

Лабораторная №3. Часть 2

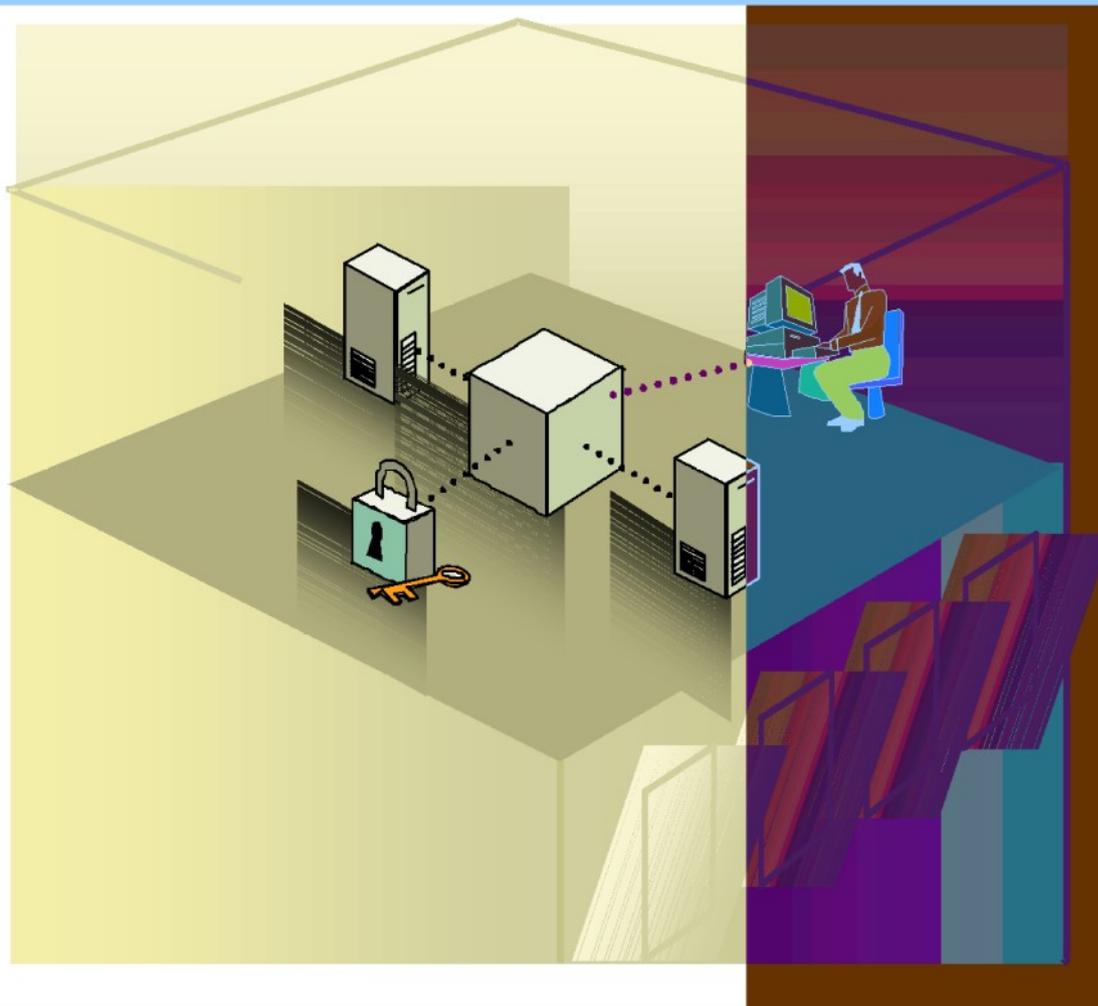
XML

XML

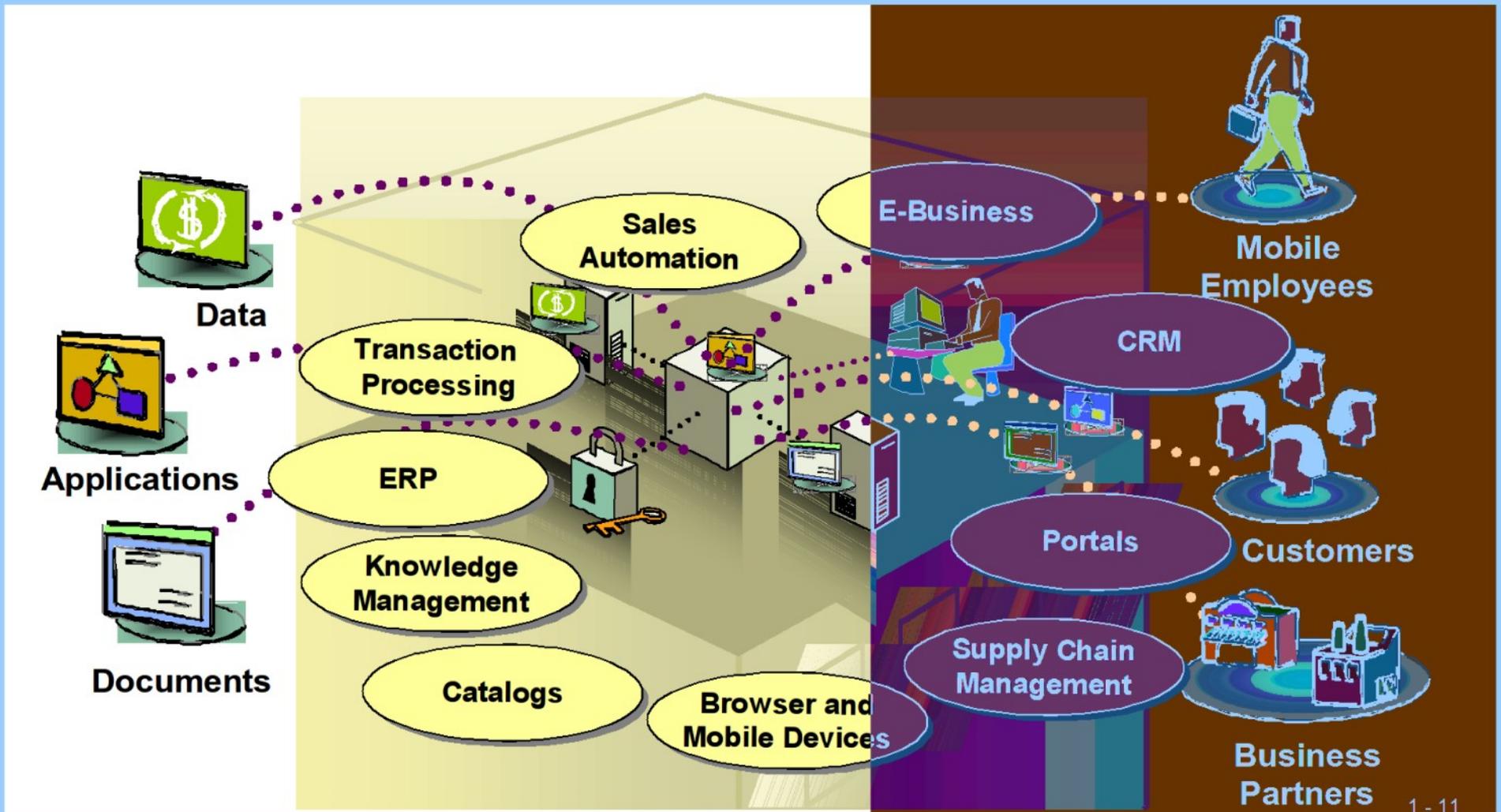
eXtensible Markup Language

расширяемый язык разметки

Ситуация раньше



Задачи сейчас



История

SGML (Standard Generalized Markup Language — стандартный обобщённый язык разметки) - гибкий, комплексный и всеохватывающий мета-язык для создания языков разметки. SGML не имеет гипертекстовой модели. Создание SGML можно с уверенностью назвать попыткой объять необъятное, так как он объединяет в себе такие возможности, которые крайне редко используются все вместе. В этом и состоит его главный недостаток — сложность и, как следствие, дороговизна этого языка ограничивает его использование только крупными компаниями, которые могут позволить себе купить соответствующее программное обеспечение и нанять высокооплачиваемых специалистов.

История

Наиболее широко SGML применяется для создания других языков разметки. С его помощью был создан язык разметки гипертекстовых документов — HTML (спецификация была утверждена в 1992 году). Появление HTML — очень простого языка разметки — быстро решило проблему простого представления сложной информации: лёгкость в изучении и богатство средств оформления документов сделали его самым популярным языком для пользователей Интернет. Но, по мере роста количества и изменения качества документов в Сети, росли и предъявляемые к ним требования, и простота HTML превратилась в его главный недостаток. Ограниченность количества тегов и полное безразличие к структуре документа побудили разработчиков в лице консорциума W3C к созданию такого языка разметки, который был бы не столь сложен, как SGML, и не настолько примитивен, как HTML. В результате на свет появился язык XML, сочетающий в себе простоту HTML, логику разметки SGML и удовлетворяющий требованиям Интернета.

Версии

1996 — черновой вариант, рождение

10.02.1998 — утверждение рекомендации от W3C, версия "1.0"

06.10.2000 — вторая редакция версии "1.0"

04.02.2004 — третья редакция версии "1.0"

26.11.2008 — пятая редакция версии "1.0"

04.02.2004 — версия "1.1"

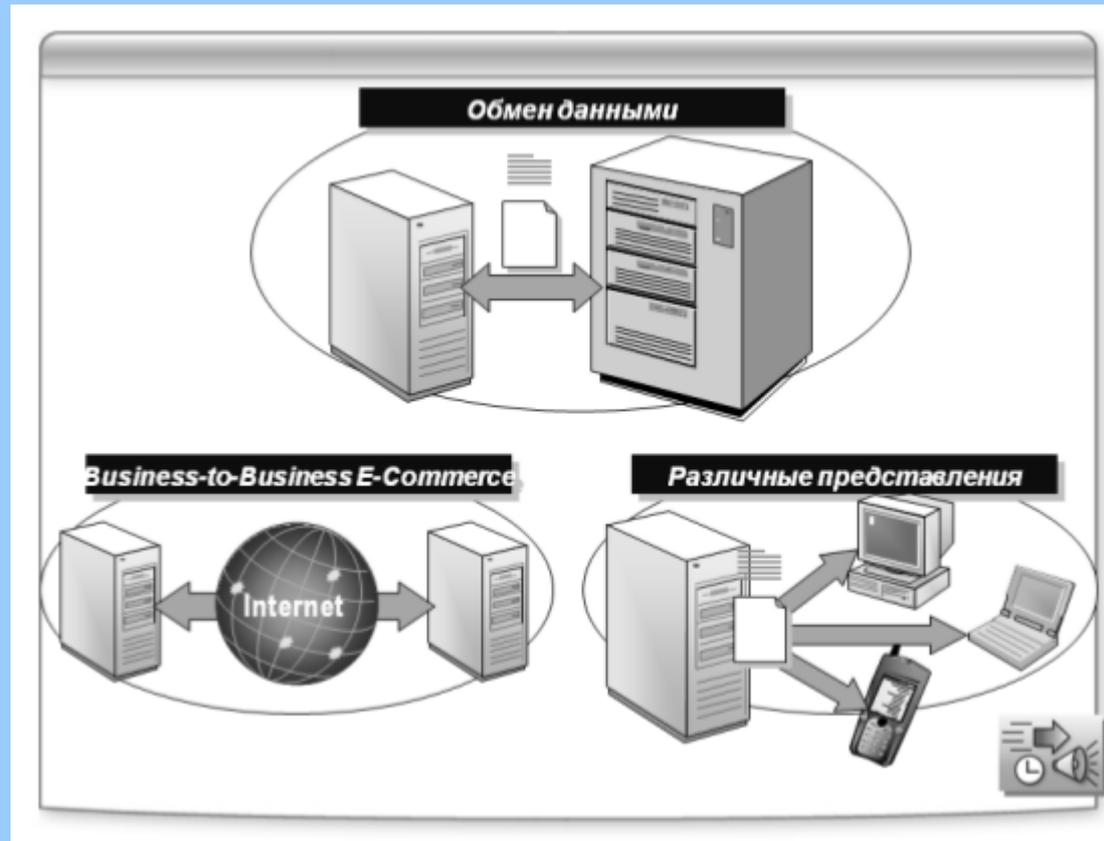
15.08.2006 — вторая редакция версия "1.1"

Основные отличия 1.1 от 1.0

В самом начале работ над XML 1.1 члены рабочей группы XML Core обсуждали возможность изменения основы XML с Unicode 2.0 на последнюю изданную версию Unicode (тогда это была версия 3.0) посредством простого добавления новых символов в существующие конструкции. Однако, это было бы временным решением, поскольку после выхода еще нескольких версий Unicode, рабочей группе пришлось бы начинать все с самого начала. Именно поэтому члены группа избрали более радикальный подход: совместимость снизу вверх.

- Изменения определения конца строки.
- Добавлены символы управления.
- Проверка нормализации символов.

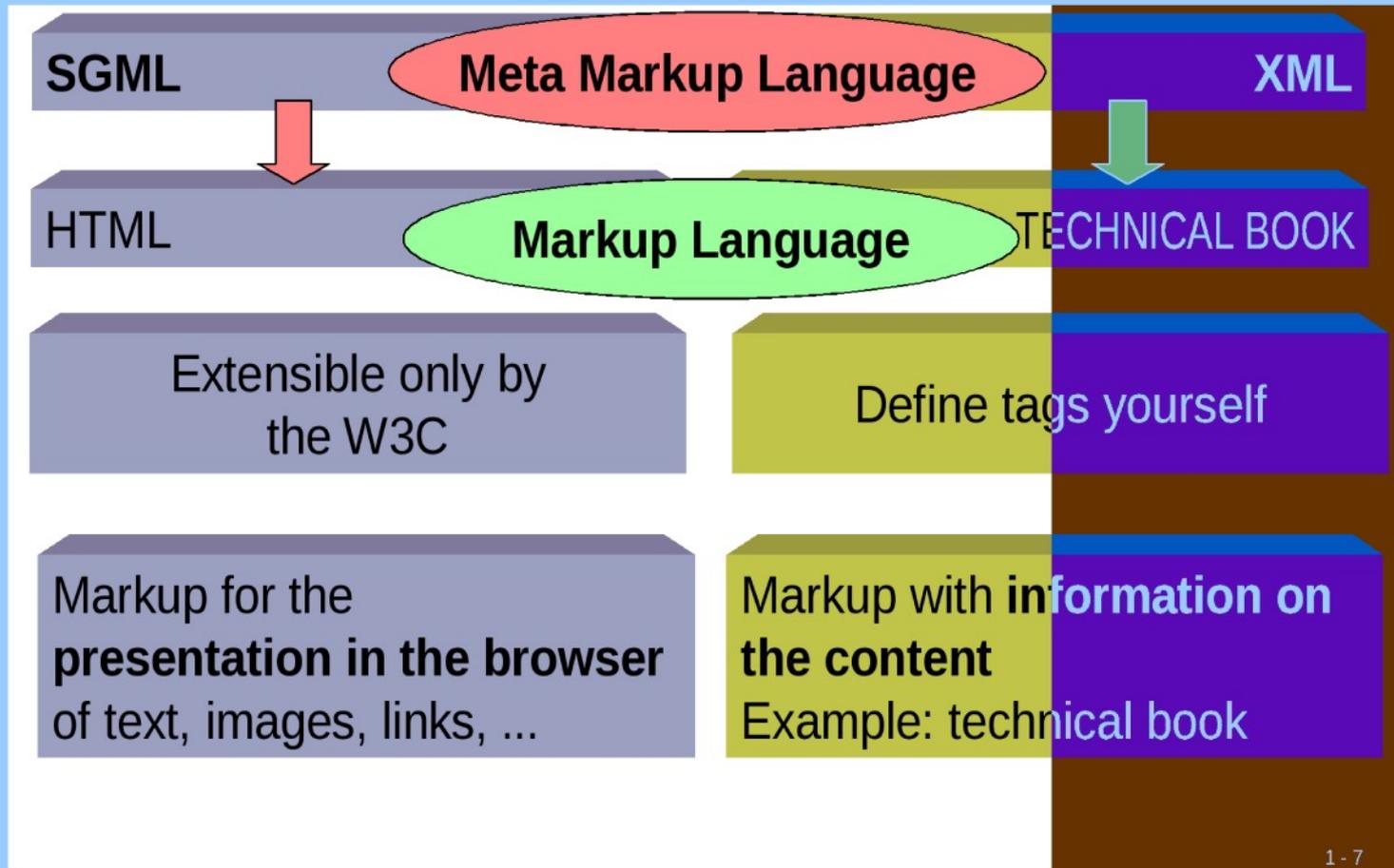
Применение



HTML / XML

```
<?xml version = "1.0" standalone="yes"?>
<DOCUMENT>
  <CUSTOMER>
    <NAME>
      <LAST_NAME>Smith</LAST_NAME>
      <FIRST_NAME>Sam</FIRST_NAME>
    </NAME>
    <DATE>October 15, 2003</DATE>
    <ORDERS>
      <ITEM>
        <PRODUCT>Tomatoes</PRODUCT>
        <NUMBER>8</NUMBER>
        <PRICE>$1.25</PRICE>
      </ITEM>
      <ITEM>
        <PRODUCT>Oranges</PRODUCT>
        <NUMBER>24</NUMBER>
        <PRICE>$4.98</PRICE>
      </ITEM>
    </ORDERS>
  </CUSTOMER>
  <CUSTOMER>
    <NAME>
      <LAST_NAME>Jones</LAST_NAME>
      <FIRST_NAME>Polly</FIRST_NAME>
    </NAME>
    <DATE>October 20, 2003</DATE>
```

HTML / XML



Достоинства

- XML — язык разметки, позволяющий стандартизировать вид файлов-данных, используемых компьютерными программами, в виде текста, понятного человеку;
- XML поддерживает Юникод;
- в формате XML могут быть описаны такие структуры данных как записи, списки и деревья;
- XML — это самодокументируемый формат, который описывает структуру и имена полей так же как и значения полей;
- XML имеет строго определённый синтаксис и требования к анализу, что позволяет ему оставаться простым, эффективным и непротиворечивым. Одновременно с этим, разные разработчики не ограничены в выборе экспрессивных методов (например, можно моделировать данные, помещая значения в параметры тегов или в тело тегов, можно использовать различные языки и нотации для именования тегов и т. д.);

Достоинства

- XML — формат, основанный на международных стандартах;
- Иерархическая структура XML подходит для описания практически любых типов документов, кроме аудио и видео мультимедийных потоков, растровых изображений, сетевых структур данных и двоичных данных;
- XML представляет собой простой текст, свободный от лицензирования и каких-либо ограничений;
- XML не зависит от платформы;
- Уже накоплен большой опыт работы с языком и созданы специализированные приложения;
- XML не накладывает требований на порядок расположения атрибутов в элементе и вложенных элементов разных типов, что существенно облегчает выполнение требований обратной совместимости;

Достоинства

- В отличие от бинарных форматов, XML содержит метаданные об именах, типах и классах описываемых объектов, по которым приложение может обработать документ неизвестной структуры (например, для динамического построения интерфейсов);
- XML имеет реализации парсеров для всех современных языков программирования;
- Существует стандартный механизм преобразования XSLT, реализации которого встроены в браузеры, операционные системы, веб-серверы;
- XML поддерживается на низком аппаратном, микропрограммном и программном уровнях в современных аппаратных решениях.

Недостатки

- Синтаксис XML избыточен.
- Размер XML-документа существенно больше бинарного представления тех же данных. В грубых оценках величину этого фактора принимают за 1 порядок (в 10 раз).
- Размер XML-документа существенно больше, чем документа в альтернативных текстовых форматах передачи данных (например JSON) и особенно в форматах данных, оптимизированных для конкретного случая использования.
- Избыточность XML может повлиять на эффективность приложения. Возрастает стоимость хранения, обработки и передачи данных.
- XML содержит метаданные (об именах полей, классов, вложенности структур), и одновременно XML позиционируется как язык взаимодействия открытых систем. При передаче между системами большого количества объектов одного типа (одной структуры), передавать метаданные повторно нет смысла, хотя они содержатся в каждом экземпляре XML описания.
- Для большого количества задач не нужна вся мощь синтаксиса XML и можно использовать значительно более простые и производительные решения.

Недостатки

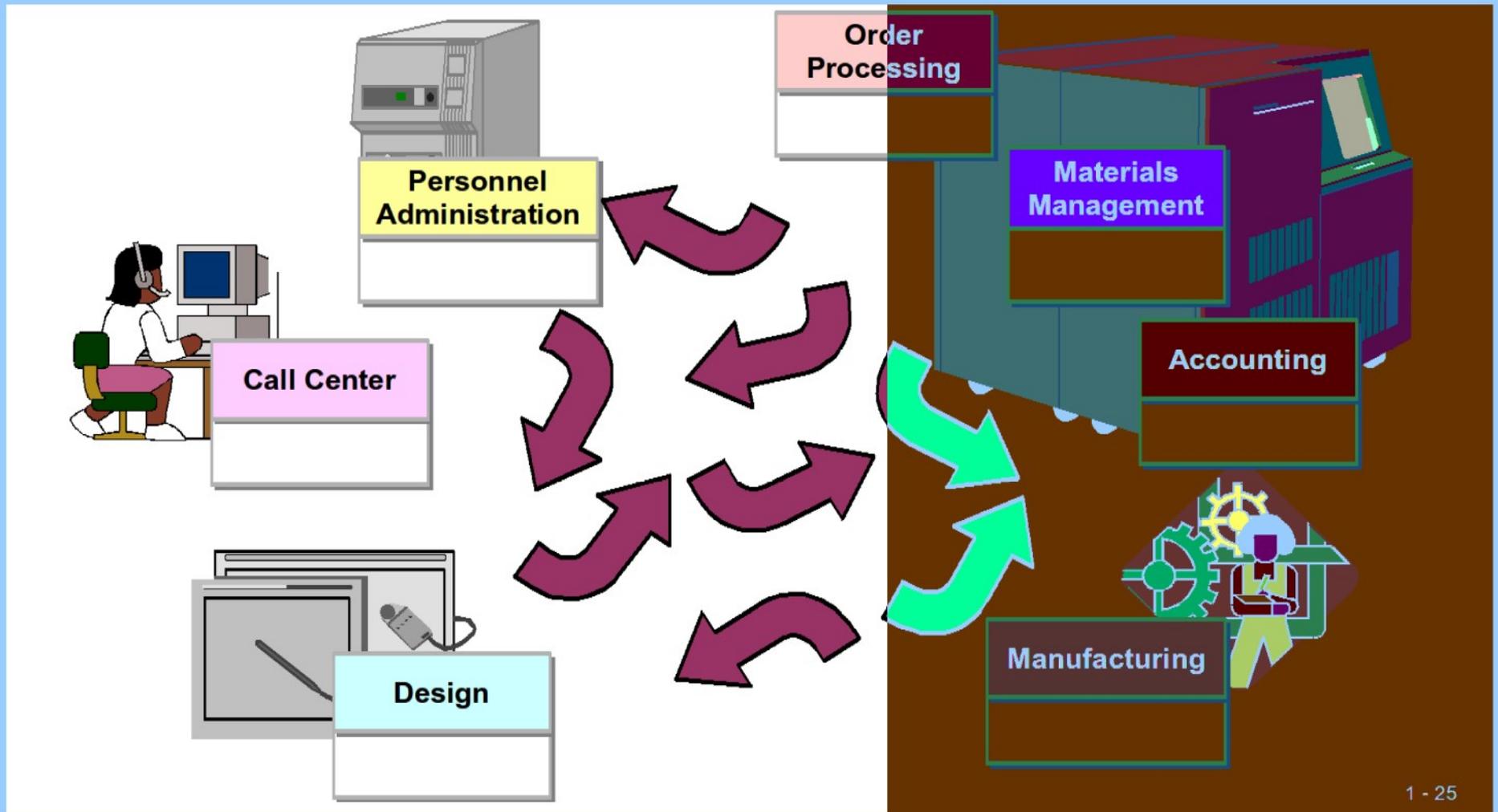
- Неоднозначность моделирования.
- Нет общепринятой методологии для моделирования данных в XML, в то время как для реляционной модели и объектно-ориентированной такие средства разработаны и базируются на реляционной алгебре, системном подходе и системном анализе.
- В природе есть множество объектов и явлений, для описания которых разные структуры данных (сетевая, реляционная, иерархическая) являются естественными, и отображение объекта в неестественную для него модель является болезненным для его сути.
- В результате большой гибкости языка и отсутствия строгих ограничений, одна и та же структура может быть представлена множеством способов (различными разработчиками), например, значение может быть записано как атрибут тега или как тело тега и т. д. Например:

```
<a b="1" c="1"/>  
<a b="1" c="1"></a>  
<a><b>1</b><c>1</c></a>  
<a><b value="1"/><c value="1"/></a>  
<a><fields b="1" c="1"/></a> и т. д.[11]
```
- Поддержка многих языков в именовании тегов дает возможность назвать, например вес русским словом, в таком случае компьютер никак не сможет установить соответствия этого поля с полем weight в англоязычной версии программы и с полями в версиях модели объекта на множестве других языков.

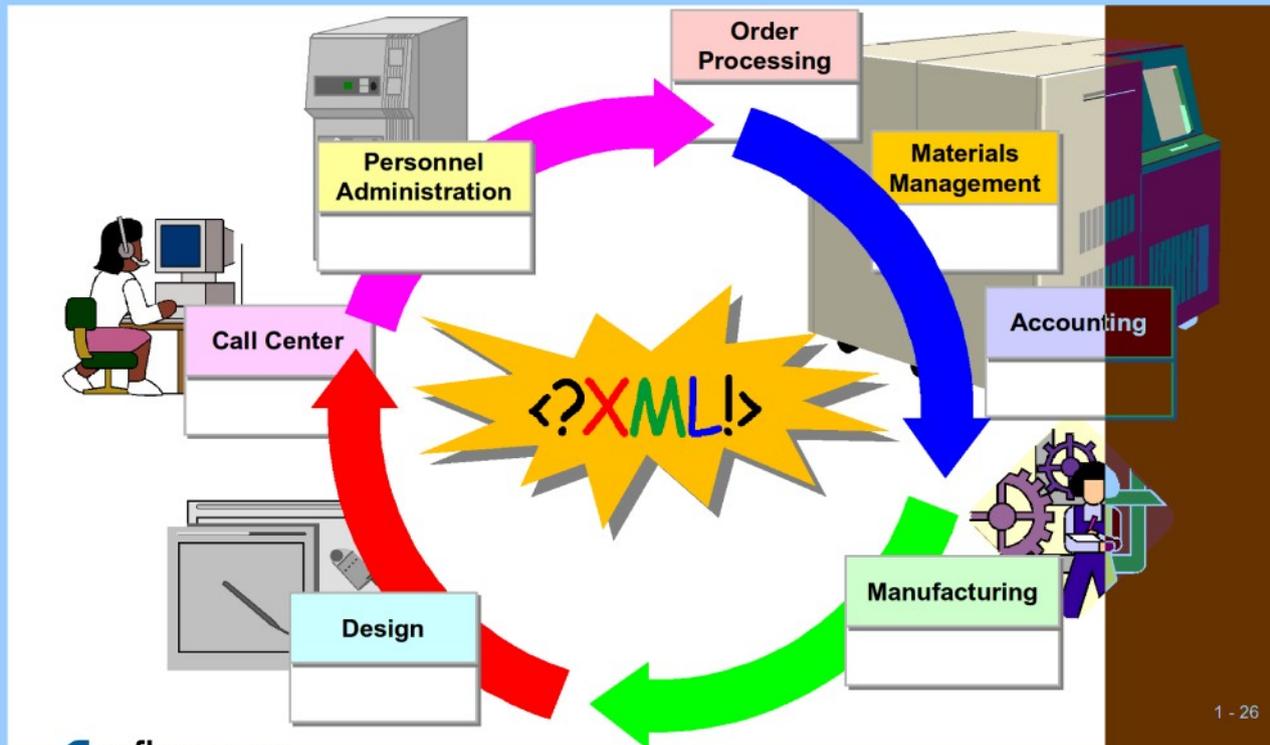
Недостатки

- XML не содержит встроенной в язык поддержки типов данных. В нём нет строгой типизации, то есть понятий «целых чисел», «строк», «дат», «булевых значений» и т. д. Иерархическая модель данных, предлагаемая XML, ограничена по сравнению с реляционной моделью и объектно-ориентированными графами и сетевой моделью данных.
- Выражение неиерархических данных (например графов) требует дополнительных усилий Кристофер Дейт, специалист в области реляционных баз данных, автор классического учебника «An Introduction to Database Systems», отмечал, что «...XML является попыткой заново изобрести иерархические базы данных...»] (в 1980-е года иерархические базы данных были вытеснены реляционными базами данных).
- Пространства имён XML сложно использовать и их сложно реализовывать в XML-парсерах. Существуют другие, обладающие сходными с XML возможностями, текстовые форматы данных, которые обладают более высоким удобством чтения человеком.

Разные форматы обмена



Единый формат обмена



Структура документа

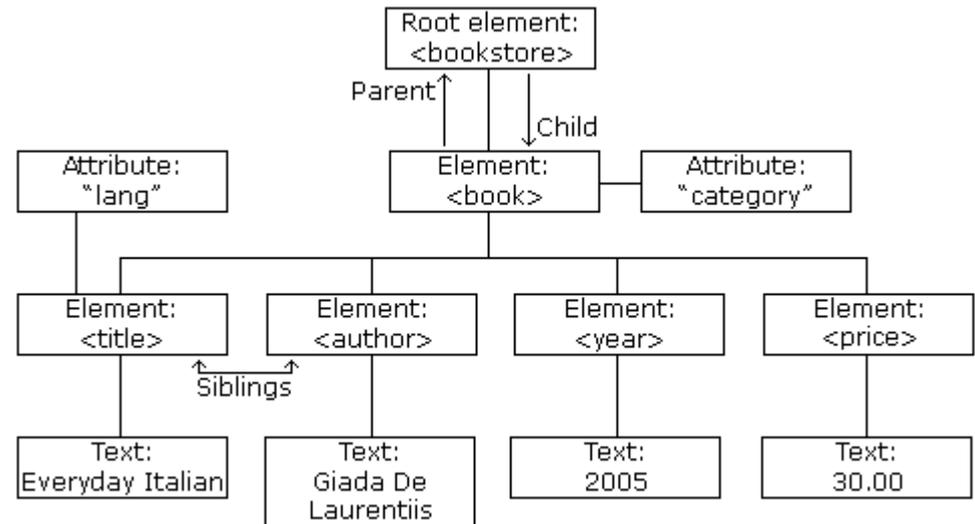
- Заголовок (XML декларация)
- Корневой элемент
- Элементы
- Атрибуты
- Данные
- Комментарии

```
<?xml version="1.0" encoding="windows-1251" ?>
<!-- Пример XML разметки -->
<pricelist>
  <book id="1">
    <title>XML и IE5</title>
    <author>Алекс Гомер</author>
    <price currency="RUR">200</price>
    <exists />
  </book>
</pricelist>
```

- XML декларация
- Комментарий
- Элемент документа (корневой элемент)
- Элемент
- Атрибут
- Текстовые данные

Структура документа

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```



Заголовок

Заголовок (пролог) XML документа должен содержать объявление XML и может содержать объявление типа документа, дополнительную информацию для обработки XML документа.

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

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

- Пролог

Декларация и процессинговые инструкции
DTD or XML схема (описание грамматики)

```
<?xml version="1.0"?>  
<!DOCTYPE employees SYSTEM "employees.dtd">
```

- Корневой элемент содержит все данные

```
<employees>  
  <name>Lars Peterson</name>  
  ...  
</employees>
```

Теги

1. Теги заключаются в угловые скобки '<','>'.
<text> <name> <MyText>

2. Все теги должны быть закрыты. Тег закрывается одноимённым тегом с '/' перед именем, либо '/' перед закрывающей скобкой.

<text></text>

<text />

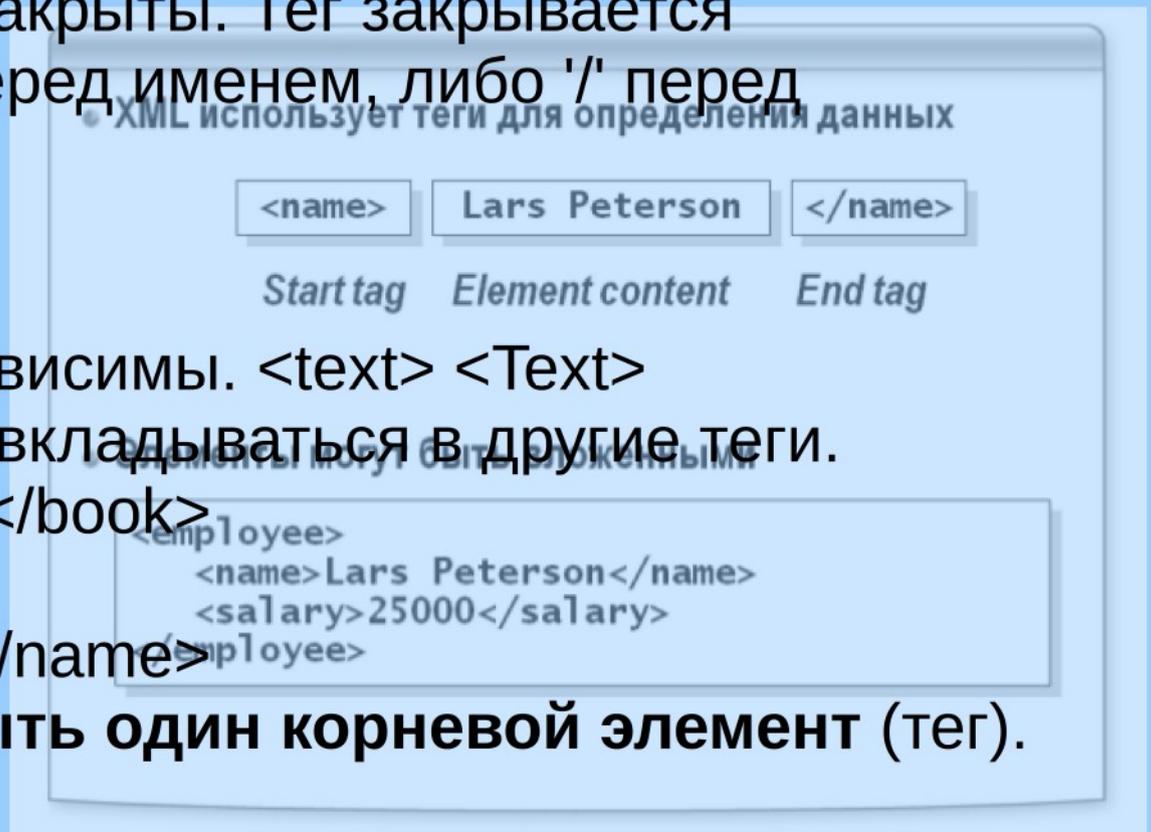
3. Имена тегов регистрозависимы. <text> <Text>

4. Теги должны корректно вкладываться в другие теги.

<book><name></name></book>

<book><name></book></name>

5. У документа **должен быть один корневой элемент (тег).**



Теги

6. Атрибуты элементов должны быть заключены в двойные кавычки " .

7. Некоторые символы в XML нельзя использовать. Для них есть специальные коды:

<	<	less than
>	>	greater than
&	&	ampersand
'	'	apostrophe
"	"	quotation mark

8. Комментарии заключаются в `<!-- текст комментария -->`

9. XML в отличие от HTML пробелы не "сжимает".

10. Завершение строки в XML документах записывается одним байтом (LF, 0x0A), как в Unix-ах, а не как в Windows (два байта).

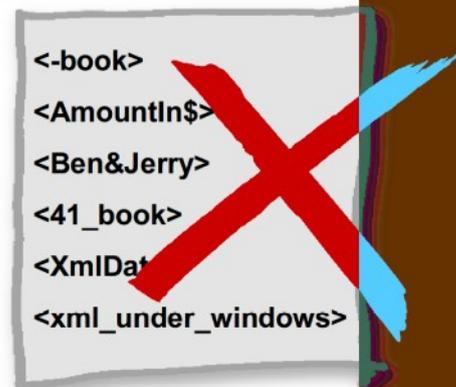
Теги

- Имена могут содержать символы (A-Z, a-z, числа);
- Имена не могут начинаться с числа или знака пунктуации;
- Имена не могут начинаться со слова xml (в любых вариантах);
- Имена не могут содержать пробелы.



<book>
<BOOK>
<Book-List>
<WROX:books>
<_book41>
<book.catalog>

Valid names



<-book>
<AmountIn\$>
<Ben&Jerry>
<41_book>
<XmlDat
<xml_under_windows>

Invalid names

Пустой тег

```
<temperature scale="C"></temperature>
```

```
<temperature scale="C"/>
```

CDATA

<temperature> <0 /temperature>



<temperature><![CDATA[<0]]:

```
<?xml version="1.0" encoding="UTF-8" ?>
<pricelist>
  <book>
    <title>XML и IE5</title>
    <author>Алекс Гомер</author>
    <price>200</price>
    <description>
      <![CDATA[
        Книга описывает применение XML
        в браузере IE5 <br>
        а также содержит много примеров
      ]]>
    </description>
  </book>
</pricelist>
```

Интернациональные символы

<example>

XML supports international character sets.

This example shows different notations for the number "1":

&#49;	(ASCII),
&#x0661;	(Devanagari),
&#x0967;	(Arabic)
&#x0d67;	(Malayalam)

</example>

Задание 1

- Создать каталог для выполнения заданий (XML)
- Создать в каталоге файл task
- Открыть его в редакторе и содержащий:
 - корневой элемент building;
 - два элемента floor с атрибутами number="1" и number="2" соответственно;
 - На первом этаже добавьте код <room>101</room>
 - На втором этаже добавьте код <room>203</room>

```
<?xml version="1.0" encoding="utf-8"?>
<building>
  <floor number="1">
    <room>101</room>
  </floor>
  <floor number="2">
    <room>203</room>
  </floor>
</building>
```

Правильность XML документов

Стандартом определены два уровня правильности документа XML:

- Правильно построенный (Well-formed);
- Действительный (Valid):
- Допустимый по типу (type-valid XML document)
- Недопустимый по типу (not-type-valid XML document)

DTD

Document Type Definition

- **Язык описания структуры документа**

- **Описывает:**

 - Какие элементы могут присутствовать в документе

 - Повторения элементов

 - Какие атрибуты могут быть у элементов

 - Какие атрибуты обязательны

 - Какие сущности могут применяться

DTD

Способы программного анализа документа

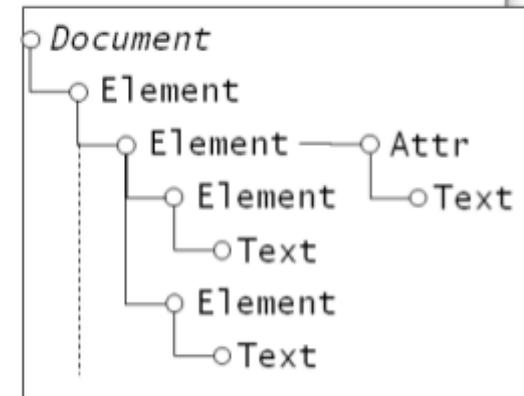
- Для программного анализа используется XML
- Парсер – синтаксический анализатор
- SAX – Simple API of XML
- DOM – document object model

DOM структура

- Парсер представляет документ как иерархию объектов
- Объекты DOM – это узлы (node) связанные друг с другом

Объект Document – основной объект документа

Другие объекты представляют элементы, текст, атрибуты, комментарии и т.д.



Виды DTD

- Встроенный (inline)
- Внешний (external)

external

```
<?xml version="1.0"?>  
<!DOCTYPE weather-report SYSTEM "weather.dtd">
```

Inline DTD

```
<?xml version="1.0"?>  
<!DOCTYPE weather-report  
[  
  <!ELEMENT weather-report  
    (date,time,area,measurements)>  
  <!ELEMENT date      (#PCDATA)>  ....  
>  
<weather-report>  
  <date> ....
```

Пример DTD

```
<!ELEMENT pricelist (book+)>
<!ELEMENT book (title, author+, price,
description?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ATTLIST price currency CDATA #IMPLIED>
```

Описание элементов

- **ELEMENT** – определение элемента
- **ATTLIST** – определение атрибута
- **ENTITY** – определение сущности

- Модификаторы
- * – ноль или много
- ? – ноль или один
- + – один или много

■ EMPTY	no content
■ ANY	no constraints on content
■ 	choice list
■ ,	sequence
■ Cardinality	
■ +	exactly one
■ *	one or more
■ ?	zero or one
■ *	zero or more
■ ()	grouping
■ (#PCDATA)	characters
■ (#PCDATA ...) *	characters or elements ("mixed content")

Подключение DTD

```
<?xml version=1.0" encoding="UTF-8"?>
<!DOCTYPE DOCUMENT [
  <!ELEMENT DOCUMENT (GREETING, ME
  <!ELEMENT GREETING (#PCDATA)>
  <!ELEMENT MESSAGE (#PCDATA)>
]>

<DOCUMENT>
  <GREETING> Hello From XML </GREETING>
    <MESSAGE>
      Welcome to the wild
      and woolly world of
    </MESSAGE>
</DOCUMENT>
```

```
<?xml version = "1.0" standalone="no"?>
<!DOCTYPE DOCUMENT SYSTEM "mydoc.dtd">

<DOCUMENT>
  <CUSTOMER>
    ...
  </CUSTOMER>
</DOCUMENT>
```

Attribute Declaration

<code><!ATTLIST</code>	<code>temperature</code>	<code>scale</code>	<code>CDATA</code>	<code>#IMPLIED></code>
	Element Type	Attribute Name	Attribute Type	Default Value

<code><!ATTLIST</code>	<code>temperature</code>	<code>scale</code>	<code>CDATA</code>	<code>#REQUIRED></code>
<code><!ATTLIST</code>	<code>temperature</code>	<code>scale</code>	<code>(C F)</code>	<code>#REQUIRED></code>
<code><!ATTLIST</code>	<code>temperature</code>	<code>scale</code>	<code>(C F)</code>	<code>"C" ></code>
<code><!ATTLIST</code>	<code>temperature</code>	<code>scale</code>	<code>CDATA</code>	<code>#FIXED "C"></code>

Attribute Type and Default Value

Attribute Type

- ◆ `CDATA` string value
- ◆ `ID` attribute value unique per document
- ◆ `IDREF(S)` attribute value matches ID value
- ◆ `(enumeration)` restricts to values

Default Value

- ◆ `#REQUIRED` value must be provided
- ◆ `#IMPLIED` no constraints
- ◆ `"string"` default value
- ◆ `#FIXED` default is only value

Атрибут ID

```
<!ELEMENT Person (Name, Address?, Email*, ...)>
<!ATTLIST Person perID ID #REQUIRED>
<!ELEMENT Team EMPTY>
<!ATTLIST Team members IDREFS #REQUIRED>
```

```
<Pers-List>
  <Person perID = "A-111">
    <Name>Smith</Name>....
  </Person>
  <Person perID = "A-112">
    <Name>Miller</Name> ....
  </Person>
  ....
  <Team members = "A-111 A-112 B-111 ..."/>
</Pers-List>
```

Элемент или атрибут?

In many use cases, elements and attributes have the same function.

There are, nonetheless, differences between elements and attributes:

Attributes cannot have child elements or other markup.

No default value can be specified for elements.

The content of elements cannot be restricted by a choice list.

The uniqueness of element values within a document cannot be enforced.

Consistency between the contents of elements cannot be enforced.

```
<measurements>
  <skies>fine</skies>
  <temperature scale="C">25</temperature>
  ....
</measurements>
```

```
<measurements>
  <skies>fine</skies>
  <temperature>25</temperature>
  <scale>C</scale>
  ....
</measurements>
```

СУЩНОСТИ

Entity Declaration in the DTD

```
<!DOCTYPE text
[
<!ENTITY sag      "Software AG">
<!ENTITY sagtext SYSTEM
    "http://www.softwareag.com/std.txt">
]>
```

Referencing in the document

```
<text>&sag; is proud to announce .
```

```
<!DOCTYPE purchase-order
[
<!ENTITY Head      SYSTEM "Headsection.xml">
<!ENTITY PositionsPC SYSTEM "Positions/PC1.xml">
<!ENTITY PositionsMonitor SYSTEM "http://monitors.de/m2.xml">
]>
<purchase-order>
  <customer-data> &Head; </customer-data>
  <order>
    <position> &PositionsPC; </position>
    <position> &PositionsMonitor; </position>
  </order>
</purchase-order>
```

Параметры

```
<!DOCTYPE example  
[  
  <!ENTITY % example-entity    "<!ELEMENT example (#PCDATA)>">  
  %example-entity;  
>  
<example>  
</example>
```

```
<!ENTITY % block "paragraph | listing | picture | box">
```

```
<!ELEMENT book (chapter+)>
```

```
<!ELEMENT chapter (heading, (%block;)*, section+)>
```

```
<!ELEMENT section (heading, (%block;)*, subsection*)>
```

```
<!ELEMENT subsection (heading, (%block;)+)>
```

```
<!ELEMENT paragraph (#PCDATA)>
```

...

```
<ENTITY % tab-model SYSTEM "table-model.dtd">
```

```
%tab-model;
```

```
<!ENTITY % block "paragraph | listing | picture | box | table">
```

```
<!ELEMENT book (chapter+)>
```

```
<!ELEMENT chapter (heading, (%block;)*, section+)>
```

```
<!ELEMENT section (heading, (%block;)*, subsection*)>
```

```
<!ELEMENT subsection (heading, (%block;)+)>
```

```
<!ELEMENT paragraph (#PCDATA)>
```

...

Integration of "Non-XML-data"

```
<!ELEMENT logo EMPTY>
```

```
<!ATTLIST logo image ENTITY #REQUIRED>
```

```
<!NOTATION GIF SYSTEM "gifmagic.exe">
```

```
<!ENTITY saglogo SYSTEM "http://www.softwareag.com/logo1.gif"  
NDATA GIF>
```

```
<logo image = "saglogo"/>
```

```
<!ELEMENT comment (#PCDATA)>
```

```
<![INCLUDE[  
  <!ELEMENT comment (#PCDATA)>  
]]>
```

```
<![IGNORE[  
  <!ELEMENT comment (#PCDATA)>  
]]>
```

```
<!ENTITY % design "INCLUDE">  
<!ENTITY % finished "IGNORE" >
```

```
<![%design; [<!ELEMENT comment (#PCDATA)>]]  
<![%finished; [<!ELEMENT comment EMPTY>]]
```

Задание 2

- Создать в файл `task1.dtd` и описать в нем структуру документа;
- Подключить DTD к XML документу;
- Проверить документ на валидность.
- Провести эксперимент с добавлением несуществующего элемента в DTD;
- Провести эксперимент с добавлением неопisanного элемента (в DTD) в XML;
- Провести эксперимент с добавлением атрибута `name` к элементу `room`.

Пример XML-документов

Электронные книги – FB2

<http://www.fictionbook.org>

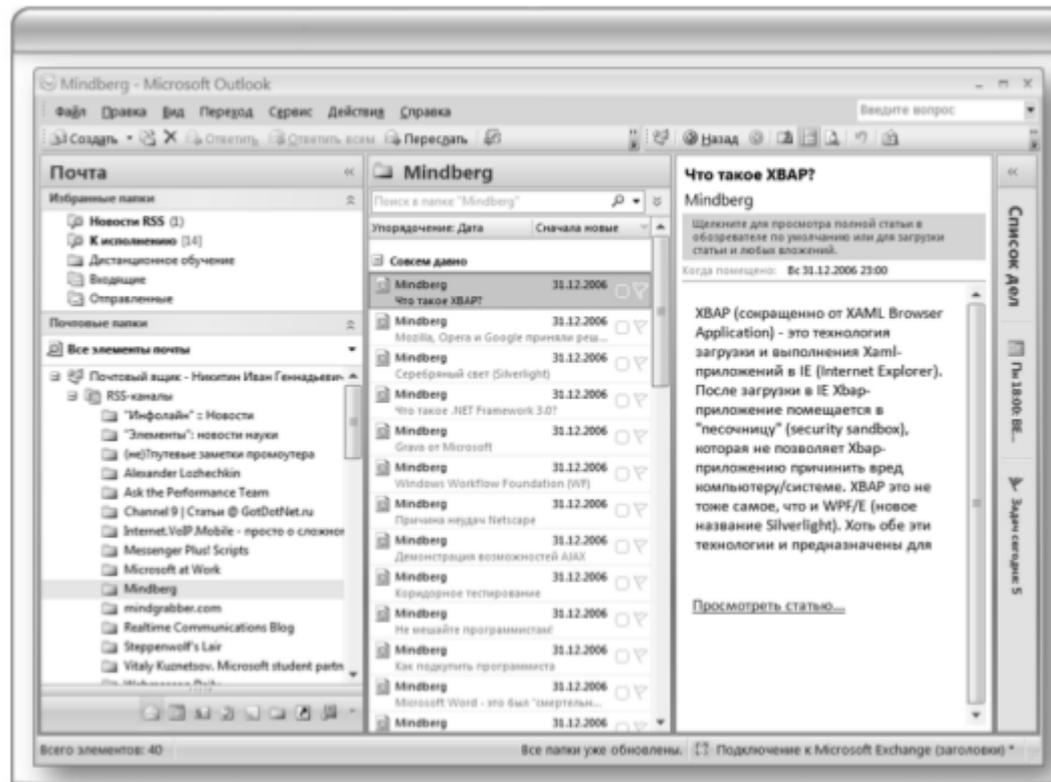
Name	Size	Author	Title	Year
Сергей Лукьяненко				
Сергей Лукьяненко. Близится...	630 KB	Сергей Лукьяненко	Близится утро	1999
Сергей Лукьяненко. Генон	649 KB	Сергей Лукьяненко	Генон	1999
Сергей Лукьяненко. Звездная тень	634 KB	Сергей Лукьяненко	Звездная тень	1999
Сергей Лукьяненко. Звезды – холодные...	634 KB	Сергей Лукьяненко	Звезды – холодные...	1999
Сергей Лукьяненко. Лабиринт Отражения	634 KB	Сергей Лукьяненко	Лабиринт Отражения	1999
Сергей Лукьяненко. Лена Грез	634 KB	Сергей Лукьяненко	Лена Грез	1995
Сергей Лукьяненко. Мой Папа - Антибиотик	634 KB	Сергей Лукьяненко	Мой Папа - Антибиотик	1992
Сергей Лукьяненко. Ночной дозор	634 KB	Сергей Лукьяненко	Ночной дозор	1998
Сергей Лукьяненко. Пограничное время	368 KB	Сергей Лукьяненко	Пограничное время	2002
Сергей Лукьяненко. СПЕКТР. (КАЖДЫЙ ОХОТНИК ЖЕЛАЕТ ЗНАТЬ)	991 KB	Сергей Лукьяненко	СПЕКТР. (КАЖДЫЙ ОХОТНИК ЖЕЛАЕТ ЗНАТЬ)	2002
Сергей Лукьяненко. Танцы на снегу	711 KB	Сергей Лукьяненко	Танцы на снегу	2000
Сергей Лукьяненко. Тени Снов	155 KB	Сергей Лукьяненко	Тени Снов	1998
Сергей Лукьяненко, Ник Перумов				
Сергей Лукьяненко, Ник Перумов. Не время для дракона	842 KB	Сергей Лукьяненко, Ник Перумов	Не время для дракона	декабрь 1995 - июль 1996
Станислав Лем				
Станислав Лем. Магелланово облако	762 KB	Станислав Лем	Магелланово облако	1955
Станислав Лем. Сумма технологий	1 272 KB	Станислав Лем	Сумма технологий	1967
Стивен Кинг				
Стивен Кинг. Календарь ипостасей	1 410 KB	Стивен Кинг	Календарь ипостасей	1999

II. Открытие профессора Декантора

Лет шесть назад, по возвращении из путешествия, когда безделье и наслаждение наивным миром домашней жизни уже приелось, – не настолько, однако, чтоб я подумал о новой экспедиции, – поздним вечером, когда я никого не ждал, ко мне пришел какой-то человек и оторвал меня

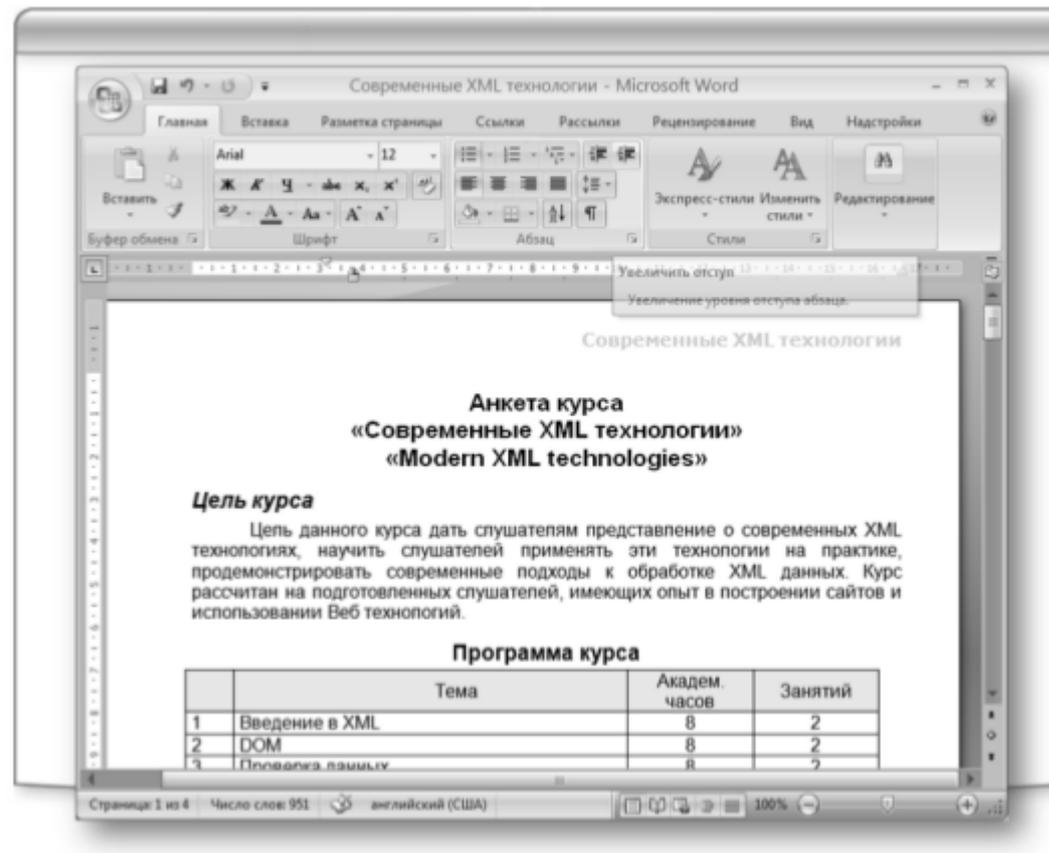
Пример XML-документов

RSS – новости



Пример XML-документов

Документы OpenOffice и Microsoft Office 2007



The screenshot displays the Microsoft Word 2007 interface with a document titled "Современные XML технологии - Microsoft Word". The ribbon shows the "Главная" (Home) tab. The document content includes a title, a course goal, and a program table.

Современные XML технологии

**Анкета курса
«Современные XML технологии»
«Modern XML technologies»**

Цель курса

Цель данного курса дать слушателям представление о современных XML технологиях, научить слушателей применять эти технологии на практике, продемонстрировать современные подходы к обработке XML данных. Курс рассчитан на подготовленных слушателей, имеющих опыт в построении сайтов и использовании Веб технологий.

Программа курса

	Тема	Академ. часов	Занятий
1	Введение в XML	8	2
2	DOM	8	2
3	Проверка данных	8	2

Страница 1 из 4 Число слов: 951 английский (США) 100%

Пространство имен

- 1) В различных документах могут быть одинаковые имена элементов
- 2) Документы XML могут включаться друг в друга и это может привести к конфликтам в именах.

- Пространства имен устраняют конфликты элементов
- Пространства имен объявляются с помощью Uniform Resource Identifier (URI)
- URI может быть URL или URN

URL: <http://www.w3.org/1999/XSL/Transform>

URN: <urn:schemas-microsoft-com:xml-data>

Пространство имен

- **Пространство имен объявляется с помощью атрибута xmlns**

```
<payment xmlns="http://www.microsoft.com/payroll">  
  <customer>1234</customer>  
</payment>
```

- **Или с помощью префикса**

```
<a:payment xmlns:a="http://www.microsoft.com/payroll">  
  <customer>1234</customer>  
</a:payment>
```

"Перегрузка" пространств имен

```
<order xmlns="http://www.net-standard.com/namespaces/order"
        xmlns:ord="http://www.net-standard.com/namespaces/order"
        xmlns:bk="http://www.net-standard.com/namespaces/books"
        xmlns:cust="http://www.net-standard.com/namespaces/customer">
  <ord:description xmlns="http://www.w3.org/1999/xhtml">
    <html>
      <head><title>Order</title></head>
      <body>
        <h1>Book order</h1>
        <p> The order of the following books is made by phone...</p>
      </body>
    </html>
  </ord:description>
  <order-date>20.10.1999</order-date>
  <order-number>55544-3</order-number>
  <bk:book>
    <bk:title>XML Handbook</bk:title>...
  </bk:book>
  ...
</order>
```

Пустое пространство имен

```
<?xml version="1.0"?>
<order xmlns="http://www.net-standard.com/namespaces/order"
        xmlns:bk="http://www.net-standard.com/namespaces/books"
        xmlns:cust="http://www.net-standard.com/namespaces/customer"
>
  <order-date>20.10.1999</order-date>
  <order-number>55544-3</order-number>
  <bk:book>
    <bk:isbn>0130811521</bk:isbn>
  </bk:book>
  <cust:customer>
    <comment xmlns="">This comment belongs to no namespace</comment>
    <cust:forename>Paul</cust:forename>
    <cust:lastname>Smith</cust:lastname>
    <cust:address> ... </cust:address>
    <cust:phone> ... </cust:phone>
    <cust:email> ... </cust:email>
  </cust:customer>
</order>
```

Задача 3

- Указать пространство имен `study` для документа

Задача 4

- Включить в информацию о здании элемент book, который должен содержать элементы из task04-add.xml
- Добавить в XML документ еще одно пространство имен
- Доработать DTD

Источники

1. Эдди С.Э. XML: справочник — СПб.: Питер, 2000
2. Материалы курса IT 581
3. Материалы курса по XML компьютерного центра обучения "Специалист" при МГТУ им. Н.Э. Баумана
4. Материалы W3Schools (<http://www.w3schools.com/xml/>)
5. Википедия (<http://ru.wikipedia.org>)
6. Наварро Э. XHTML: учебный курс — СПб.: Питер, 2001