например,

```
<element type="#author" occurs="REQUIRED"/>
```

Модель содержимого элемента

Тип содержимого текущего элемента

<string/>

Пример использования:

```
<elementType id="flower">  
<string/>  
</elementType>
```

<any/>

<mixed/>

<empty/>

Если не задано ни одной инструкции - содержимое элемента было бы неопределенно.

Модель содержимого элемента

Пример:

```
<?xml version="1.0" ?>
<schema id="OurSchema">
  <elementType id="title">
    <string/>
  </elementType>
  <elementType id="chapter">
    <string/>
  </elementType>
  <elementType id="chapters-list">
    <any/>
  </elementType>
  <elementType id="content">
    <element type="#chapters-list" occurs="OPTIONAL">
  </elementType>
  <elementType id="article">
    <mixed><element type="#title"></mixed>
    <element type="#content" occurs="OPTIONAL">
  </elementType>
</schema >
```

Группировка элементов

```
<elementType id="contacts">  
  <element type="#tel" occurs="ONEORMORE">  
    <group occurs="OPTIONAL">  
      <element type="#email"/>  
      <element type="#url"/>  
    </group>  
</elementType>
```

Корректными будут являться следующие фрагменты документов:

```
<contacts>  
  <tel>12-12-12</tel>  
  <email>info@j.com</email>  
  <url>http://www.j.com</url>  
</contacts>  
<contacts>  
  <tel>12-12-12</tel>  
</contacts>  
<contacts>  
  <tel>12-12-12</tel>  
  <email>info@j.com</email>  
</contacts>
```

Группировка элементов

groupOrder задает режим использования группированных элементов
(возможные значения: OR, AND)

```
<elementType id="contacts">  
  <element type="#tel" occurs="ONEORMORE"/>  
  <group groupOrder="AND" occurs="OPTIONAL">  
    <element type="#email"/>  
    <element type="#url"/>  
  </group>  
</elementType>
```

Будут правильными только следующие варианты:

```
<contacts>  
  <tel>12-12-12</tel>  
  <email>info@j.com</email>  
  <url>http://www.j.com</url>  
</contacts>
```

```
<contacts>  
  <tel>12-12-12</tel>  
</contacts>
```

Закрытая и открытая модели описания содержимого элемента

Если необходимо ограничить создаваемые нами правила от включения дополнительных элементов, надо **использовать атрибут content** и **установить для него специальное значение CLOSED**:

```
<elementType id="contacts" content="CLOSED">  
  <element type="#tel"/>  
  <element type="#email"/>  
  <element type="#url"/>  
</elementType>
```

Иерархия классов

Инструкция <domain> указывает, что текущий объект должен определяться строго внутри элемента, заданного этим тэгом. Например, в следующем фрагменте указывается, что элемент <author> может быть определен строго внутри тэга <article>:

```
<elementType id="author">  
  <element type="#lastname"/>  
  <element type="#firstname"/>  
  <domain type="#article"/>  
</elementType>
```

Ограничения на значения

Значения элементов могут быть ограничены при помощи тэгов <min> и <max>:

```
<elementType id="room">  
<element type="#floor"><min>0</min><max>100</max></element>  
</elementType>
```

Внутри этих элементов могут указываться и символьные ограничения:

```
<elementType id="line">  
<element type="#character"><min>A</min><max>Z</max></element>  
</elementType>
```

Использование правил из внешних схем

Схема может использовать элементы и атрибуты из других схем.
Для этого надо использовать атрибут href, в котором указывается название внешней схемы.

Пример

```
<?xml version="1.0" ?>
<schema>
  <elementType id="author">
    <string/>
  </elementType>
  <elementType id="title">
    <string/>
  </elementType>
  <elementType id="Book">
    <element type="#title" occurs="OPTIONAL"/>
    <element type="#author" occurs="ONEORMORE"/>
    <element href="http://www.mysite.org/ex1.xsd" />
  </elementType>
</schema>
```

Типы данных

Задание типа данных для содержимого элемента, с использованием элемента `<datatype>` с указанием конкретного типа:

```
<elementType id="counter">  
  <datatype dt="int"/>  
</elementType>
```

Для обеспечения программной независимости необходимо сначала договориться об обозначениях типов данных (значения, которые должны передаваться параметру `dt` элемента `datatype`), для чего могут использоваться, например, универсальные идентификаторы ресурсов URI.



Описание структуры документа



Раздел:
Технологии создания сайтов