

Chương 2 : Đặc tả nội dung & cấu trúc tài liệu XML

Mục tiêu :

- Trình bày các khả năng của công nghệ Xml khi được ứng dụng trong giai đoạn thiết kế thành phần dữ liệu
- Rèn luyện kỹ năng đặc tả tài liệu Xml biểu diễn thông tin của đối tượng trong thực tế
- Bước đầu ứng dụng Xml trong ứng dụng nhỏ, đơn giản

I. Mở đầu

1. Nội dung tài liệu XML

Nội dung của tài liệu XML bao gồm 2 phần

- + **Nội dung chính:** Hệ thống các thẻ đánh dấu (có hay không có nội dung) tương ứng với các thông tin cần biểu diễn.
- + **Nội dung phụ:** Hệ thống các thẻ khác có ý nghĩa bổ sung, tăng cường một số thông tin về tài liệu XML. Các thẻ này có tác dụng giúp cho việc sử dụng, xử lý trên tài liệu XML tốt hơn trong một số trường hợp nhất định

Các thẻ bên trong nội dung phụ bao gồm loại sau:

- Thẻ khai báo tham số
- Thẻ chỉ thị xử lý
- Thẻ ghi chú
- Thẻ CDATA
- Thẻ khai báo cấu trúc
- Thẻ khai báo thực thể

Các thẻ khai báo tham số, thẻ chỉ thị xử lý, thẻ ghi chú và thẻ CDATA có ý nghĩa sử dụng đơn giản sẽ được diễn giải chi tiết ngay trong phần sau

Thẻ khai báo cấu trúc liên quan đến cấu trúc tài liệu XML với nhiều khái niệm khác. Thẻ này sẽ được trình bày chi tiết trong phần “Đặc tả cấu trúc với DTD”

Thẻ khai báo thực thể liên quan đến nhiều kỹ thuật khác nhau có thể áp dụng trên tài liệu XML. Thẻ này sẽ được trình bày chi tiết trong phần “Kỹ thuật đặc tả nội dung tài liệu XML”

a) Thẻ khai báo tham số

Thẻ khai báo tham số cho phép mô tả thêm một số thông tin chung (tham số) về tài liệu XML ngoài các thông tin đã biểu diễn trong nội dung chính.

Dạng khai báo chung như sau

```
<?xml Ten_1="Gia_tri_1" Ten_2="Gia_tri_2" ... ?>
```

Ten_1, Ten_2, ... là các tên của các tham số và Gia_tri_1, Gia_tri_2, ... là các giá trị tương ứng. Cho đến hiện nay có 3 tham số được dùng là version, encoding, và standalone. Tham số version bắt buộc phải có nếu các tham số khác được sử dụng

Tham số version : Khai báo về phiên bản của định chuẩn XML được sử dụng

Ví dụ :

Tài liệu XML thuộc định chuẩn 1.0

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

Tham số encoding : Khai báo về cách mã hóa các ký tự trong tài liệu

Ví dụ

Tài liệu XML sử dụng cách mã hóa Unicode ký hiệu utf-8

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

Tài liệu XML sử dụng cách mã hóa Unicode ký hiệu utf-16

<?xml version="1.0" encoding="utf-16" ?>

Tham số standalone : Khai báo về liên kết của tài liệu XML và các tài liệu khác. Tham số này chỉ có 2 giá trị hợp lệ là “yes” , “no”. Giá trị định sẵn là “no”

Ví dụ :

Tài liệu XML có liên kết với các tài liệu khác

<?xml standalone="yes" ?>

Tài liệu XML không có liên kết với các tài liệu khác

<?xml version="1.0" standalone="no" ?>

b) Thẻ chỉ thị xử lý

Ý nghĩa chung của các thẻ chỉ thị xử lý là cho phép mô tả thêm một số thông tin (liên quan xử lý) về tài liệu XML có ý nghĩa riêng với một công cụ xử lý nào đó. Đây chính là một phương pháp cho phép mở rộng, bổ sung các xử lý riêng vào một lớp tài liệu XML cùng thuộc một hệ thống phân lớp nào đó

Dạng khai báo chung như sau

<? Bo_xu_ly Du_lieu ?>

Bo_xu_ly là ký hiệu của bộ xử lý sẽ tiến hành một số xử lý nào đó trên tài liệu XML . Du_lieu là thông tin được gửi đến Bo_xu_ly

Ví dụ:

<?xml-stylesheet type="text/css" href="Dinh_dang.css" ?>

Là thẻ chỉ thị cần xử lý định dạng thể hiện tài liệu XML với “chương trình định dạng ” theo ngôn ngữ css được lưu trữ bên trong tập tin Dinh_dang.css

Thẻ này sẽ có ý nghĩa với một số trình duyệt Web như IE (phiên bản 5.0 về sau), Netscape (phiên bản 6.0 về sau)

c) Thẻ ghi chú

Thẻ ghi chú cho phép bổ sung các thông tin ghi chú có ý nghĩa đối với con người và hoàn toàn không có ý nghĩa với các hệ thống xử lý tài liệu XML

Dạng khai báo chung như sau:

<-- Nội dung ghi chú -->

d) Thẻ CDATA

Thẻ CDATA có ý nghĩa yêu cầu các bộ phân tích tài liệu XML bỏ qua và không phân tích vào nội dung bên trong của thẻ này. Tác dụng của thẻ là cho phép sử dụng trực tiếp bên trong thẻ một số ký hiệu không được phép nếu sử dụng bên ngoài (ví dụ các ký tự “<” , “>” , ...)

Dạng khai báo chung như sau:

<![CDATA [Nội dung]]>

2. Cấu trúc tài liệu XML

Khái niệm về cấu trúc tài liệu XML

- Chỉ tương ứng cấu trúc của nội dung chính
- Cách thức tổ chức, sắp xếp của các thẻ (có hay không có nội dung) trong nội dung chính

Khái niệm về đặc tả cấu trúc tài liệu XML

- Mô tả ngắn gọn, chính xác cấu trúc tài liệu XML
- Mô tả ngắn gọn, chính xác cách thức tổ chức, sắp xếp của các thẻ

a) Ngôn ngữ đặc tả cấu trúc

Có rất nhiều ngôn ngữ đặc tả được đề xuất để mô tả cấu trúc tài liệu Xml như DTD, XML Schema, XML-Data, Schematron, RELAX NG, v.v.. Trong số đó có 2 ngôn ngữ thông dụng là DTD, XML Schema

+ Đặc điểm của DTD

- Ra đời rất sớm
- Cho phép mô tả văn bản có cấu trúc bất kỳ
- Đơn giản, dễ học và sử dụng
- Chỉ cho phép đặc tả một số “kiểu dữ liệu đơn giản” trong nội dung chính của tài liệu XML

+ Đặc điểm của XML Schema

- Được đề xuất bởi W3C
- Chỉ áp dụng cho tài liệu XML
- Khó học và sử dụng so với DTD
- Cho phép đặc tả chi tiết về các “kiểu dữ liệu” được sử dụng trong nội dung chính của tài liệu XML

Ví dụ: Với tài liệu Xml

```
<?xml version="1.0" encoding="utf-8"?>
<PHAN_SO>
  <Tu_so> 4 </Tu_so>
  <Mau_so> 3 </Mau_so>
</PHAN_SO>
```

Đặc tả với DTD

```
<!DOCTYPE PHAN_SO [
<!ELEMENT PHAN_SO (Tu_so, Mau_so) >
<!ELEMENT Tu_so #PCDATA >
<!-- Tu_so : Số nguyên // >0 -->
<!ELEMENT Mau_so #PCDATA>
<!-- Mau_so : Số nguyên // >0 -->
]>
```

Đặc tả với Xml Schema

```
<?xmlversion="1.0"encoding="utf-8"?>
<xs:schemaid="PHAN_SO" targetNamespace="http://tempuri.org/PHAN_SO.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:elementname="PHAN_SO" type="PHAN_SO"/>
<xs:complexTypename="PHAN_SO">
<xs:sequence>
  <xs:elementname="Tu_so" type="SO_NGUYEN_DUONG" minOccurs="1"maxOccurs="1"/>
  <xs:elementname="Mau_so" type="SO_NGUYEN_DUONG" minOccurs="1"maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
```

```
<xs:simpleTypeName="SO_NGUYEN_DUONG">
  <xs:restrictionbase="xs:int">
    <xs:minExclusivevalue="0"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

b) Sử dụng đặc tả cấu trúc

Có 2 trường hợp chính cần thiết sử dụng các tài liệu đặc tả cấu trúc

- + Trường hợp 1 : Sử dụng cho việc trao đổi thông tin người – người
- + Trường hợp 2 : Sử dụng cho việc trao đổi thông tin người – hệ thống xử lý

Trường hợp 1 là trường hợp thông dụng nhất, với trường hợp này tài liệu đặc tả cấu trúc

- Được sử dụng như phương tiện giao tiếp giữa các chuyên viên tin học có liên quan đến tài liệu XML tương ứng
- Có thể được lưu trữ theo bất kỳ định dạng nào thích hợp cho việc sử dụng (trình bày, xem báo cáo , v.v..)

Ví dụ : Có thể sử dụng các tài liệu đặc tả cấu trúc (DTD/ XML Schema trên) trong

- Hồ sơ thiết kế phần mềm
- Tài liệu mô tả cách thức trao đổi thông tin giữa các chuyên viên tin cùng xây dựng các phần mềm bài tập phân số

Ghi chú :

- Đây là trường hợp sử dụng chính và thông dụng nhất
- Đây là trường hợp dễ sử dụng nhất vì không yêu cầu thật chặt chẽ về cú pháp. Nếu trong tài liệu đặc tả cấu trúc có sai sót một ít về cú pháp thì người sử dụng cũng có thể hiểu hay cũng có thể phát hiện và trao đổi lại với người tạo lập
- Với trường hợp này, tùy vào từng trường hợp cụ thể với các một số qui ước riêng mang tính cục bộ trong một nhóm người nào đó, có thể mở rộng các ngôn ngữ đặc tả cấu trúc hiện có để bổ sung thêm các từ vựng, cú pháp và ngữ nghĩa riêng

Trường hợp 2 chỉ được sử dụng khi

- Có hệ thống xử lý (phần mềm, hàm , đối tượng thư viện) “hiểu” và thực hiện các xử lý tương ứng nào đó với tài liệu đặc tả cấu trúc (xử lý thông dụng nhất là kiểm tra một tài liệu XML có theo đúng cấu trúc được mô tả trong tài liệu đặc tả cấu trúc hay không.)
- Thật sự có nhu cầu cần đến các xử lý của hệ thống xử lý nói trên

II. Một số kỹ thuật đặc tả nội dung

1. Sử dụng thẻ thực thể

Ý nghĩa chung các thẻ khai báo thực thể là cho phép tài liệu XML tham chiếu đến một tập hợp các giá trị đã chuẩn bị trước dưới dạng một tên gọi nhớ (tên thực thể).

Mỗi cách thức tham chiếu và “loại” của tập hợp giá trị được tham chiếu tương ứng với một ý nghĩa/mục tiêu (dạng sử dụng) riêng và sẽ yêu cầu dạng thẻ khai báo thực thể thích hợp.

Có 4 dạng sử dụng chính các thực thể

- Dạng 1 : Tham chiếu đến một chuỗi giá trị bên trong tài liệu XML đang xem xét
- Dạng 2 : Tham chiếu đến các ký tự đặc biệt được định nghĩa trước
- Dạng 3 : Tham chiếu đến một tập hợp các giá trị bên ngoài tài liệu
- Dạng 4 : Tham chiếu đến một tài liệu XML khác

Cách thức khai báo và sử dụng chung các thẻ khai báo thực thể (cho cả 4 dạng trên) như sau:

Khai báo

```
<!DOCTYPE Ten_goc [  
    Khai báo thực thể X  
    Khai báo thực thể Y  
>
```

Sử dụng

&X; <-- Sử dụng tham chiếu của X -->

&Y; <-- Sử dụng tham chiếu của Y -->

a) Dạng 1

Tham chiếu đến một chuỗi giá trị bên trong tài liệu XML đang xem xét

Ý nghĩa :

- Tăng cường tính dễ đọc của tài liệu XML
- Tăng cường tính dễ bảo trì của tài liệu XML

Dạng khai báo và sử dụng :

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

```
<!DOCTYPE Goc [  
    <!ENTITY Ten_1 “Chuoi_1” >  
    <!ENTITY Ten_2 “Chuoi_2” >  
>  
<Goc>  
    <A X=”&Ten_1;”>  
        <B>  
            &Ten_2;  
        </B>  
    </A>
```

```

        <C Y="&Ten_1;"> &Ten_2; </C>
        <D Y> &Ten_2; </C>
    </Goc>

```

Tài liệu XML trên khai báo và sử dụng 2 thực thể

Thực thể thứ 1 : Tên là Ten_1 và được sử dụng trong thuộc tính của 2 thẻ A,C

Thực thể thứ 2 : Tên là ten_2 và được sử dụng trong nội dung của 3 thẻ B,C,D

Việc sử dụng thực thể trong trường hợp này giúp

- Tài liệu dễ đọc hơn : Dùng gián tiếp các tên Ten_1, ten_2 với ngữ nghĩa cụ thể nào đó thay vì trực tiếp các Chuoi_1, Chuoi_2

- Tài liệu dễ bảo trì hơn khi cần thay đổi nội dung của Chuoi_1, Chuoi_2 (chỉ cần thay đổi trong khai báo)

b) Dạng 2

Tham chiếu đến các ký tự đặc biệt được định nghĩa trước

Ý nghĩa : Cho phép sử dụng một số ký tự đặc biệt

Sử dụng ký tự đặc biệt được định nghĩa trước

<	Ký tự <
>	Ký tự >
"	Ký tự nháy kép “
'	Ký tự nháy đơn ‘
&	Ký tự &

c) Dạng 3

Tham chiếu đến một tập hợp các giá trị bên ngoài tài liệu XML

Ý nghĩa :

Cho phép tham chiếu đến tập tin chứa giá trị cần sử dụng nào đó. Các giá trị này không nhất thiết theo định chuẩn XML.

Cách sử dụng này của thực thể thông thường để bổ sung vào nội dung các hình ảnh, âm thanh, v.v.v.

Dạng khai báo thông dụng :

```
<!ENTITY Ten_thuc_the SYSEM Ten_tap_tin >
```

Thực thể Ten_thuc_the tham chiếu đến tập tin có vị trí được cho bởi Ten_tap_tin

Ghi chú :

- Ten_tap_tin bao hàm cả đường dẫn
- Có thể dùng địa chỉ URL như Ten_tap_tin

Ví dụ :

Giả sử đã có tập tin Hinh.jpg lưu trữ hình ảnh một nhân viên trong thư mục hiện hành

```

<!DOCTYPE NHAN_VIEN [
    <!ENTITY Hinh_nhan_vien SYSTEM "Hinh.jpg" >
]>
<NHAN_VIEN Hinh="&Hinh_nhan_vien;" ....>
....
</NHAN_VIEN>

```

d) Dạng 4

Tham chiếu đến một tài liệu XML khác

Ý nghĩa : Cho phép phân rã tài liệu XML thành các tài liệu con được lưu trữ trong các tập tin độc lập.

Dạng khai báo thông dụng : tương tự dạng trên

<!ENTITY Ten_thuc_the **SYSEM** Ten_tap_tin >

Ví dụ :

Giả sử đã có các tập tin Thu_tien_1.xml , Thu_tien_2.xml , Thu_tien_12.xml lưu trữ thông tin về các phiếu thu tiền trong các tháng 1,2,..12 của năm đang xét.

Tập tin Thu_tien.xml lưu trữ thông tin về các phiếu thu trong năm đang xét như sau

```
<!DOC_TYPE THU_TIEN [  
  <!ENTITY Thu_tien_1 SYSTEM “Thu_tien_1.xml” >  
  <!ENTITY Thu_tien_2 SYSTEM “Thu_tien_2.xml” >  
  ...  
  <!ENTITY Thu_tien_12 SYSTEM “Thu_tien_12.xml” >  
>  
<THU_TIEN>  
  &Thu_tien_1;  
  &Thu_tien_2;  
  ...  
  &Thu_tien_12;  
</THU_TIEN>
```

2. Sử dụng tên

Tên thẻ, tên các thuộc tính trong tài liệu XML thuộc về 1 trong 2 loại sau

Loại 1 : Tên không tiền tố

Loại 2 : Tên có tiền tố

Tên không tiền tố

Mô tả đầy đủ các qui tắc đặt tên cho các tên thẻ, thuộc tính là công việc không đơn giản và đặc biệt là cũng không cần thiết.

Một cách tóm tắt (và tất nhiên chưa đầy đủ !) tên là chuỗi bao gồm các ký tự chữ (a-z, A-Z), ký số (0-9) và một số ký tự khác như ‘-’, ‘_’, ‘.’.

Tên có tiền tố

Tên có tiền tố có dạng 2 chuỗi ký tự cách nhau bởi ký tự

‘.’ Chuoi_tien_to : Chuoi_ten

Ví dụ :

<A:MAT_HANG>

<B:MAT_HANG>

Thẻ A:MAT_HANG tương ứng thông tin về mặt hàng trong công ty A. Thẻ B:MAT_HANG tương ứng thông tin về mặt hàng trong công ty B. 2 thẻ này có thể có các thuộc tính khác nhau.

Sử dụng tên có tiền tố :

Nếu chỉ sử dụng tài liệu XML đơn lẻ, riêng cho ứng dụng cục bộ thì không cần thiết dùng tiền tố trong tên. Tuy nhiên nếu cần thiết tiếp nhận, kết xuất toàn bộ/một phần tài liệu XML từ/đến một ứng dụng khác (rất thông dụng trong thương mại điện tử) việc sử dụng tên với tiền tố là rất cần thiết.

Tiền tố của tên sẽ dùng để phân biệt được nguồn gốc của một thẻ trong tài liệu XML được tạo thành từ nhiều tài liệu XML khác có các thẻ trùng phần tên không tiền tố

III. Đặc tả cấu trúc với DTD

Có nhiều dạng khác nhau cho phép khai báo (đặc tả) cấu trúc của tài liệu XML

Dạng 1 : Khai báo cấu trúc tài liệu XML được lưu trữ ngay bên trong chính tài liệu XML đó

```
<!DOCTYPE Ten_the_goc [  
    Đặc tả cấu trúc nội dung các thẻ  
    Đặc tả thuộc tính các thẻ  
>]
```

Dạng 2 : Khai báo cấu trúc tài liệu XML được lưu trữ bên ngoài dưới dạng một tập tin chứa đặc tả cấu trúc nội dung các thẻ, đặc tả thuộc tính các thẻ

```
<!DOCTYPE Ten_the_goc SYSTEM Ten_tap_tin >
```

Ví dụ :

```
<!DOCTYPE DUONG_TRON SYSTEM "DUONG_TRON.dtd" >
```

Dạng 3 : Khai báo cấu trúc tài liệu XML đã được chuẩn hóa, có phạm vi sử dụng rộng rãi. Dạng này thường được dùng với các ngôn ngữ XML chung có phạm vi áp dụng toàn cầu như MathML, VML, XHTML, v.v...

```
<!DOCTYPE Ten_the_goc PUBLIC Chuoi_nhan_dang >
```

1) Đặc tả cấu trúc nội dung các thẻ

Cú pháp chung đặc tả cấu trúc nội dung của một thẻ như sau

```
<!ELEMENT Ten_the Bieu_thuc_dac_ta__cau_truc_noi_dung >
```

Trong đó:

+ Bieu_thuc_dac_ta__cau_truc_noi_dung có thể chỉ là một từ khóa

+ Bieu_thuc_dac_ta__cau_truc_noi_dung cũng có thể bao gồm nhiều từ khóa khác mô tả cách bố trí, sắp xếp các thành phần con bên trong thẻ

Với A, B là 2 thẻ con của thẻ X

A, B A, B sắp xếp theo thứ tự tuần tự A đến B

A* A có thể lặp lại ít nhất 0 lần

B+ B có thể lặp lại ít nhất 1 lần

A? A có thể có hay không có

A|B Có thể chọn sử dụng A hay B

a) Dạng 1

– Từ khóa **ANY** : Thẻ có nội dung bất kỳ theo định chuẩn XML

Ví dụ :

```
<!ELEMENT X ANY >
```

X có thể chứa nội dung bất kỳ. Thông thường cách khai báo này chỉ để mô tả sự tồn tại của X bên trong một thẻ khác

– Từ khóa **EMPTY** : Thẻ không có nội dung

Ví dụ :

```
<!ELEMENT PHAN_SO EMPTY >
```

PHAN_SO không thể có nội dung mà chỉ có thể có các thuộc tính

– Từ khóa **#PCDATA** : Thẻ với nội dung là chuỗi văn bản

Ví dụ :

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

Ho_ten có nội dung là chuỗi và không thể chứa các thẻ khác. Đây là một trong các giới hạn chính của DTD vì không cho phép mô tả chi tiết về “kiểu” hay “loại” của chuỗi văn bản.

Với DTD muốn mô tả chi tiết hơn có thể dùng thẻ ghi chú

Ví dụ :

```
<!ELEMENT He_so (#PCDATA) >
<!-- He_so : A_Float -->
```

b) Dạng 2

Bieu_thuc_dac_ta_cau_truc_noi_dung cũng có thể bao gồm nhiều từ khóa khác mô tả cách bố trí, sắp xếp các thành phần con bên trong thẻ

Với A, B là 2 thẻ con của thẻ X

A, B A, B sắp xếp theo thứ tự tuần tự A đến B

A* A có thể lặp lại ít nhất 0 lần

B+ B có thể lặp lại ít nhất 1 lần

A? A có thể có hay không có

A|B Có thể chọn sử dụng A hay B

*** Tuần tự**

Dạng tuần tự : Các thẻ con chỉ có thể xuất hiện 1 lần duy nhất và phải theo đúng thứ tự xuất hiện trong biểu thức

Cú pháp :

```
<!ELEMENT Ten_the (Ten_the_1, Ten_the_2, ....) >
```

Ý nghĩa :

The_1, The_2, ..., The_k phải xuất hiện một lần duy nhất theo đúng thứ tự trên

Ví dụ :

```
<!ELEMENT DON_THUC (He_so,So_mu) >
```

Thẻ DON_THUC phải bao hàm bên trong 2 thẻ con He_so,So_mu theo đúng thứ tự trên

Ghi chú : Các thẻ bên trong có thể có tên trùng nhau

Ví dụ :

```
<!ELEMENT TAM_GIAC (DIEM,DIEM,DIEM) >
```

Thẻ TAM_GIAC phải bao hàm bên trong đúng 3 thẻ con với tên thẻ là DIEM

- Có thể sử dụng từ khóa #PCDATA trong biểu thức tuần tự (và các loại biểu thức khác)

Ví dụ :

```
<!ELEMENT X (#PCDATA,A,#PCDATA)>
```

Thẻ X phải bao gồm 3 thành phần :

Thành phần thứ 1 là chuỗi văn bản

Thành phần thứ 2 là thẻ có tên A

Thành phần thứ 3 là chuỗi văn bản

*** Tùy chọn**

Dạng tùy chọn : Thẻ con có thể được sử dụng hay không sử dụng

Cú pháp (dạng đơn giản) :

```
<!ELEMENTNT Ten_the (Ten_the_con ?) >
```

Thẻ đang xét có thể chứa 1 hay 0 lần xuất hiện của thẻ có tên là Ten_the_con

Ví dụ :

<!ELEMENT DON_THUC (Ten?) >

Thẻ DON_THUC có thể chứa hay không thẻ Ten

Ghi chú : Có thể kết hợp với biểu thức tuần tự

<!ELEMENT X (A,B?,C) >

Thành phần đầu tiên của thẻ X là thẻ A, kế đến có thể có hay không có thẻ B và thành phần cuối cùng phải là C

-Có thể cho phép tùy chọn một tập hợp các thẻ

<!ELEMENT X (A,B,C)? >

X có thể bao hàm bên trong các thẻ A,B,C (theo thứ tự trên) hay cũng có thể không chứa bất kỳ thẻ nào

Dạng chọn : Bắt buộc chọn một thẻ con để sử dụng trong tập hợp thẻ cho trước

Cú pháp (dạng đơn giản) :

<!ELEMENT Ten_the(Ten_the_1|Ten_the_2|..|Ten_the_k) >

Thẻ đang xét bắt buộc phải chứa duy nhất một trong các thẻ có tên Ten_the_1 hay Ten_the_2, hay ... Ten_the_k

Ghi chú :

- Có thể kết hợp với biểu thức tuần tự

<!ELEMENT X (A,B|C,D) >

Thành phần đầu tiên của thẻ X là thẻ A, kế đến là thẻ B hay thẻ C và thành phần cuối cùng phải là D

- Có thể cho phép chọn một tập hợp các thẻ

<!ELEMENT X ((A,B) | (C,D)) >

X có thể bao hàm bên trong cặp thẻ A,B (theo thứ tự trên) hay cặp thẻ C,D (theo thứ tự trên)

*** Lặp**

Dạng lặp ít nhất 0 lần : Các thẻ con có thể lặp lại nhiều lần hay có thể không có lần nào

Cú pháp :

<!ELEMENT Ten_the (Ten_the_con*) >

Ý nghĩa :

Thẻ đang xét có thể bao hàm bên trong nhiều thẻ có tên là Ten_the_con hay cũng có thể là thẻ rỗng (không có nội dung)

Ví dụ :

<!ELEMENT LOP (HOC_SINH*) >

Thẻ LOP có thể chứa nhiều thẻ HOC_SINH hay không có thẻ HOC_SINH nào

Ghi chú :

- Có thể mô tả lặp đồng thời nhiều thẻ con

<!ELEMENT X (A,B,C)* >

Các thẻ A,B,C theo thứ tự trên có thể lặp lại ít nhất 0 lần trong thẻ X

- Có thể kết hợp với biểu thức tuần tự

Ví dụ :

<!ELEMENT X (A,B*,C) >

Thẻ X có thành phần đầu tiên là thẻ A, kế đến có thể có nhiều hay 0 lần lặp của thẻ B và cuối cùng là thẻ C

- Có thể kết hợp với biểu thức tùy chọn

Ví dụ :

`<!ELEMENT X (A,B*,C?,D) >`

Thẻ X có thành phần đầu tiên là thẻ A, kế đến có thể có nhiều hay 0 lần lặp của thẻ B, kế đến có thể có hay không thẻ C và cuối cùng là thẻ D

- Có thể kết hợp với biểu thức chọn

Ví dụ :

`<!ELEMENT X (A|B,C*,D) >`

Thẻ X có thành phần đầu tiên là thẻ A hay thẻ B , kế đến có thể có nhiều hay 0 lần lặp của thẻ B và cuối cùng là thẻ D

Dạng lặp ít nhất 1 lần : Các thẻ con có thể lặp lại nhiều lần và ít nhất là một lần

Cú pháp :

`<!ELEMENT Ten_the (Ten_the_con+) >`

Ý nghĩa :

Thẻ đang xét có thể bao hàm bên trong ít nhất một thẻ có tên là Ten_the_con

Ví dụ :

`<!ELEMENT DA_THUC (DON_THUC+) >`

Thẻ DATHUC phải bao hàm bên trong ít nhất một thẻ DON_THUC

Ghi chú :

- Có thể mô tả lặp đồng thời nhiều thẻ con

`<!ELEMENT CT_HOA_DON (Mat_hang,So_luong,Don_gia) + >`

Các thẻ CT_HOA_DON phải bao hàm ít nhất 3 thẻ Mat_hang,So_luong,Don_gia

- Có thể kết hợp với biểu thức tuần tự

Ví dụ :

`<!ELEMENT DA_GIAC (DIEM,DIEM,DIEM+) >`

Các thẻ DA_GIAC phải bao hàm ít nhất 3 thẻ DIEM

- Có thể kết hợp với biểu thức tùy chọn

Ví dụ :

`<!ELEMENT BIEU_THUC (Ten?,PHAN_SO+) >`

Thẻ BIEU_THUC có thể chứa hay không thành phần đầu là thẻ Ten và kế đến ít nhất một thẻ PHAN_SO

- Có thể kết hợp với biểu thức chọn

Ví dụ :

`<!ELEMENT X (A|B,C+,D) >`

Thẻ X có thành phần đầu tiên là thẻ A hay thẻ B , đến ít nhất một thẻ B và cuối cùng là thẻ D

2) Đặc tả thuộc tính của thẻ

Cú pháp khai báo chung :

Cú pháp khai báo các thuộc tính của thẻ tương tự như cú pháp khai báo kiểu cấu trúc trong ngôn ngữ lập trình

```
<!ATTLIST Ten_the
    Ten_thuoc_tinh_1 Kieu_1 Tham_so_1
    Ten_thuoc_tinh_2 Kieu_2 Tham_so_2
    ...
    Ten_thuoc_tinh_k Kieu_k Tham_so_k
>
```

Ý nghĩa :

Ten_the : tên thẻ cần khai báo các thuộc tính

Ten_thuoc_tinh_1, Ten_thuoc_tinh_2, ... Ten_thuoc_tinh_k : Tên các thuộc tính của thẻ đang khai báo

Kieu_1, Kieu_2, ..., Kieu_k : Mô tả tập hợp các giá trị mà thuộc tính có thể nhận

Tham_so_1, Tham_so_2, ..., Tham_so_k : Mô tả một số tính chất trên thuộc tính tương ứng

Ví dụ : đặc tả cấu trúc tài liệu XML biểu diễn thông tin về biểu thức phân số $P = 4/5 + 6/7 * 1/3 - 10/3 + 11/2 * 2/3$

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE BIEU_THUC [
<!ELEMENT BIEU_THUC (PHAN_SO | TICH_SO)+ >
<ATTLIST BIEU_THUC
    Ten          CDATA          #IMPLIED
    <!-- Ten : A_String      -->
>
<!ELEMENT PHAN_SO EMPTY >
<ATTLIST PHAN_SO
    Tu_so        CDATA          #REQUIRED
    <!-- Tu_so : A_Int      -->
    Mau_so       CDATA          #REQUIRED
    <!-- Mau_so : A_Int // >0 -->
>
<!ELEMENT TICH_SO (PHAN_SO)+ >
]>
```

a) Kiểu

Kiểu : Mô tả tập hợp các giá trị của thuộc tính

Có nhiều cách khác nhau cho phép mô tả tập hợp các giá trị có thể có của một thuộc tính. Phần sau chỉ giới thiệu 2 cách mô tả chính và thông dụng. Để biết thêm chi tiết về các cách mô tả khác xin tham khảo các tài liệu chuyên biệt về DTD

Cách 1 : Dùng từ khoá CDATA

Cú pháp :

```
<!-- ATTLIST Ten_the
    ...
    Ten_thuoc_tinh      CDATA
    ...
>
```

Ý nghĩa :

Tập hợp các giá trị của thuộc tính với khai báo CDATA chính là tập hợp các chuỗi. Đây là trường hợp sử dụng thông dụng nhất, và đây cũng là một trong các giới hạn của DTD vì không cho phép mô tả chi tiết hơn về kiểu của thuộc tính. Tương tự như nội dung văn bản của thẻ, để mô tả thêm thông tin cần sử dụng các ghi chú

Ví dụ : đặc tả cấu trúc tài liệu XML biểu diễn phương trình đường thẳng trong mặt phẳng

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE DUONG_THANG [
<!ELEMENT DUONG_THANG EMPTY>
    <!--phương trình ax + by+c =0 -->
<!ATTLIST DUONG_THANG
    Ten          CDATA          #IMPLIED
                <!-- Ten : A_String -->
    a            CDATA          #REQUIRED
                <!-- a : A_Float -->
    b            CDATA          #REQUIRED
                <!-- b : A_Float -->
    c            CDATA          #REQUIRED
                <!-- c : A_Float -->
>
    <!--a,b không đồng thời là 0 -->
]>
```

Cách 2 : Dùng biểu thức liệt kê

Cú pháp :

```
<!ATTLIST Ten_the
...
    Ten_thuoc_tinh ( Gia_tri_1,Gia_tri_2,....gia_tri_k)
...
>
```

Ý nghĩa :

Tập hợp các giá trị có thể có của thuộc tính đang xét chính là tập hợp các giá trị được liệt kê Gia_tri_1,Gia_tri_2, ...,Gia_tri_k. Các giá trị này là các chuỗi ký tự

Ví dụ : Đặc tả cấu trúc tài liệu XML biểu diễn thông tin về phiếu điểm của một học sinh

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE PHIEU_DIEM [
<!ELEMENT PHIEU_DIEM (HOC_SINH, DIEM_SO+ ) >
<!ELEMENT HOC_SINH EMPTY >
<ATTLIST HOC_SINH
    Ho_ten          CDATA          #REQUIRED
                <!-- Ho_ten : A_String -->
    Ngay_sinh       CDATA          #REQUIRED
                <!--Ngay_sinh : A_Date -->
    Xep_loai        ("Giỏi", "Khá", "Trung bình", "Yếu") #IMPLIED
>
<!ELEMENT DIEM_SO EMPTY >
<ATTLIST DIEM
    Ten_mon          CDATA          #REQUIRED
                <!-- Ten_mon : A_String -->
    Gia_tri          CDATA          #REQUIRED
                <!-- Gia_tri : A_Float // từ 0 đến 10 -->
>
]>
```

b) Tham số

Tham_số : Mô tả tính chất của thuộc tính

Có nhiều cách khác nhau cho phép mô tả tập hợp các giá trị có thể có của một thuộc tính. Phần sau chỉ giới thiệu 3 cách mô tả chính và thông dụng. Để biết thêm chi tiết về các cách mô tả khác xin tham khảo các tài liệu chuyên biệt về DTD

Cách 1 : Dùng từ khóa **#REQUIRED**

Cú pháp :

```
<!ATTLIST Ten_the
...
Ten_thuoc_tinh Kieu #REQUIRED
...
>
```

Ý nghĩa : Thuộc tính đang xét là thuộc tính bắt buộc phải có. Đây là cách sử dụng phổ biến nhất

Ví dụ : đặc tả cấu trúc tài liệu XML biểu diễn thông tin về các đơn thức với tên bắt buộc phải có

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE DON_THUC [
<!ELEMENT DON_THUC (He_so, So_mu ) >
<ATTLIST DON_THUC
    Ten          CDATA          #REQUIRED
    <!-- Ten : A_String      -->
    Bien_so      CDATA          #REQUIRED
    <!-- Bien_so: A_String   -->
>
<!ELEMENT He_so #PCDATA >
    <!-- He_so : A_Float      -->
<!ELEMENT So_mu #PCDATA >
    <!-- So_mu : A_Int // >=0  -->
]>
```

Cách 2 : Dùng từ khóa **#IMPLIED**

Cú pháp :

```
<!ATTLIST Ten_the
...
Ten_thuoc_tinh Kieu #IMPLIED
...
>
```

Ý nghĩa : Thuộc tính đang xét là tùy chọn và không bắt buộc phải có

Ví dụ : đặc tả cấu trúc tài liệu XML biểu diễn thông tin về tam thức $P(x) = 2x^2 - 4x + 6$

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TAM_THUC [
<!ELEMENT TAM_THUC (DON_THUC,DON_THUC,DON_THUC) >
<ATTLIST TAM_THUC
    Ten          CDATA          #IMPLIED
    <!-- Ten : A_String      -->
    Bien_so      CDATA          "x"
    <!-- Bien_so: A_String // định sẵn là x -->
>
<!ELEMENT DON_THUC EMPTY >
<ATTLIST DON_THUC
    He_so        CDATA          #REQUIRED
    <!-- He_so : A_Float // Khác 0 nếu So_mu=2 --> So_mu
    (0,1,2)      #REQUIRED
    <!-- So_mu : A_Int // =0,1,2 và khác nhau -->
```

>
>

Cách 3 : Dùng từ khóa **#FIXED**

Cú pháp :

```
<!-- ATTLIST Ten_the
...
Ten_thuoc_tinh Kieu #FIXED Gia_tri
...
-->
```

Ý nghĩa : Thuộc tính đang xét phải có giá trị cố định là Gia_tri. Trường hợp này ít được sử dụng

Ví dụ : đặc tả cấu trúc tài liệu XML biểu diễn thông tin về các đơn thức chỉ với biến số x

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE DON_THUC [
<!ELEMENT DON_THUC (He_so, So_mu) >
<ATTLIST DON_THUC
    Ten          CDATA #REQUIRED
    <!-- Ten : A_String    -->
    Bien_so      CDAT  #FIXED "x"
    <!-- Bien_so: A_String  -->
...
<!ELEMENT He_so (#PCDATA) >
    <!-- He_so : A_Float    -->
<!ELEMENT So_mu (#PCDATA) >
    <!-- So_mu : A_Int // >=0    -->
]>
```

Bài tập

1. đặc tả

Yêu cầu chung

đặc tả nội dung & cấu trúc (với DTD) của tài liệu XML tương ứng các đối tượng trong thực tế

Hướng dẫn chung :

- Sử dụng thẻ gốc biểu diễn thông tin của đối tượng trong thực tế đang xét
- Sử dụng các thẻ con của thẻ gốc biểu diễn các "đối tượng con" của đối tượng thực tế đang xét (và tiếp tục nếu "đối tượng con" đang xét lại bao gồm bên trong các "đối tượng con" khác)

*** Dây số nguyên**

Yêu cầu

đặc tả nội dung & cấu trúc (với DTD) của tài liệu XML tương ứng dãy các số nguyên 1, 4, 5, -9, 10

*** đa giác**

Yêu cầu

đặc tả nội dung & cấu trúc (với DTD) của tài liệu XML tương ứng đa giác

ABCDE với

A(0,0) , B(1,6) , C(1,1) , D(7,7) , E(0,2)

*** Danh sách các khối lớp**

Yêu cầu

đặc tả nội dung & cấu trúc (với DTD) của tài liệu XML tương ứng danh sách các khối lớp của trường cấp X. Biết rằng trường X có 3 khối lớp 10,11,12.

Khối 10 có 8 lớp: 10A1, 10A2,10A3, 10A4, 10A5,10A6,10A7,10A8

Khối 11 có 7 lớp : 11A1,11A2,11A4,11A5,11A6,11A7,11A8

Khối 12 có 5 lớp : 12A1, 12A2,12A4, 12A6,12A8

*** Phiếu điểm**

Yêu cầu

đặc tả nội dung & cấu trúc (với DTD) của tài liệu XML tương ứng phiếu điểm của một học sinh

Phiếu điểm

Họ và tên : Giới tính :....

Ngày sinh :....

địa chỉ

Môn học TBHK1 TBHK2 TBNK

....

.....

*** Hóa đơn bán hàng**

Yêu cầu

đặc tả nội dung & cấu trúc (với DTD) của tài liệu XML tương ứng hóa đơn bán hàng

Hóa đơn bán hàng

Khách hàng : Ngày lập :....

Stt Mặt hàng Số lượng Đơn giá Thành tiền

....

.....

Tổng trị giá :

*** Bảng chấm công**

Yêu cầu

đặc tả nội dung & cấu trúc (với DTD) của tài liệu XML tương ứng bảng chấm công tháng của một đơn vị

Bảng chấm công tháng đơn

vị Nhân viên Số ngày công

....

.....

2. Xây dựng ứng dụng

Yêu cầu chung

Thiết kế và lập trình ứng dụng với các yêu cầu chức năng cho trước

Hướng dẫn chung

1. Thiết kế dữ liệu

Sử dụng tập tin Xml biểu diễn thông tin các đối tượng trong thực tế

2. Thiết kế xử lý

*** Tính tiền thuê phòng**

Hệ thống thực tế

Khách sạn X có địa chỉ 123 ABC và điện thoại 333111 có bảng đơn giá thuê phòng như sau

Loại phòng Đơn giá/Ngày

Loại A 250.000

Loại B 220.000

Loại C 180.000

đặc biệt 340.000

Ghi chú :

Nếu khách thuê quá 5 ngày được giảm 10%

Yêu cầu:

Thiết kế và lập trình ứng dụng tính tiền thuê phòng với các yêu cầu chức năng như sau

1. Cập nhật thông tin về khách sạn

2. Bổ sung loại phòng mới

3. Cập nhật thông tin về loại phòng

4. Thanh lý loại phòng

5. Tính tiền thuê phòng

Hướng dẫn thiết kế

1. Thiết kế dữ liệu

Sử dụng tập tin Khach_san.xml với

Thẻ gốc : Biểu diễn khách sạn

Các thẻ con của thẻ gốc : Biểu diễn các loại phòng
2. Thiết kế xử lý

- Dữ liệu: Đặc tả cấu trúc (với DTD)

```
<!DOCTYPE KHACH_SAN [  
  <!ELEMENT KHACH_SAN (LOAI_PHONG+)>  
  <!ATTLIST KHACH_SAN  Ten  CDATA ,  
                        Dien_thoai CDATA ,  
                        Dia_chi CDATA ,  
                        Muc_giam CDATA ,  
                        Ty_le_giam CDATA >  
  <!ELEMENT LOAI_PHONG EMPTY >  
  <!ATTLIST LOAI_PHONG  Ten  CDATA, Don_gia CDATA >  

```

Nội dung :

```
<KHACH_SAN Ten="222"  Dien_thoai="2222"  Dia_chi="33333"  
            Muc_giam="5"  Ty_le_giam="10">  
<LOAI_PHONG Ten="Loại  A"  Don_gia="250000" />  
<LOAI_PHONG Ten="Loại  B"  Don_gia="220000" />  
<LOAI_PHONG Ten="Loại  C"  Don_gia="180000" />  
<LOAI_PHONG Ten="đặc biệt"  Don_gia="380000" />  
</KHACH_SAN>
```

*** đổi ngoại tệ**

Hệ thống thực tế

Cửa hàng đổi tiền X có địa chỉ 124 ABC và điện thoại 333112 có bảng tỷ giá như sau

Ngoại tệ Tỷ giá

....

....

Yêu cầu

Thiết kế và lập trình ứng dụng đổi ngoại tệ với các yêu cầu chức năng như sau

1. Cập nhật thông tin về cửa hàng đổi tiền
2. Bổ sung ngoại tệ mới
3. Cập nhật tỷ giá của ngoại tệ
4. Thanh lý ngoại tệ
5. Tính tiền khi đổi ngoại tệ

Hướng dẫn thiết kế

1. Thiết kế dữ liệu

Sử dụng tập tin Cua_hang.xml với

Thẻ gốc : Biểu diễn thông tin về cửa hàng

Các thẻ con của thẻ gốc : Biểu diễn các ngoại tệ

2. Thiết kế xử lý

*** Trắc nghiệm**

Yêu cầu

Thiết kế và lập trình ứng dụng trắc nghiệm với các yêu cầu sau

1. Biên soạn câu hỏi mới
2. Cập nhật câu hỏi đã soạn
3. Xóa câu hỏi đã soạn
4. Tự rèn luyện với các câu hỏi đã soạn

Ghi chú :

- Câu hỏi trắc nghiệm chỉ bao gồm văn bản, có nhiều chọn lựa khác nhau và chỉ có duy nhất một đáp án đúng
- Với chức năng tự rèn luyện, ứng dụng phát sinh ngẫu nhiên 1 câu hỏi

Hướng dẫn thiết kế

1. Thiết kế dữ liệu

Sử dụng tập tin Trac_nghiem.xml với

Thẻ gốc (DE_BAI) : Biểu diễn thông tin chung của các câu hỏi (nếu cần thiết)

Các thẻ con của thẻ gốc (CAU_HOI) : Biểu diễn các câu hỏi

Các thẻ con của thẻ CAU_HOI : Biểu diễn các chọn lựa

2. Thiết kế xử lý