Лабораторная работа № 11

Тема: Описание структуры XML-документа средствами DTD

*Цель работы:*формирование практических умений описания структуры XML-документа средствами DTD.

***Теоретические сведения***

В XML-документах DTD определяет набор действительных элементов, идентифицирует элементы, которые могут находиться в других элементах, и определяет действительные атрибуты для каждого из них. В XML использовать DTD не обязательно – документы, созданные без этих правил, будут правильно обрабатываться программой-анализатором, если они удовлетворяют основным требованиям синтаксиса XML. Однако контроль за типами элементов и корректностью отношений между ними в этом случае будет полностью возлагаться на автора документа. До тех пор, пока грамматика нашего нового языка не описана, его сможем использовать только мы, и для этого мы будем вынуждены применять специально разработанное программное обеспечение, а не универсальные программы-анализаторы.

В DTD для XML используются следующие типы правил:

- правила для элементов и их атрибутов;

- описания категорий (макроопределений);

- описание форматов бинарных данных.

Все они описывают основные конструкции языка – элементы, атрибуты, символьные константы, внешние файлы бинарных данных.

Для того чтобы использовать DTD в документе, можно или описать его во внешнем файле и при описании DTD просто ука­зать ссылку на этот файл, или же непосредственно внутри са­мого документа выделить область, в которой определить нуж­ные правила. В первом случае в документе указывается имя файла, содержащего DTD- описания:

*<?xml version=“1.0” standalone=“no” ?>*

*<! DOCTYPE documents SYSTEM «def.dtd»>*

…

Внутри документа DTD-декларации включаются следую­щим образом:

*…*

*<! DOCTYPE documents [*

*<!ELEMENT documents (books+, articles+)>*

*…*

]>

…

Значение *standalone* определяет, может ли этот документ быть обработан без чтения каких-либо других файлов. Напри­мер, если XML-документ не ссылается на другие файлы, вы должны указать *standalone=“yes”.* Если же XML-документ ссы­лается на другие файлы, которые описывают, что документ мо­жет содержать, нужно указать*standalone=“no”*.

Атрибут *SYSTEM* указывает на то, что *def.dtd* находится на локальном сервере.

В том случае, если используются одновременно внутренние и внешние описания, то программой-анализатором будут сна­чала рассматриваться внутренние, то есть их приоритет выше. При проверке документа XML-процессор в первую очередь ищет DTD внутри документа. Если правила внутри документа не определены и не задан атрибут *standalone =“yes”* , то программа загрузит указанный внешний файл и правила, находящиеся в нем, будут считаны оттуда. Если же атрибут *standalone*имеет значение *yes*, то использование внешних DTD-описаний будет запрещено.

Элемент в DTD определяется с помощью дескриптора *!ELEMENT*, в котором указывается название элемента и струк­тура его содержимого.

Например, для элемента *<author>* можно определить сле­дующее правило:

*<!ELEMENT author (#PCDATA)>*

В определении элемента нужно указать сначала название элемента(*author*), а затем модель его содержимого – другие эле­менты или типы данных, которые могут встречаться внутри него. В данном случае содержимое элемента *author* будет опре­деляться при помощи специального маркера #*PCDATA* (что оз­начает *parseable character data*– любая информация, с которой может работать программа-анализатор). Существуют еще две инструкции, определяющие тип содержимого: *EMPTY* и *ANY*. Первая указывает на то, что элемент должен быть пустым (на­пример, <*red*/>), вторая – на то, что содержимое элемента специ­ально не описывается.

Последовательность дочерних для текущего элемента объ­ектов задается в виде списка названий элементов, разделенных запятыми. При этом для того, чтобы указать количество повто­рений включений этих элементов, могут использоваться симво-лы «+», «?».

Например:

*<!ELEMENT book (author+, title, translate?, year\_publica-tion)>*

При этом определении элемент <*book*> должен включать элементы *author, title, translate и year\_publication*, причем элементы *title* и *year\_publication* являются обязательными и могут встречаться лишь однажды, элемент *author*может встречаться несколько раз, а элемент *translate* является опциональным, то есть может отсутствовать. В том случае, если существуют несколько возможных вариантов содержимого определяемого элемента, их следует разделять при помощи символа «|».

Например:

*<!ELEMENT flower (#PCDATA | title)\*>*

Символ «\*» указывает на то, что определяемая последовательность внутренних элементов может быть повторена несколько раз или же совсем не использоваться.

Если в определении элемента указывается «смешанное» со­держимое (текстовые данные или набор элементов), то необ­ходимо сначала указать *PCDATA*, а затем разделенный символом «|» список элементов.

Списки атрибутов элемента определяются с помощью клю­чевого слова *!ATTLIST*. Внутри него задаются названия атрибу­тов, типы их значений и дополнительные параметры.

Например, для элемента <*name*>

*<name first= “Иван” second= “Петрович” surname= “Си­доров”>*

атрибуты могут быть определены следующим образом:

*<!ATTLIST name first CDATA #REQUIRED second CDATA #IMPLIED surname CDATA #REQUIRED>*

Всего существуют шесть возможных типов значений атри­бута:

*CDATA* – содержимым документа могут быть любые сим­вольные данные;

*NMTOKEN*(*NMTOKENS*) – содержимым элемента может быть только одно отдельное слово (этот параметр является ограниченным вариантом *CDATA*);

*ID* – определяет уникальный идентификатор элемента в документе;

*IDREF* (*IDREFS*) – указывает, что значением атрибута должно выступать название (или несколько таких названий, разделенных пробелами во втором случае) уникального идентифи­катора определенного в этом документе элемента;

*ENTITY* (*ENTITIES*) – значение атрибута должно быть на­званием (или списком названий, если используется *ENTITIES*) компонента (макроопределения), определенного в документе;

*Список допустимых значений* – определяется список зна­чений, которые может иметь данный атрибут (в скобках без ка­вычек).

В определении атрибута также можно использовать сле­дующие параметры:

*#REQUIRED*– определяет обязательный атрибут, который должен быть задан во всех элементах данного типа;

*#IMPLIED*– атрибут не является обязательным;

*#FIXED «значение»* – указывает, что атрибут должен иметь только указанное значение, однако само определение атрибута не является обязательным, но в процессе разбора его значение в любом случае будет передано программе-анализатору;

*Значение*– задает значение атрибута по умолчанию.

Рассмотрим XML-документ с DTD-определением (пример 2.1).

**Пример 2.1**(*books\_dtd.xml*)

*<?xml version=“1.0” encoding=“Windows-1251”?>*

*<!DOCTYPE documents SYSTEM [*

*<!ELEMENT documents (books+, articles+)>*

*<!ELEMENT books (book\*)>*

*<!ELEMENT articles (article\*)>*

*<!ELEMENT book (author+, title, translate?, year\_publi-cation)>*

*<!ELEMENT article (author+, title, year\_publication)>*

*<!ATTLIST book id ID #REQUIRED>*

*<!ATTLIST article id ID #REQUIRED>*

*<!ELEMENT author (#PCDATA)>*

*<!ELEMENT title (#PCDATA)>*

*<!ELEMENT translate (#PCDATA)>*

*<!ELEMENT year\_publication (#PCDATA)>*

*]>*

*<documents>*

*<books>*

*<book id=“Book1” >*

*<author>Эрих Мария Ремарк</author>*

*<title>Тритоварища</title>*

*<translate>ИвановаИ.В. </translate>*

*<year\_publication>2000</year\_publication>*

*</book>*

*<book id=“Book2” >*

*<author>РэйБрэдбери</author>*

*<title>Вино из одуванчиков</title>*

*<translate>ПетроваА.К. </translate>*

*<year\_publication>2009</ year\_publication>*

*</book>*

*</books>*

*<articles>*

*<article id=“Article1” >*

*<author>Виктор Кожевников</author>*

*<title>Классификация электронных средств обучения</title>*

*<year\_publication>1998</year\_publication>*

*</article>*

*</articles>*

*</documents>*

Довольно часто при создании XML-элемента разработчику требуется определить, данные какого типа могут использоваться в качестве его содержимого.

Например, некоторый элемент *<last-modified>10.10.98</last-modified>* должен содержать строку, представляющую собой дату, а не число или произвольную последовательность символов.

С помощью типизации данных можно создавать элементы, значения которых могут использоваться, например, в качестве параметров SQL-запросов. Программа-клиент в этом случае должна знать, к какому типу данных относится текущее значе­ние элемента, и в случае соответствия формирует SQL-запрос.

Если в качестве программы на стороне клиента использу­ется верифицирующий XML-процессор, то информацию о типе можно передавать при помощи специально созданного для этого атрибута элемента, имеющего соответствующее DTD-определе­ние. В процессе разбора программа-анализатор передаст значе­ние этого атрибута клиентскому приложению, которое сможет использовать эту информацию должным образом. Например, чтобы указать, что содержимое элемента должно быть длинным целым, можно использовать следующее DTD-определение:

*<!ELEMENT counter (#PCDATA)>*

*<!ATTLIST counter data\_long CDATA #FIXED «LONG»>*

Задав атрибуту значение по умолчанию *LONG* и определив его как *FIXED*, мы, тем самым, позволили программе-клиенту получить необходимую информацию о типе содержимого дан­ного элемента, и теперь она может самостоятельно определить соответствие типа этого содержимого указанному в DTD-опре­делении.

В XML-документе, приведенном в примере 2.2, определя­ются и используются несколько элементов с различными типами данных:

**Пример 2.2**(*def.xml*)

*<?xml version=“1.0” encoding=“Windows-1251”?>*

*<!DOCTYPE houses SYSTEM [*

*<!ELEMENT price (#PCDATA)>*

*<!ATTLIST price data\_currency CDATA #FIXED «CURRENCY»>*

*<!ELEMENT rooms\_num (#PCDATA)>*

*<!ATTLIST rooms\_num data\_byte CDATA #FIXED «BYTE»>*

*<!ELEMENT floor (#PCDATA)>*

*<!ATTLIST floor data\_byte CDATA #FIXED «INTEGER»>*

*<!ELEMENT living\_space (#PCDATA)>*

*<!ATTLIST living\_space data\_float CDATA #FIXED «FLOAT»>*

*<!ELEMENT counter (#PCDATA)>*

*<!ATTLIST counter data\_long CDATA #FIXED «LONG»>*

*<!ELEMENT is\_tel (#PCDATA)>*

*<!ATTLIST is\_tel data\_bool CDATA #FIXED «BOOL»>*

*<!ELEMENT houses (house\*)>*

*<!ELEMENT house (rooms\_num, floor,living\_space, is\_tel, counter, price)>*

*<!ATTLIST house id ID #REQUIED>*

*]>*

*<houses>*

*…*

*<house id=«0»>*

*<rooms\_num>5</rooms\_num>*

*<floor>2</floor>*

*<living\_space>32.5</living\_space>*

*<is\_tel>true</is\_tel>*

*<counter>18346</counter>*

*<price>34 р. 28 к.</price>*

*</house>*

*…*

*</houses>*

Как видно из примера, механизм создания элементов доку­мента при этом нисколько не изменился. Вся необходимая для проверки типов данных информация заложена в определении элементов внутри блока DTD.

**Задание**

Выполните DTD-описание XML-документа, созданного в лабораторной работе 10, которое должно включать: описание элементов; описание атрибутов элементов; определение типа эле­ментов.

**Контрольные вопросы**

1. Для чего используется DTD?

2. Как определить элемент в DTD?

3. Как определить атрибут в DTD?

4. Какие параметры могут использоваться в определении ат-рибута?

5. Как определить тип элемента?

6. Как связать XML-документ c DTD-определением?