Схема XML-Data Reduced (XDR).

Презентацию подготовила Гайнетдинова Алсу, гр. 950б

Рассматриваются следующие темы для изучения:

1. Что такое схемы XDR. Различие между схемами DTD и другими схемами, такими как XDR.
2. Написание и интерпретация схем XDR.
3. Объявление элементов и атрибутов в схеме XDR.
4. Использование типов данных в схемах XDR.
5. Несколько примеров использования схем XDR и почему они важны для программистов на XML.

Что такое схемы XDR.

Стандарт XML-Data Reduced, или просто XDR, - это только часть полной спецификации XML-Data, направленной компанией *Microsoft* в консорциум *W3S.*

Как и схемы DTD, схемы XDR определяют элементы, представленные в экземпляре XML, а также любые атрибуты, которые связаны с указанными элементами. Для того, чтобы провести проверку на соответствие стандартам, каждый элемент в экземпляре должен быть объявлен в соответствующей схеме. Отсутствие элемента в схеме определяет его отсутствие в правильном экземпляре XML.

Взаимосвязи элементов также определяются в схеме XDR; таким образом, все дочерние элементы, содержащиеся в определенных элементах экземпляра, объявляются и в дальнейшем определяются в соответствии с их собственным содержанием. Сюда относятся последовательность, в которой дочерние Элементы приводятся в документе, а также количество дочерних элементов.

Схема определяет и то, пустой ли элемент или он содержит текстовые данные, другие элементы или определенную комбинацию последних. Схема также может определять значения атрибутов.

Объявления элементов и атрибутов XDR.

Схемы XDR – это экземпляры XML, а значат они состоят из элементов и атрибутов XML. Так как схемы XDR соответствуют всем синтаксическим правилам и стандартам XML, они должны соответствовать наборам правил, установленных для данного языка описания схем. Схемы XDR расширяемы.

Поскольку схемы XDR проверяются с помощью других систем, к их структурам применяются определенные правила и ограничения. Например, корневой элемент в схеме XDR всегда называется ***Shema*** и всегда содержит атрибут ***xmlns***, определяющий пространство имен как для него, так и для содержащихся в нем дочерних элементов.

Это означает, что вы создаете документ XML, который соответствует схеме, которая, в свою очередь, должна соответствовать еще одной схеме. К счастью, вам придется иметь дело только с 2 из этих трех документов: ваших документов XML и схемой, используемой для его проверки.

Создадим документ XML и сохраним его под именем message01.xml.

```
<?xml version=“1.0”?>
<note xmlns = “x-schema:message01.xdr”>Remembler to buy milk on the way home from work</note>
```

Корневой элемент в этом файле содержит атрибут xmlns со значением “x-schema:message01.xdr”>. Это указывает проверяемому обработчику на то, что схема, связанная со стандартным пространством имен для этого документа, содержится в файле message01.xdr.

Вы создадите схему XDR для этого элемента, содержащую корневой элемент Schema, и элемент ElementType, определяющий элемент note в экземпляре XML.

Корневой элемент Schema в схеме XDR содержит элементы и атрибуты, соответствующие всем ограничивающим правилам для экземпляра документа XML. Все определения в схеме XML содержатся в рамках элемента Schema верхнего уровня. Определение элемента Schema должно базироваться на пространстве имен xmlns = “urn: schemas-microsoft-com:xml-data”. Элемент Schema в документе схемы XML также должен содержать объявления пространств имен для других схем, таких как пространство имен, определяющее встроенные типы данных для схемы XML. Типы данных используются для ограничения строковых данных таким образом, чтобы они принимали только такие значения, как даты, числовые значения, числа с плавающей запятой и т.д. Пространство имен для схемы XDR: xmlns = “urn:schemas-microsoft-com:xml-data”.

Таким образом, корневой элемент Schema схемы XML практически всегда выглядит следующим образом:

```
<Schema
        name=“имя_схема xmlns=“urn:schemas-microsoft-com:xml-data”
        xmlns:=“urn:schemas-microsoft-com:datatypes”>
<--Объявления других элементов-->
</Schema>
```

Элемент ElementType – это элемент XDR, используемый для определения ограничений, которые накладываются на элементы в соответствующем экземпляре XML.

Элемент ElementType в схемах XDR имеет следующий синтаксис:

<ElementType
content="{empty | textOnly | mixed }"
dt:type"datatype"
model="{open | closed}"
name="idref"
order="{one | seq | many}" ?>

Каждый из этих атрибутов имеет значение, зависящее от описываемого элемента. Например,

атрибут content свидетельствует о том, что содержание объявляемого элемента должно быть пустым, быть только текстовыми данными, другими элементами или соответствовать модели смешанного содержания.

Атрибут dt:type объявляет тип данных для элемента. Допустимые типы данных – date, number, time и т.д.

Атрибут model указывает на то, может ли содержание быть только определяемого моделью типа, или оно может выходить за ее рамки. Если модель содержания определена как открытая, элемент может включать в себя дополнительные элементы или атрибуты, явно не указанные в этой модели. Если модель содержания определена как закрытая, элемент не может включать в себя доп.элементы и атрибуты.

Атрибут name используется для идентификации самого определяемого элемента.

Атрибут order определяет порядок следования дочерних элементов. Значение one допускает включение только 1 элемента из набора, значение seq требует указания элементов в строго определенном порядке, а значение many допускает указание элементов в любом порядке, вплоть до отсутствия некоторых из них.

Вложенные объявления **ElementType.**

Как и в случае DTD, вложенные элементы, указанные в схеме XDR, должны быть объявлены и определены. Изменим документ message01.xml так, чтобы в нем содержался один дочерний элемент, message, и один корневой элемент note, сохраним изменения в файле message02.xml:

```
<?xml version="1.0"?>
<note xmlns = "x-schema:message02.xdr">
<message>Remembler to buy milk on the way home from work</message>
</note>
```

Приведем схему XDR для проверки документа – message02.xml. Создадим эту схему и сохраним как файл – message02.xdr:

```
<?xml version="1.0"?>
<Schema
        name="Untitled-schema"
        xmlns="urn:schemas-microsoft-com:xml-data"
        xmlns:dt="urn:schemas-microsoft-com:datatypes">
<ElementType name="note" model="closed" content="eltOnly">
     <element type="message" />
</ElementType>
<ElementType name="message" model="closed" content="textOnly"
  dt:type="string"/>
</Shema>
```

Строки 1-5 практически совпадают с аналогичными строками из схемы message01.xdr. Однако в стр.3 указано имя схемы – его указывать не обязательно. Строки с 6 по 8 указывают на то, что элемент note содержит

только другие элементы (content="eltOnly"). Элемент message определяется как дочерний по отношению к элементу note.

В стр.9 элемент message определяется как содержащий только текстовые данные. Этот элемент указан в схеме дважды, один раз в стр.7, где он объявляется как дочерний по отношению к элементу note, а второй раз – в стр.9, где он определяется полностью.

Элемент element – это дочерний элемент по отношению к элементу ElementType в схеме XDR. В элемент ElementType могут быть включены дочерние элементы, перечисленные в следующей ***таб.1***:

Имя элемента:	***Описание:***
element	**Объявляет дочерний элемент**
description	**Обеспечивает описание элемента ElementType**
datatype	**Определяет тип данных элемента ElementType**
group	**Определяет порядок следования элементов**
AttributeType	**Определяет атрибут**
attribute	**Определяет сведения о дочернем элементе AttributeType**

Вложенные элементы.

Обязательно нужно помнить, что пустые элементы в экземпляре XML – это такие элементы, которые не содержат данные или другие элементы. Пустые элементы все равно могут содержать атрибуты, но между их открывающим и закрывающим дескрипторами не содержатся никаких данных. Это не означает, что пустые элементы совершенно бесполезны.

Рассмотрим файл message03.xml, который содержит пустой элемент number:

```
<?xml version="1.0"?>
<note xmlns = "x-schema:message03.xdr">
 <number />
<message>Remembler to buy milk on the way home from work</message>
</note>
```

Схема XDR для документа XML с пустым элементом (файл message03.xdr):

```
<?xml version="1.0"?>
<Schema
        name="Untitled-schema"
        xmlns="urn:schemas-microsoft-com:xml-data"
        xmlns:dt="urn:schemas-microsoft-com:datatypes">
<ElementType name="note" model="closed" content="eltOnly" order="seq">
     <element type="number" />
     <element type="message" />
</ElementType>
<ElementType name="number" model="closed" content="empty" />
<ElementType name="message" model="closed" content="textOnly"
  dt:type="string"/>
</Shema>
```

Проверка атрибутов в экземпляре XML с помощью схемы XDR.

Никогда не поздно передумать и объявить как элемент тот объект, который только что вы планировали объявить как атрибут.

Создадим пример, в котором к элементу message добавлен новый атрибут.

message04.xdr:

```
<?xml version=“1.0”?>
<note xmlns = “x-schema:message03.xdr”>
 <message number =“10”>
        Remembler to buy milk on the way home from work
  </message>
</note>
```

Поскольку был добавлен атрибут в экземпляр XML, необходимо отразить это дополнение и в схеме XDR. Атрибуты объявляются в схемах XDR с помощью элемента AttributeType:

```
<AttributeType
    default=“default-value”
    dt:type=“primitive-type”
    dt:values=“enumerated-values”
    name=“idref”
    reguired=“{yes | no}”>
```

Атрибуты элемента AttributeType

Таб2:

Атрибут:	*Описание:*
default	**Значение атрибута по умолчанию. Это значение допустимо для данного экземпляра атрибута. Например, если атрибут относится к типу с перечислением, значение по умолчанию должно указываться в списке**
dt:type	**Указывает тип данных для атрибута определенного типа. Атрибут может относиться к одному из следующих типов: *entity, entities, enumeration, id, idref, idrefs, nmtoken, nmtokens, notation* или *string.***
dt:values	**Если атрибут *dt:type* задан равным *enumeration,* то атрибут *dt:type* содержит все допустимые значения.**
name	**Имя типа атрибута. Этот атрибут обязателен.**
required	**Указывает на обязательное наличие атрибута в описании элемента.**

В нашем примере атрибут относится к элементу message. Таким образом, элемент ElementType для элемента note будет содержать элемент AttributeType и элемент attribute для объявления нового атрибута.

Рассмотрим элемент attribute:

```
<attribute
   default=“default-value”
   type=“attribute -type”
   [required=“{yes | no}”]>
```

Таб3

Атрибуты элемента attribute **в схемах XDR**

Атрибут:	***Описание:***
default	**Значение атрибута по умолчанию. Имеет преимущество перед любым значением по умолчанию, указанным в элементе** AttributeType
type	**Имя элемента** AttributeType**, определенного в данной схеме (или другой схеме, указанной с помощью соответствующего пространства имен). Указанное значение должно соответствовать атрибуту** name **элемента** AttributeType
required	**Указывает на обязательное наличие атрибута в описании элемента. Необязателен, если необходимый атрибут присутствует в соответствующем элементе** AttributeType

Объявление и определение атрибутов в схеме XDR (message04.xdr):

```
<?xml version=“1.0”?>
<Schema
        name=“Untitled-schema”
        xmlns=“urn:schemas-microsoft-com:xml-data”
        xmlns:dt=“urn:schemas-microsoft-com:datatypes”>
<ElementType name=“note” model=“closed” content=“eltOnly”>
     <element type=“message” />
</ElementType>
<ElementType name=“message” model=“closed” content=“textOnly”
   dt:type=“string”/>
<AttributeType name=“number” dt:type=“number” reguired=“yes”/>
  <attribute type=“number”/>
</ElementType>
</Shema>
```

В строках 10 и 11 объявляется и определяется атрибут number элемента message.

Типы данных в схемах XDR.

В схемах XDR вы можете указывать типы данных для элементов и атрибутов. Различных типов данных существует достаточно много, что идеально подходит для построения тех или иных предложений, работающих с данными. Типы данных определяют их форматирование и разрешают синтаксическим анализаторам XML и приложениям проверять типы данных во время проверки документов на соответствие стандартам. Для того, чтобы в схеме XDR могли быть определены типы данных, в объявлении корневого элемента Schema должно быть указано пространство имен типов данных *Microsoft.*

xmlns:dt="urn:schemas-microsoft-com:datatypes">

По соглашению префикс dt: назначается этому пространству имен и используется в схеме XDR для идентификации типов данных.

Типы данных для схем XDR

Таб4

Тип данных:	*Описание:*
bin.base64	**Определяет двоичные данные с использованием шифрования MIME Base64**
bin.hex	**Определяет двоичные данные в шестнадцатеричном формате**

boolean	**0 или 1, где 0==“ложь”, а 1==“истина”**
char	**Строка состоящая из 1 элемента**
date	**Указывает дату (без времени) в подмножестве формата ISO 8601. Например, 2002-11-05**
dateTime	**Указывает дату (указание времени не обязательно) в подмножестве формата ISO 8601. Доли секунд можно указывать вплоть до наносекунд.**
dateTime.tz	**Указывает дату (указания времени и часового пояса не обязательно) в подмножестве формата ISO 8601. Доли секунд можно указывать вплоть до наносекунд**
fixed.14.4	**Определяет число, содержащее не больше 14 цифр и не больше 4 цифр справа от десятичной точки. Например 12.0044**
float	**Определяет действительное число (без ограничений на количество цифр); может содержать знак, дроби, а также показатель степени. Значения изменяются от 1.7976931348623157E+308 до 2.2250738585072014E-308**
int	**Целое число с необязательным знаком**
number	**Определяет число (без ограничений на количество цифр); может содержать знак, дроби, а также показатель степени. Значения изменяются от 1.7976931348623157E+308 до 2.2250738585072014E-308**
time	**Указывает время без даты и часового пояса в подмножестве формата ISO 8601. Например 08:15:27**

Time.tz	**Указывает время и часовой пояс (необязательно) без даты в подмножестве формата ISO 8601. Например 08:15:27-05:00**
I1	**Определяет целое число, представленное один байтом (знак необязателен; дробные части и знаки возведения в степень не допускаются). Например 1,127,-128**
i2	**Определяет целое число, представленное одним словом (знак необязателен; дробные части и знаки возведения в степень не допускаются). Например 1, 703, -32768**
i4	**Определяет целое число, представленное четырьмя байтами (знак необязателен; дробные части и знаки возведения в степень не допускаются). Например 1, 703, -32768, 148343, 1000000000**
i8	**Определяет целое число, представленное восьмью байтами (знак необязателен; дробные части и знаки возведения в степень не допускаются). Диапазон значений от -9223372036854775808 до 9223372036854775807**
r4	**Действительное число с точностью в семь цифр. Может содержать знак, дробную часть, а также необязательный знак степени. Диапазон значений от 1.17549435E-38A до 3.40282347E+38F**
r8	**Действительное число с точностью в семь цифр. Может содержать знак, дробную часть, а также необязательный знак степени. Диапазон значений от 2.2250738585072014E-308 до 1.7976931348623157E+308**
ui1	**Целое число без знака, дробной части или степени. Например 1,255**
ui2	**Состоящее из двух байтов целое число без знака, дробной части или степени. Например 1, 255, 65535**
ui4	**Состоящее из четырех байтов целое число без знака, дробной части или степени. Например 1, 703, 3000000000**

ui8	**Состоящее из восьми байтов целое число без знака, дробной части или степени. Диапазон значений от 0 до 18446744073709551615**
uri	**Унифицированный идентификатор ресурса URI (uniform Resource identifier). Например urn:schemas-microsoft-com:Office9**
uuid	**Шестнадцатеричное число, разбитое на октеты, содержит необязательный переносы, которые игнорируются, например 333C7ИС4-460F-11D0-BC04-0080C7055A83**

Примеры типов данных.

Рассмотрим пример добавления новых атрибутов к документу message05.xml

```
<?xml version="1.0"?>
<note xmlns = "x-schema:message05.xdr">
 <message number ="10" date="2001-07-29" from="Kathy Shepherd">
        Remembler to buy milk on the way home from work
  </message>
</note>
```

Схема XDR для проверки указанных типов данных message05.xdr

```
<?xml version="1.0"?>
<Schema
        name="Untitled-schema"
        xmlns="urn:schemas-microsoft-com:xml-data"
        xmlns:dt="urn:schemas-microsoft-com:datatypes">
<ElementType name="note" model="closed" content="eltOnly">
     <element type="message" />
</ElementType>
<ElementType name="message" model="closed" content="textOnly"
   dt:type="string"/>
<AttributeType name="number" dt:type="int" reguired="yes"/>
<AttributeType name="date" dt:type="date" reguired="yes"/>
<AttributeType name="from" dt:type="string" reguired="yes"/>
  <attribute type="number"/>
  <attribute type="date"/>
  <attribute type="from"/>
</ElementType>
</Shema>
```

Индикаторы вхождения в схемах XDR

В схемах XDR необходимо указывать наличие нескольких дочерних элементов. Укажите, сколько раз дочерний элемент может встречаться в родительском элементе, используя атрибуты minOccur и maxOccur элемента element. Полный синтаксис приведен ниже:

```
<element
    type="element-type"
    [minOccur="{0 | 1"}"]
    [maxOccur="{0 | *"}"]
```

Вы уже сталкивались с элементами element, который содержали обязательный атрибут type. Этот атрибут связывает объявление с элементом в экземпляре XML.

Атрибут minOccur может принимать значения только от “0” до “1”. Строка minOccur=“1” указывает на то, что дочерний элемент должен содержаться как минимум 1 раз. Строка minOccur=“0” указывает на то, что дочерний элемент необязателен.

Атрибут maxOccur может принимать значения только от “1” или “*”. Строка maxOccur=“1” указывает на то, что дочерний элемент должен содержаться максимум 1 раз. Строка minOccur=“0” указывает на то, что дочерний элемент может встречаться в родительском элементе неограниченное число раз.

Атрибуты minOccur и maxOccur по умолчанию равны “1”.

Экземпляр XML с несколькими дочерними элементами message07.xml

```
<?xml version="1.0"?>
<note xmlns = "x-schema:message05.xdr">
  <message number ="10" date="2001-07-29" from="Kathy Shepherd">
      Remembler to buy milk on the way home from work
  </message>
  <message number ="12" date="2001-07-30" from="Greg Shepherd">
      I need some help with my homework
  </message>
  <message number ="12" date="2001-07-30" from="Greg Shepherd">
      I need some help with my homework
  </message>
  <message number ="14" date="2001-07-31" from="Kristen Shepherd">
      I need some help with my homework
  </message>
</note>
```

Схема XDR с атрибутами minOccur и maxOccur – message07.xdr

```
<?xml version="1.0"?>
<Schema
        name="Untitled-schema"
        xmlns="urn:schemas-microsoft-com:xml-data"
        xmlns:dt="urn:schemas-microsoft-com:datatypes">
<ElementType name="note" model="closed" content="eltOnly" order="seq">
    <element type="message" minOccurs="1" maxOccurs="*" />
</ElementType>
<ElementType name="message" model="closed" content="textOnly"
  dt:type="string"/>
<AttributeType name="number" dt:type="int" reguired="yes"/>
<AttributeType name="date" dt:type="date" reguired="yes"/>
<AttributeType name="from" dt:type="string" reguired="yes"/>
 <attribute type="number"/>
 <attribute type="date"/>
 <attribute type="from"/>
</ElementType>
</Shema>
```

Дополнительные ограничения типов данных в схемах XDR

Если вы решили использовать тип данных number или string (или bin.hex или bin.base64), вы можете задать минимальную или максимальную длину этих атрибутов. В пространстве имен типов данных Microsoft существуют специальные атрибуты minLength и maxLength, которые используются для проверки связанного документа XML. Эти атрибуты указываются в рамках элемента AttributeType; при этом используется следующий синтаксис:

```
<AttributeType name="idref"
        dt:type="string"
        reguired="yes"
        dt: minLength="положительное целое число"
        dt: maxLength="положительное целое число"/>
```

Определение групп содержания

Предположим, вам необходимо проверить целый ряд необязательных элементов в экземпляре XML, используя схему XDR. Например, в вашем приложении управления сообщениями вам надо реализовать выбор между 2 вариантами, такими как complete и incomplete. Например, если элемент message содержит элемент complete, это означает, что соответствующее сообщение принято приложением полностью. Возможно, какие-то неполадки не позволяют получить сообщение целиком. В этом случае элемент message будет содержать элемент incomplete, а не элемент complete.

Правильный экземпляр с группированием элементов message09.xml

```
<?xml version="1.0"?>
<note xmlns = "x-schema:message05.xdr">
  <message number ="10" date="2001-07-29" from="Kathy Shepherd">
       Remembler to buy milk on the way home from work
  </message>
                <receipt>
                    <complete/>
                <receipt/>
  <message number ="12" date="2001-07-30" from="Greg Shepherd">
       I need some help with my homework
  </message>
                <receipt>
                  <complete/>
                <receipt/>
  <message number ="12" date="2001-07-30" from="Greg Shepherd">
       I need some help with my homework
  </message>
                  <receipt>
                     <complete/>
                  <receipt/>
  <message number ="14" date="2001-07-31" from="Kristen Shepherd">
       I need some help with my homework
  </message>
</note>
```

Добавлены элементы receipt и complete.

Элемент group схемы XDR позволяет вам задавать ограничения для группы дочерних элементов в рамках родительского элемента. Этот элемент содержится в элементе ElementType.

(message09.xdr):

```
<?xml version="1.0"?>
<Schema
```

```
name="message"
          xmlns="urn:schemas-microsoft-com:xml-data"
          xmlns:dt="urn:schemas-microsoft-com:datatypes">
<ElementType name="note" model="closed" content="eltOnly" order="many">
     <element type="message" minOccurs="1" maxOccurs="*" />
      <element type="receipt" />
</ElementType>
<ElementType name="complete" model="closed" content="empty" />
 <ElementType name="incomplete" model="closed" content="textOnly"
                dt:type="string"/>
<ElementType name="message" model="closed" content="mixed" />
<AttributeType name="number" dt:type="int" reguired="yes"/>
<AttributeType name="date" dt:type="date" reguired="yes"/>
<AttributeType name="from" dt:type="string" reguired="yes"/>
  <attribute type="number"/>
  <attribute type="date"/>
  <attribute type="from"/>
</ElementType>
<ElementType name="receipt" model="closed" content="eltOnly" />
       <group order="one">
        <element type="complete" />
        <element type="incomplete" />
       <group/>
</ElementType>
</Shema>
```

В строке 9 объявляется, что новый элемент receipt содержится в корневом элементе note. Атрибут order="one" указывает на то, что в документе XML должен содержаться только один элемент complete или incomplete в рамках элемента message, чтобы этот документ считался правильным.

Расширяемость в схемах XDR.

Схемы XDR расширяемы, поскольку базируются на моделях открытого содержания, в которых используется синтаксис XML, расширяемый по умолчанию. При необходимости расширения, вам нужно включить пользовательское объявление типа данных, чтобы отличить пользовательские элементы и атрибуты от стандартных элементов и атрибутов XDR. Расширяемость возможна только в тех случаях, когда стандартный словарь XDR не обеспечивает простого способа ограничени1, необходимых для решения определенной задачи.

Рассмотрим пример системы управления сообщениями:

```
<ElementType name=“limit” xmlns:mystuff=“urn:mystuff-
    extensions:limits”>
      <datatype dt:type=“int”>
      <mystuff:max>100</mystuff:max>
</ElementType>
```