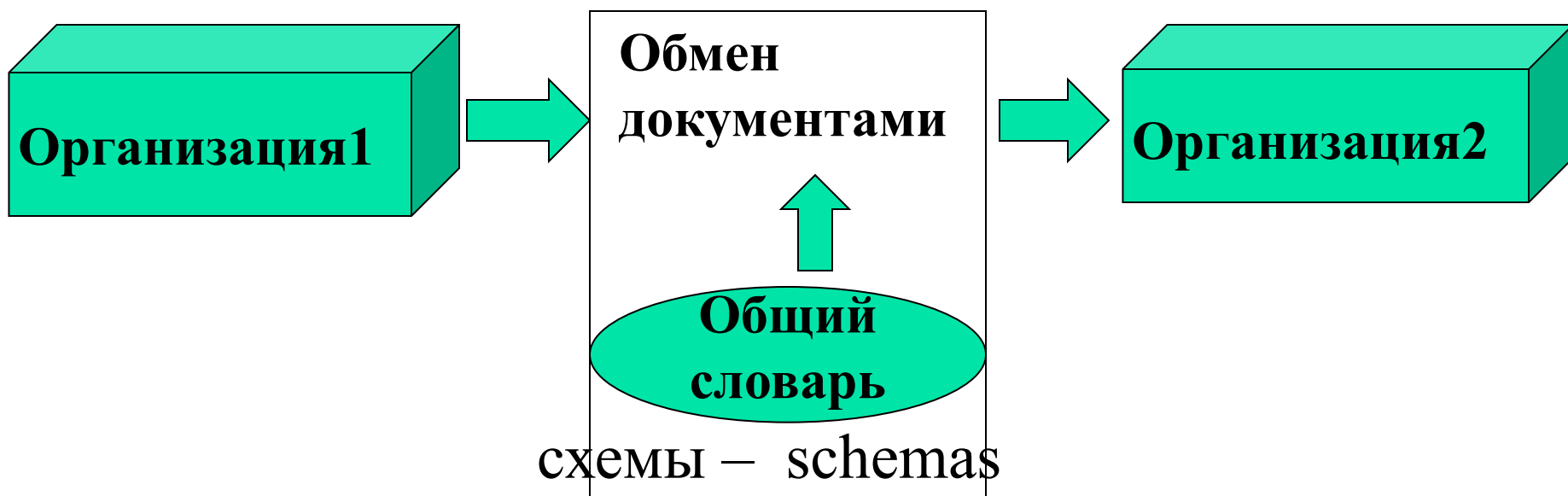
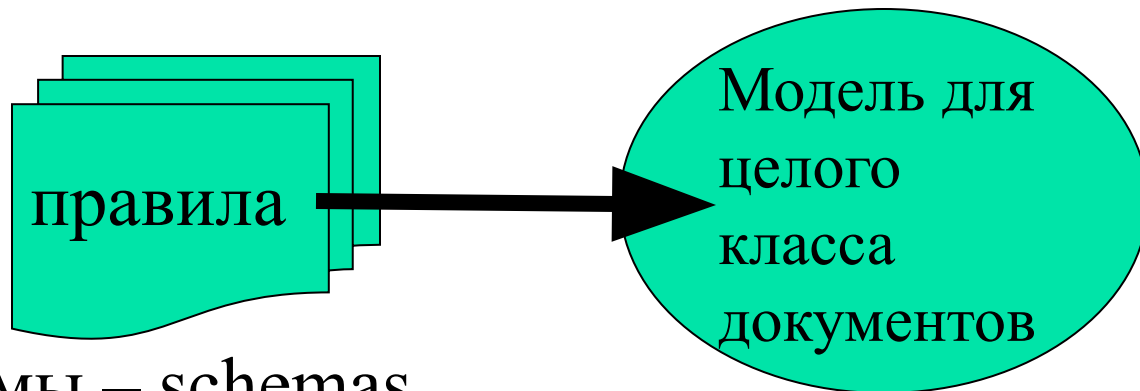


Схемы – Schemas

Цели занятия

- Понимание схем (Schema)
- Различие между схемой (Schema) и DTD
- Определение словаря схемы XML Schema Vocabulary
- Описание элементов схемы XML Schema
- Понимание проверки и ограничений
- Понимание наследования элементов в схеме (Schema)

Схемы – Schemas



Ограничения DTD

- DTD записываются не по правилам синтаксиса XML
- DTD не являются расширяемыми
- DTD не поддерживают пространства имён
- DTD лишь предлагают строго регламентированную типизацию данных

Новые функциональные возможности XML Schema

- Синтаксис XML используется, как основа для создания документов XML schema
- XML Schemas могут быть переданы и обработаны точно так же, как и любой другой документ XML
- Более богатый набор типов данных
- Архетипы (Archetypes)
- Группирование атрибутов
- Совершенствуемые архетипы

Типы данных в Schema

**Основные
типы
данных
(basetype)**

ID
IDREF
ENTITY
NMTOKEN
NOTATIONS
string
boolean
number
dateTime
binary
uri

**Дополнительные
типы данных
генерируются из
основных
встроенных типов
данных**

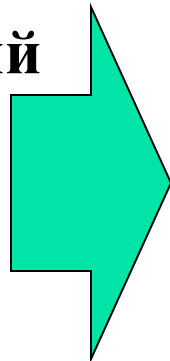
integer
decimal
real
time
timePeriod

**Генерируемые пользователем
типы данных создаются с
помощью встроенных типов
данных и дополнительных
условий, применяемых к ним, если
это необходимо**

Словарь схемы

XML Schema Vocabulary

**DTD,
предоставленный
Microsoft для
Schemas**



- Schema**
- datatype**
- ElementType**
- element**
- group**
- Attributetype**
- attribute**
- description**

Это корневой элемент всех документов, соответствующих XML schema

- Дочерние элементы**
- AttributeType**
- ElementType**
- description**

Образец документа XML со схемой Schema – 1

Файл schedule.xml

```
<batchschedule xmlns="x-schema:scheduleSchema.xml">
  <batch type="students" date="3/18/00">
    <timeslot>03:00:00</timeslot>
    <rating>8.5</rating>
    <location>M.G Road Centre</location>
    <report>Objective Achieved.</report>
  </batch>
  <batch type="professionals" date="3/18/00">
    <timeslot>08:00:00</timeslot>
    <rating>5</rating>
    <location>Wilson Road Centre</location>
    <report>Session Overflow.</report>
  </batch>
  <batch type="housewife" date="3/18/00">
    <timeslot>09:00:00</timeslot>
    <rating>4.5</rating>
    <location>Rex Street Centre</location>
    <report>Not Punctual.</report>
  </batch>
</batchschedule>
```


Образец документа XML со схемой Schema – 2

Схема (schema), ассоциированная с документом xml

```
<?xml version="1.0"?>|
<Schema name="scheduleSchema"
  xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="timeslot" content="textOnly" dt:type="time"/>
  <ElementType name="rating" content="textOnly" dt:type="float"/>
  <ElementType name="location" content="textOnly"/>
  <ElementType name="report" content="textOnly"/>
  <AttributeType name="type" dt:type="enumeration"
    dt:values="students professionals housewife "/>
  <AttributeType name="date" dt:type="date"/>
  <ElementType name="batch" content="eltOnly" order="seq">
    <description>Information of single batch. </description>
    <element type="timeslot" minOccurs="1" maxOccurs="1"/>
    <element type="rating" minOccurs="1" maxOccurs="1"/>
    <element type="location" minOccurs="1" maxOccurs="1"/>
    <element type="report" minOccurs="0" maxOccurs="1"/>
    <attribute type="type" default="students"/>
    <attribute type="date"/>
  </ElementType>
  <ElementType name="batchschedule" content="eltOnly">
    <description>Information of one or more batch. </description>
    <element type="batch" minOccurs="1" maxOccurs="*/>
  </ElementType>
</Schema>
```

Выходные данные примера – 1

```
- <Schema name="scheduleSchema" xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes">
  <ElementType name="timeslot" content="textOnly" dt:type="time" />
  <ElementType name="rating" content="textOnly" dt:type="float" />
  <ElementType name="location" content="textOnly" />
  <ElementType name="report" content="textOnly" />
  <AttributeType name="type" dt:type="enumeration"
    dt:values="students professionals housewife" />
  <AttributeType name="date" dt:type="date" />
  <ElementType name="batch" content="eltOnly" order="seq">
    <description>Information of single batch.</description>
    <element type="timeslot" minOccurs="1" maxOccurs="1" />
    <element type="rating" minOccurs="1" maxOccurs="1" />
    <element type="location" minOccurs="1" maxOccurs="1" />
    <element type="report" minOccurs="0" maxOccurs="1" />
    <attribute type="type" default="students" />
    <attribute type="date" />
  </ElementType>
  <ElementType name="batchschedule" content="eltOnly">
    <description>Information of one or more batch. training sessions.</description>
    <element type="batch" minOccurs="1" maxOccurs="*" />
  </ElementType>
</Schema>
```

Выходные данные примера – 2

```
<?xml version="1.0" standalone="no" ?>
- <batchschedule xmlns="x-schema:scheduleSchema.xml">
- <batch type="students" date="3/18/00">
  <timeslot>03:00:00</timeslot>
  <rating>8.5</rating>
  <location>M.G Road Centre</location>
  <report>Objective Achieved.</report>
</batch>
- <batch type="professionals" date="3/18/00">
  <timeslot>08:00:00</timeslot>
  <rating>5</rating>
  <location>Wilson Road Centre</location>
  <report>Session Overflow.</report>
</batch>
- <batch type="housewife" date="3/18/00">
  <timeslot>09:00:00</timeslot>
  <rating>4.5</rating>
  <location>Rex Street Centre</location>
  <report>Not Punctual.</report>
</batch>
</batchschedule>
```

Элемент Schema

- ‘Schema’ – это корневой элемент для любого документа XML schema
- Атрибутами элемента Schema являются:
 - ◆ name
 - ◆ xmlns
- Для типов данных XML schema должно быть определено пространство имён
- Структура документа XML schema:

```
<Schema                                name="scheduleSchema"  
xmlns="urn:schemas-microsoft-com:xml-data">  
<!-- тело схемы -->  
</Schema>
```

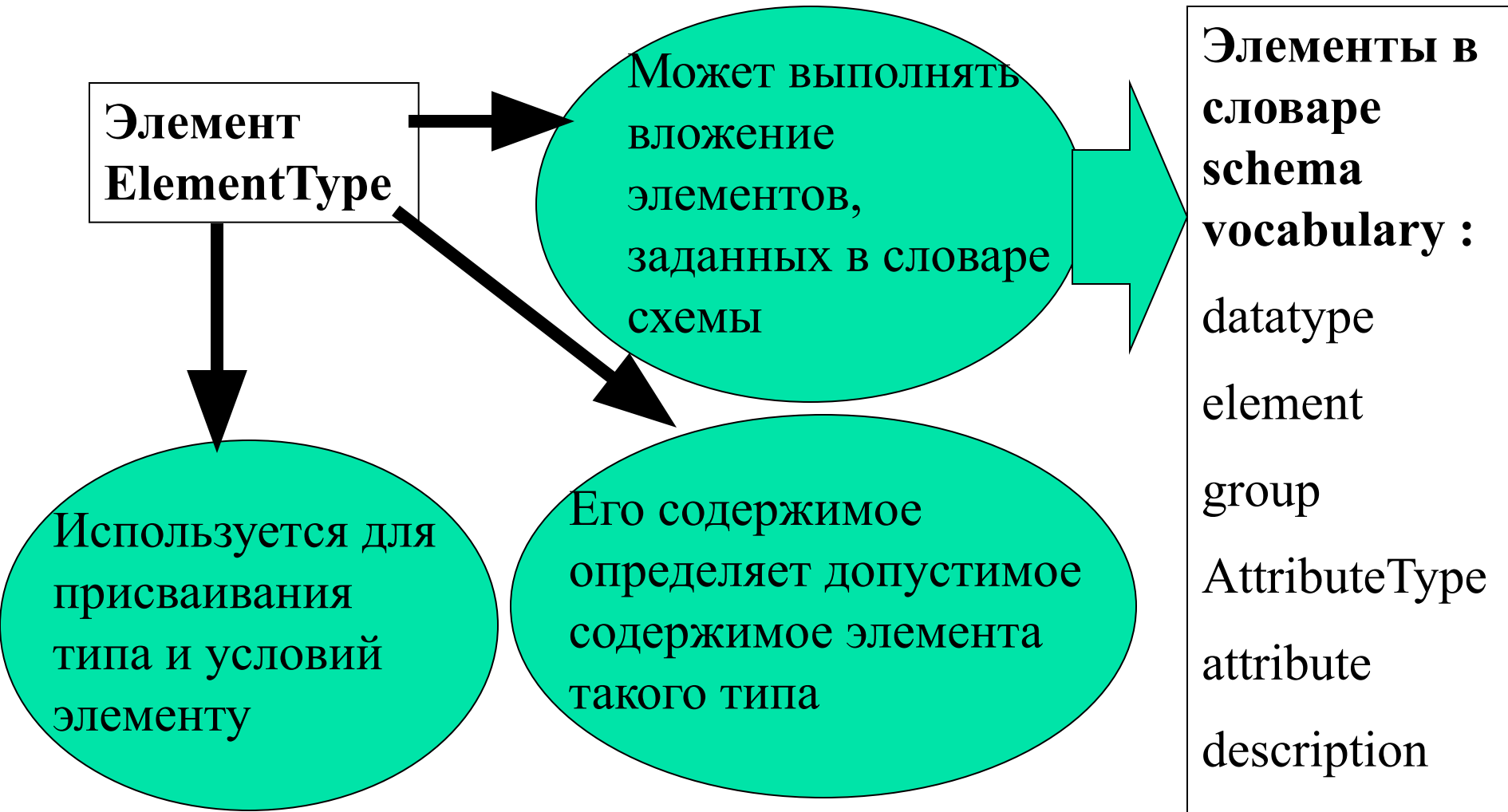
Элемент datatype

- Определяет тип данных datatype элемента или атрибута
- Включает атрибут 'dt:type'
- Значения, допустимые для атрибута dt:type:
 - ◆ char
 - ◆ boolean
 - ◆ int
 - ◆ float
 - ◆ number

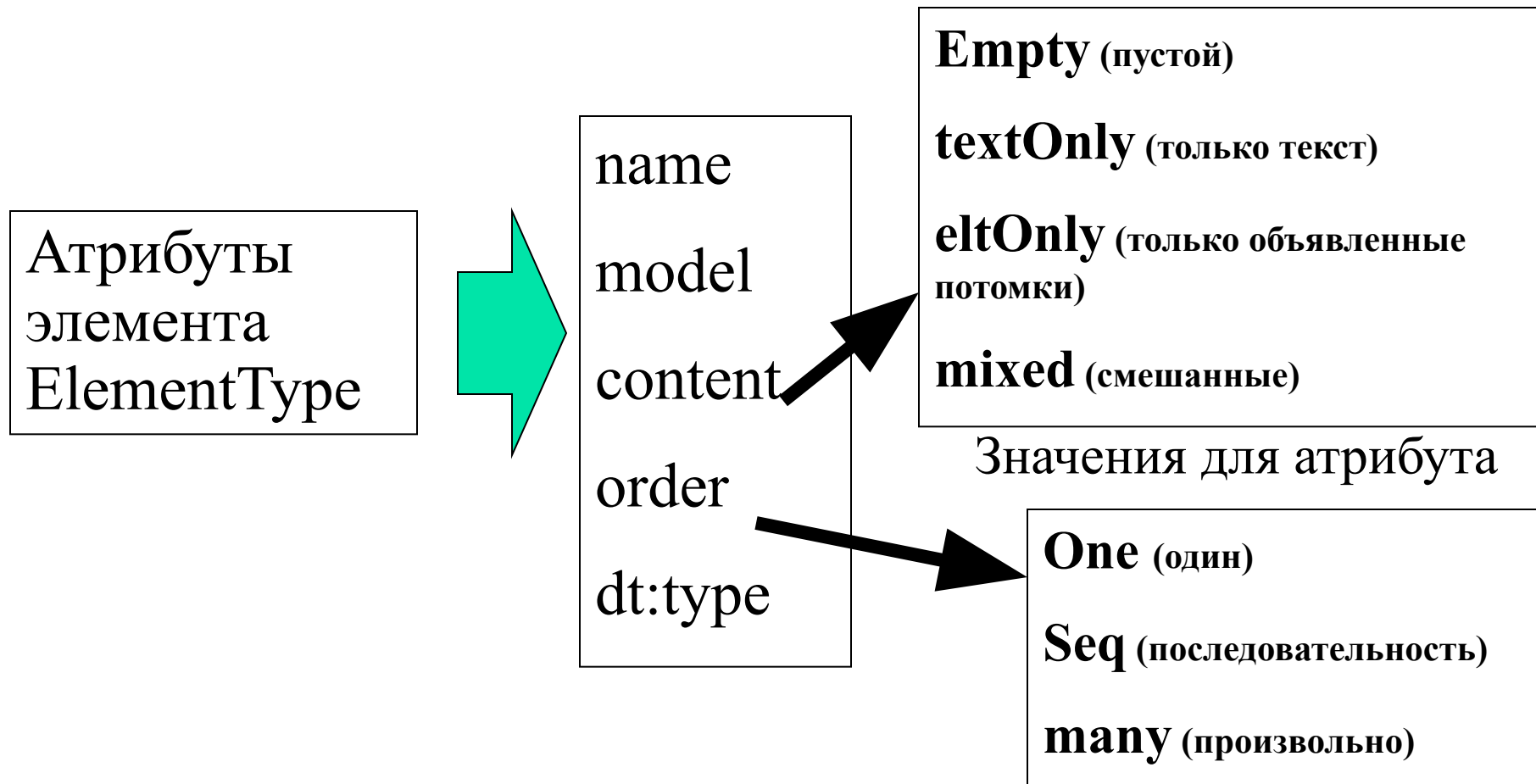
Пример:

```
<ElementType name="rating" content="textOnly" dt:type="float"/>
```

Элемент ElementType – 1



Элемент ElementType – 2



Элемент Element – 1

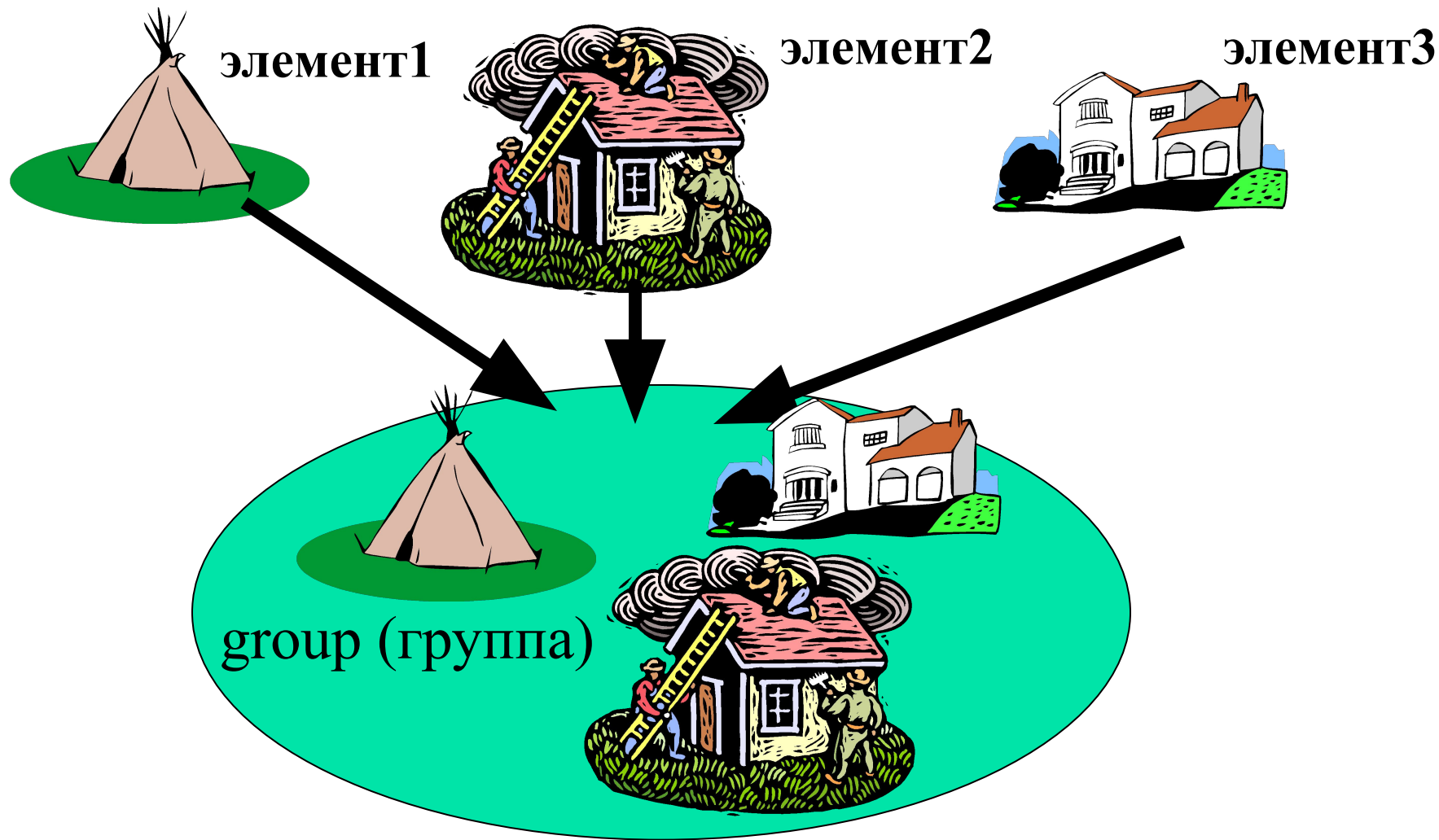
- Используется для объявления экземпляра элемента с группой или типом элемента
- Три атрибута элемента ‘element’:
 - ◆ type: используется для определения типа элемента
 - ◆ minOccurs: минимальное количество вхождений, разрешённое для данного элемента
 - ◆ maxOccurs: максимальное количество вхождений, разрешённое для данного элемента

Элемент Element – 2

Отношения между атрибутами minOccurs и maxOccurs

minOccur	maxOccur	Сколько раз может встречаться элемент
0	1	0 или 1
1	1	1
0	*	Не ограничено
1	*	Как минимум, один
>0	*	Как минимум, minOccurs раз
>maxOccurs	>0	0
Любое значение	<minOccurs	0

Элемент group



Элемент `AttributeType`

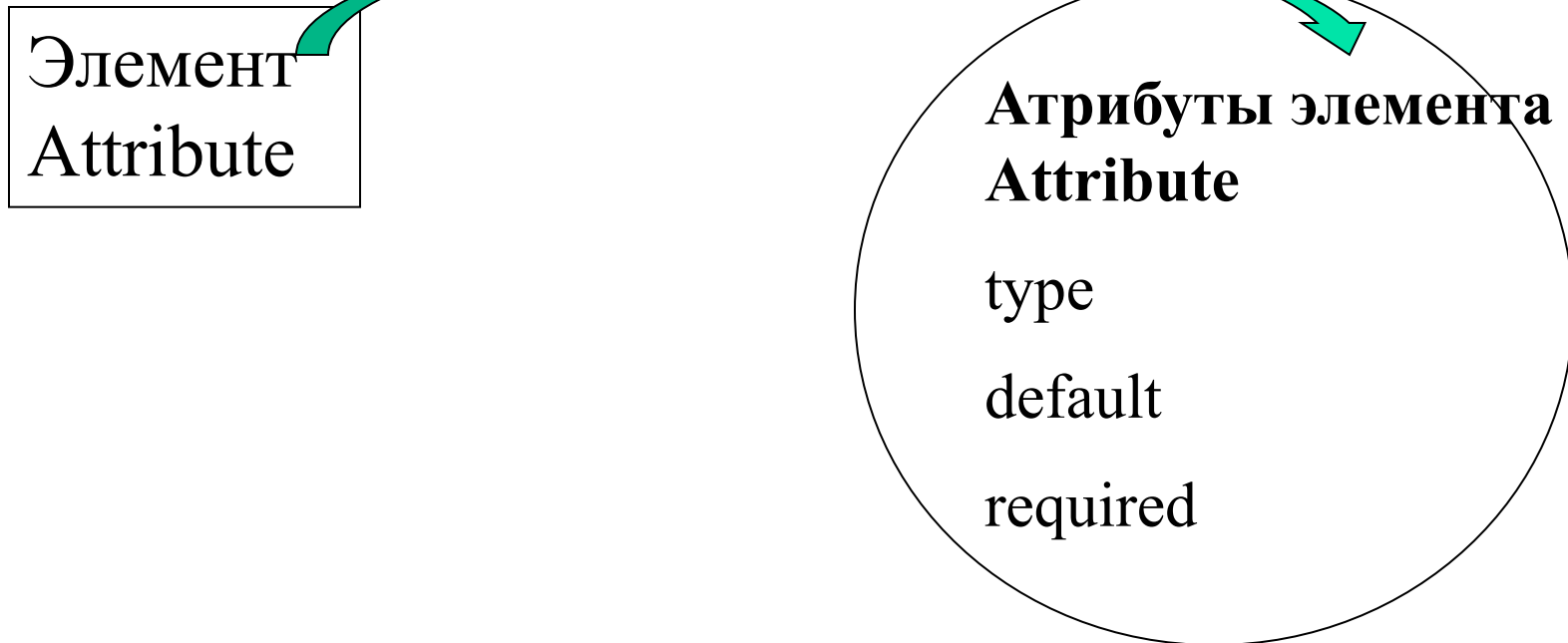
- Похож на элемент `ElementType`
- Определяет тип атрибута
- Атрибуты элемента `AttributeType`:
 - ◆ `name`
 - ◆ `dt:type`
 - ◆ `dt:values`
 - ◆ `default`
 - ◆ `required`

Пример :

```
<AttributeType name= "type" dt:type= "enumeration"  
    dt:values= "students professional housewife"/>
```

Элемент attribute – 1

- Объявляет экземпляр атрибута для типа элемента



Элемент attribute – 2

Пример :

...

```
<AttributeType name="type" dt:type="enumeration"  
    dt:values="students professionals housewife "/>
```

```
<AttributeType name="date" dt:type="date"/>
```

```
<ElementType name="batch" content="eltOnly" order="seq">  
    <element type="timeslot" minOccurs="1" maxOccurs="1"/>  
    <element type="rating" minOccurs="1" maxOccurs="1"/>  
    <element type="location" minOccurs="1" maxOccurs="1"/>  
    <element type="report" minOccurs="0" maxOccurs="1"/>  
    <attribute type="type" default="students"/>  
    <attribute type="date"/>  
</ElementType>
```

...

Элемент description

- Описывает смысл текста, размещаемого
внутри схемы

Пример :

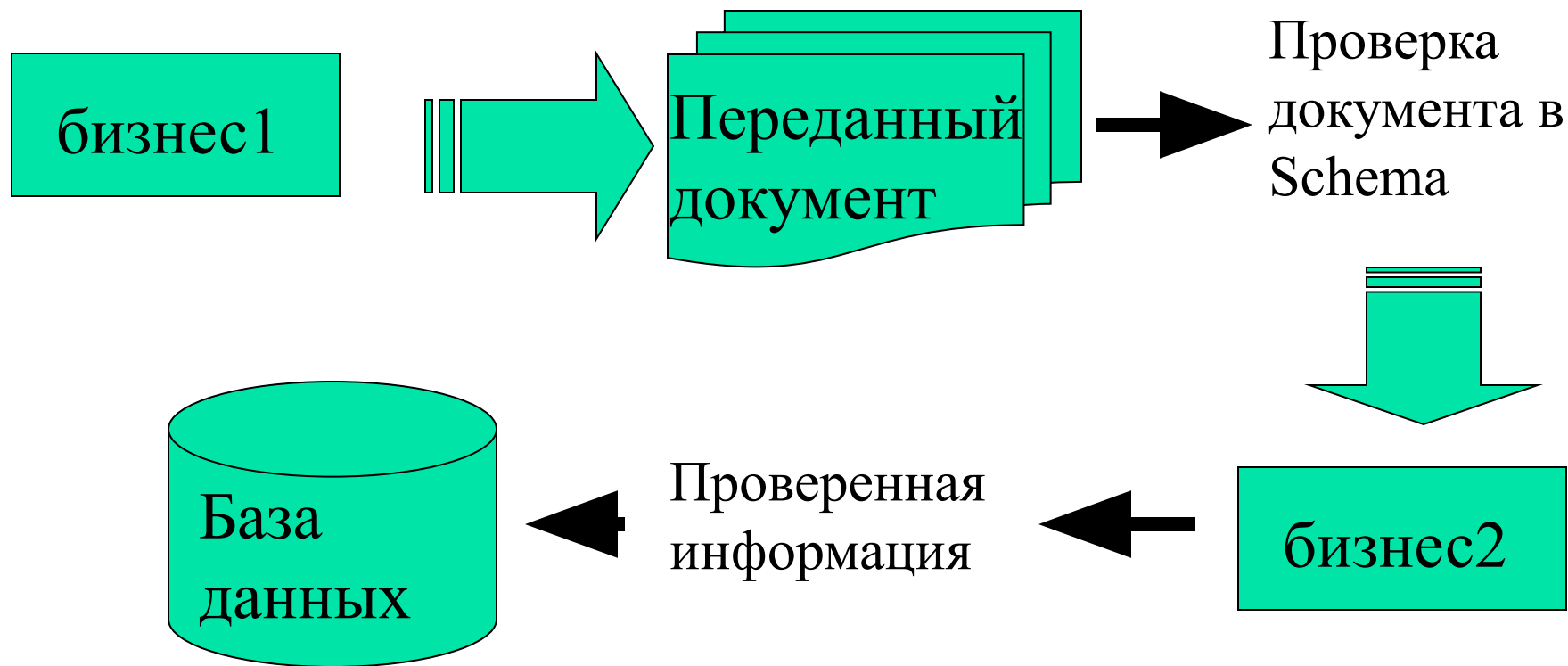
```
...  
<ElementType name="batchschedule" content="eltOnly">  
  <description>  
    Information of one or more batch.  
  </description>  
  <element type="batch" minOccurs="1" maxOccurs="*" />  
</ElementType>  
...
```

Ограничения DTD для Schema

- Модели содержания DTD являются закрытыми и, следовательно, не могут отображать открытость XML Schema
- DTD не поддерживает более широкий набор типов данных, разрешённых в XML Schema
- DTD не поддерживает гибкость пространств имён, свойственную XML Schema

Проверка (Validity)

Причины для проверки



Классификация документов

- Не XML документы
- Документ XML, который не идентифицирует схему
- Документ XML, не соответствующий модели, описанной ассоциированной схемой
- Документ XML, который не нарушает ни одного ограничения (условия) схемы

Виды проверок

- Content Model Validity (Корректность модели содержания)
 - Проверяет правильность порядка следования и вложенности тэгов
- Datatype Validity (Корректность типов данных)
 - Обозначает возможность проверить, имеют ли заданные блоки информации корректный тип, и находятся ли в пределах заданных допустимых значений

Ограничения в Schema

- Схемы описаны в терминах ограничений
- Ограничение определяет, что может появляться в документе и каким образом
- Существуют два вида ограничений:
 - ◆ Content model constraints (Ограничения модели содержания)
 - ◆ Datatype constraints (Ограничения модели типов данных)

Пример:

```
<datatype name="ISBNCode">  
  <basetype name="string"/>  
  <lexicalRepresentation>  
    <lexical>99</lexical>  
    <lexical>99-9999999</lexical>  
  </lexicalRepresentation>  
</datatype>
```

Наследование элементов

- Использует объектно-ориентированный подход для поддержки отношений между элементами
- Может быть использовано для создания архетипов (Archetypes), которые являются шаблонами (templates) элементов, и от которых могут быть произведены другие элементы
- XML Schema является технологией Microsoft
- Активное использование Schema возможно не всегда, так как на текущий момент она реализована лишь частично

Пример архетипа (Archetype)

...

```
<archetype name="BOOK" model="refinable">
  <elementTypeRef name="Title" minOccurs="1" maxOccurs="1"/>
  <elementTypeRef name="Author" minOccurs="1" maxOccurs="4"/>
  <elementTypeRef name="Theme" minOccurs="1" maxOccurs="2"/>
  <elementTypeRef name="Publisher" minOccurs="1" maxOccurs="1"/>
  <elementTypeRef name="ISBN" minOccurs="1" maxOccurs="1"/>
  <elementTypeRef name="Price" minOccurs="1" maxOccurs="1"/>
  <elementTypeRef name="Edition" minOccurs="1" maxOccurs="1"/>
</archetype>
<elementType name="Indian.BOOK">
  <archetypeRef name="BOOK"/>
</elementType>
<elementType name="Foreign.BOOK">
  <archetypeRef name="BOOK"/>
</elementType>
```

...

Подведение итогов – I

- В соответствии с определением консорциума W3C схема – это "набор правил, регламентирующих структуру и выполняющих сопряжения информационных наборов в документах XML". Схема – это модель, применяемая для описания структуры базы данных.
- XML унаследовал концепцию определений DTD (Document Type Definitions) от языка SGML. DTD используются для определения моделей содержимого, правильного (корректного) порядка размещения и вложенности элементов и для некоторого ограниченного расширения, а также определяют типы данных атрибутов.
- XML Schema предлагает целый ряд новых функциональных возможностей.
 - ◆ Синтаксис XML используется, как основа для создания документов XML schema.
 - ◆ XML Schemas могут быть переданы и обработаны точно так же, как и любой другой документ XML.
 - ◆ Более богатый набор типов данных.
 - ◆ Архетипы (Archetypes)
 - ◆ XML Schemas поддерживает интеграцию пространств имён, которая позволяет ассоциировать отдельные узлы документа с объявлениями типов в схеме.
 - ◆ Группирование атрибутов
 - ◆ Усовершенствованные архетипы

Подведение итогов – II

- Схемы поддерживают content model validity (Проверка модели содержания) и datatype validity (Проверка типов данных)
- Схемы описаны в терминах ограничений. Ограничение определяет, что может появляться в документе
- Наследование элементов, в соответствии с проектом рабочего стандарта XML Schema, использует объектно-ориентированный подход для поддержки отношений между элементами.
- Мы можем создавать архетипы (archetypes), которые являются шаблонами (templates) элементов, и от которых могут быть произведены другие элементы.
- Используя наследование элементов, процессор XML может ассоциировать производные элементы друг с другом и сохранить их отношение.