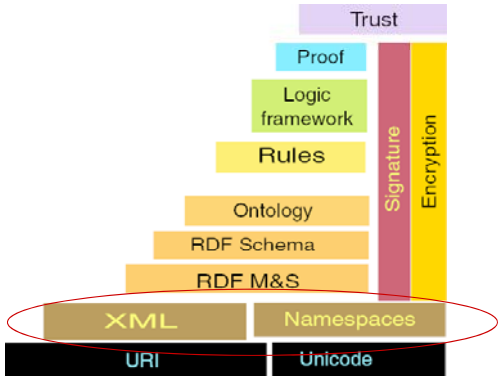
BIỂU DIỄN TẦNG DỮ LIỆU XML

Lê Thanh Hương

Kiến trúc phân tầng của web ngữ nghĩa



XML

- XML (Extensible Markup Language), là tập con của SGML – được sử dụng cho các tài liệu điện tử.
- XML cho phép tạo ra tài liệu có cấu trúc -> máy tính có thể dễ dàng trích thông tin từ tài liệu

So sánh HTML và XML

```
<p><b>Mrs. Mary
    McGoon</b>
<br>
1401 Main Street
<br>
Anytown, NC 34829</p>
```

- HTML được thiết kế như trong ý nghĩ của con người, máy không hiểu được.
- Các thẻ HTML không chỉ cho trình duyệt **thông tin đó là gì**
- XML đưa ý nghĩa vào các thẻ trong văn bản

```
<address>
  <name>
    <title>Mrs.</title>
    <first-name>
      Mary
    </first-name>
    <last-name>
      McGoon
    </last-name>
  </name>
  <street>
    1401 Main Street
  </street>
  <city state="NC">Anytown</city>
  <postal-code>
    34829
  </postal-code>
</address>
```

Các phần của một văn bản XML

- **Thẻ** là phần chữ giữa dấu ngoặc đơn bên trái (<) và ngoặc đơn bên phải (>). Có thẻ bắt đầu (như <name>) và thẻ kết thúc (như </name>)
- **Phần tử** là thẻ bắt đầu, thẻ kết thúc, và mọi thứ giữa chúng. VD, phần tử <name> gồm 3 phần tử con: <title>, <first-name>, và <last-name>.
- **Thuộc tính** là một cặp giá trị tên trong thẻ bắt đầu của một phần tử. VD, state là một thuộc tính của phần tử <city>

5

XML thay đổi Web như thế nào

- **XML đơn giản hóa việc trao đổi dữ liệu.** vì các tổ chức hiếm khi làm chuẩn hóa trên một bộ công cụ duy nhất → có thể dễ dàng chuyển đổi những định dạng dữ liệu bên trong thành XML và ngược lại.
- **XML cho phép mã hóa thông minh.** có thể viết chương trình để xử lí văn bản XML mà không cần con người tác động..
- **XML cho phép tìm kiếm thông minh.** VD, tìm người có tên "Nam"
  - với các trang HTML, sẽ thấy "Việt Nam", hướng Nam, …
  - với văn bản XML, tìm <first-name> chứa từ Chip, kết quả tốt hơn rất nhiều.

6

Các quy tắc văn bản XML

Có ba loại văn bản XML:

- **Văn bản không hợp lệ** không theo nguyên tắc cú pháp được quy định bởi đặc tính kỹ thuật XML hoặc được định nghĩa bởi nhà phát triển
- **Văn bản hợp lệ** tuân theo cả hai nguyên tắc, nguyên tắc cú pháp XML và nguyên tắc quy định trong DTD hoặc lược đồ.
- **Văn bản chuẩn** tuân theo quy tắc cú pháp XML nhưng không có DTD hoặc lược đồ.

7

Các quy tắc văn bản XML

- Phần tử gốc
  - Một văn bản XML phải được chứa trong một phần tử tố đơn gọi là **phần tử gốc**, nó chứa tất cả các từ ngữ và bất cứ phần tử nào trong văn bản. VD:

    ```
    <?xml version="1.0"?> <!-- A well-formed document -->
        <greeting> Hello, World! </greeting>
    ```
- Các phần tử XML không thể đan chéo
- Cần thẻ kết thúc
- Phân biệt chữ hoa/chữ thường
- Thuộc tính phải có giá trị. Các giá trị đặt trong dấu trích dẫn (" hoặc ')

8

Các quy tắc văn bản XML

- Khai báo XML

```
<?xml version="1.0" encoding="ISO-8859-1"
standalone="no"?>
```

- Không gian tên (Namespaces): để sử dụng không gian tên, dùng *tiền tố xmlns* và đặt chúng trong một chuỗi riêng biệt:

```
<?xml version="1.0"?>
<customer_summary
    xmlns:addr="http://www.xyz.com/addresses/"
    xmlns:books="http://www.zyx.com/books/"
    xmlns:mortgage="http://www.yyz.com/title/" >
    ... <addr:name><title>Mrs.</title> ... </addr:name>...
    ... <books:title>Lord of the Rings</books:title> ...
    ... <mortgage:title>NC2948-388-1983</mortgage:title>...
```


Xác định nội dung văn bản

Những yếu tố sử dụng để trình bày dữ liệu

- **Document Type Definition** (Định nghĩa kiểu của Văn bản): DTD xác định các phần tử có thể xuất hiện trong văn bản, thứ tự chúng xuất hiện, cách chúng được sắp xếp trong cái khác, và các chi tiết cơ bản trong cấu trúc văn bản XML.
- **Lược đồ XML.** xác định tất cả các cấu trúc văn bản mà bạn có để đặt trong một DTD, nó cũng có thể xác định kiểu dữ liệu và các quy tắc phức tạp hơn DTD có thể làm.

Xác định nội dung văn bản

- DTD xác định cấu trúc cơ bản của văn bản địa chỉ

```
<!-- address.dtd -->
<!ELEMENT address (name, street, city, state, postal-code)>
<!ELEMENT name (title? first-name, last-name)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT first-name (#PCDATA)>
<!ELEMENT last-name (#PCDATA)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT postal-code (#PCDATA)>
```


Xác định nội dung văn bản

- XML DTDs hạn chế trong việc định nghĩa tài liệu – nó chỉ định nghĩa cấu trúc cú pháp bên trong
- Lược đồ XML (XML schema) có thể mở rộng được, giống như XML
- Lược đồ XML có thể:
  - Sử dụng lại lược đồ trong các lược đồ khác
  - Tạo kiểu dữ liệu mới từ các kiểu chuẩn
  - Tham chiếu nhiều lược đồ từ cùng một tài liệu

Xác định nội dung văn bản

- Xác định thuộc tính

```
<!ELEMENT city (#PCDATA)>
<!ATTLIST city state CDATA #REQUIRED postal-code CDATA #REQUIRED>
```

- Xác định phần tử có trong lược đồ

```
<xsd:element name="address">
   <xsd:complexType> -> xác định một loại dữ liệu mới
      <xsd:sequence>
            <xsd:element ref="name"/>
            <xsd:element ref="street"/>
            <xsd:element ref="city"/>
            <xsd:element ref="state"/>
            <xsd:element ref="postal-code"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

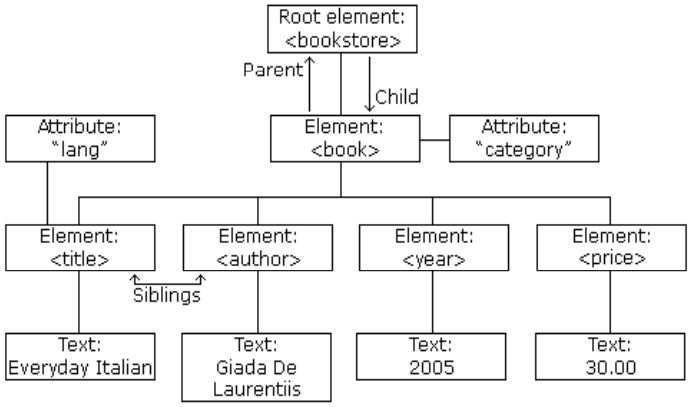

Ví dụ

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="TITLE" type="xs:string"/>
<xs:element name="ARTIST" type="xs:string"/>
<xs:element name="COUNTRY" type="xs:string"/>
<xs:element name="COMPANY" type="xs:string"/>
<xs:element name="PRICE" type="xs:decimal"/>
<xs:element name="YEAR" type="xs:positiveInteger"/>
<!-- definition of complex elements -->
<xs:element name="CD">
     <xs:complexType><xs:sequence>
          <xs:element ref="TITLE"/>
          <xs:element ref="ARTIST" maxOccurs="unbounded"/>
          <xs:element ref="COUNTRY"/>
          <xs:element ref="COMPANY"/>
          <xs:element ref="PRICE"/>
          <xs:element ref="YEAR"/>
     </xs:sequence> </xs:complexType>
</xs:element>
...
</xs:schema>
```


Các giao diện lập trình XML

- DOM (Document Object Model): định nghĩa cách truy cập và thao tác trên tài liệu.
- DOM xây dựng một cây lưu trữ của toàn văn bản. Nếu văn bản quá lớn, nó đòi hỏi một số lượng bộ nhớ rất lớn.

Ví dụ cây lưu trữ



Các kiểu nút

- Tài liệu (Document)
  - Biểu diễn toàn bộ văn bản (nút gốc của cây DOM)
- Phần tử (Element)
- Thuộc tính (Attr)
- Văn bản (Text)
  - Biểu diễn nội dung của 1 thuộc tính hoặc 1 phần tử
- CDATASection
  - Biểu diễn CDATA section trong tài liệu (phần DOM không phân tích)
- EntityReference
  - Biểu diễn tham chiếu thực thể
- Các kiểu khác của DTD

17

Bài tập 1

- Cho 1 tài liệu XML có chứa thông tin về người với họ là Alan, tên là Turing, nghề là computer scientist, mathematician và cryptographer.
- Vẽ cây lưu trữ của tài liệu trên.
- Biến đổi tài liệu để "first" and "last" là các thuộc tính của phần tử person.

Bài tập 2

- Cho 1 tài liệu XML có chứa các thông tin sau: số bảo hiểm xã hội (123456789A), người có họ là Jack, tên là Taylor, địa chỉ gồm postcode (0500), thành phố (Boston), phố (Hamilton street), số điện thoại là 12345 và 6789.
- Vẽ cây lưu trữ của tài liệu trên.
- Đưa ra DTD của tài liệu trên
- Đưa ra lược đồ XML của tài liệu trên