STRUTTURA DI UN DOCUMENTO XML

Corso di Basi di Dati II Mod B
A.A 2009/2010
Prof: F. Cutugno
Slides a cura di: Enza Leano

Documento XML

- Un oggetto XML è detto Documento XML se è ben formato (Well Formed).
- I documenti XML hanno caratteristiche logiche e fisiche.
  - Fisiche: il documento è composto da unità chiamate entities.
  - Logiche: il documento è composto da dichiarazioni, elementi, commenti, caratteri e processing instruction.
- Formalmente:
  - `document ::= prolog element Misc*`

XML Declaration

- I documenti XML dovrebbero iniziare con una dichiarazione XML in cui viene specificata la versione di XML in uso
- Le dichiarazioni devono apparire prima dell'elemento radice
- `prolog::=XMLDecl? Misc* (doctypedecl Misc*)?`
- `XMLDecl::='<?xml' VersionInfo EncodingDecl? SDDecl? S? '?>'`
- `VersionInfo::=S 'version' Eq ("'" VersionNum "'" | '"' VersionNum '"')`
- `Eq::=S? '=' S?`
- `VersionNum::='1.' [0-9]+`
- `Misc::= Comment | PI | S`
  - <?xml version="1.0"?>

Element – definizione formale

- `element::=EmptyElemTag|NotEmptyElement`
- Well-Formed Constraint:
  - Esiste un unico elemento, chiamato radice, che non appare come contesto di nessun altro elemento.

Elementi XML non vuoti

- `NotEmptyElement::= Stag content Etag`
- Un elemento XML è tutto ciò che è compreso tra un tag di apertura (`Stag`) ed il corrispettivo tag di chiusura (`ETag`)
- `Stag ::='<' Name (S Attribute)* S? '>'`
  - Name è una sequenza di caratteri che non inizi per xml
  - S: indica lo spazio
- `ETag ::= '</' Name S? '>'`
- Well-Formed Constraint:
  - il nome dello start-tag e dell'end-tag devono coincidere. N.B.:Xml è case sensitive.
  - Il nome di un attributo deve essere unico all'interno di uno startTag

Attribute

- `Attribute::=Name Eq AttValue`
- `Eq::=S? '=' S?`
- `AttValue::='"' ([^<&"] | Reference)* '"'| "'" ([^<&'] | Reference)* "'"`
- Well Formed Constraint:
  - Nessun riferimento ad entità esterne
  - il simbolo < non può comparire come valore di attributo

Reference

- Una **entity reference** si riferisce al contenuto di un'altra entità
- Per referenziare un'entità si usano l' ampersand (&) e il (;) come delimitatori
- `Reference::=EntityRef | CharRef`
  - `EntityRef::= '&' Name ';'`
  - `CharRef:` insieme di caratteri ammissibili.
- WFC:
  - Entity declared: l'elemento a cui si fa riferimento deve già essere stato dichiarato altrove nel documento

CONTENT

- `Content::=  CharData?`
  `((element | Reference | CDSect | PI | Comment) CharData?)*`
- Tra i due tag si trova il contenuto (content) dell'elemento, che può essere:
  - Simple content: se il contenuto è un semplice testo (CharData)
  - Element content: se il contenuto è costituito da altri elementi (element)
  - Mixed content: se contiene testo intramezzato da altri elementi.

CDSect

- L'elemento **CData** permette di introdurre del testo in modo che questo non venga elaborato dal parser XML, ma venga semplicemente restituito all'utente
- **CDATA sections** possono occorrere ovunque sia consetito inserire un insieme di caratteri
- Le CDATA sections iniziano con la stringa " <![CDATA[" e finiscono con "]]> ":
- CDSect::=CDStart CData CDEnd
- CDStart::='<![CDATA['
- CData ::=(Char* - (Char* ']]>' Char*))
- CDEnd ::=']]>'

PI: Processing instructions

- `PI::='<?' PITarget (S (Char* - (Char* '?>' Char*)))? '?>'`
- `PITarget::= Name - (('X' | 'x') ('M' | 'm') ('L' | 'l'))`
- Sono istruzioni da passare allo strato applicativo:
  - **<?xml-stylesheet type="text/xsl" href="bpg4-0.xsl"?>**
  - IP target = **xml-stylesheet**
  - IP data = **type="text/xsl" href="bpg4-0.xsl“**
- Non processate a livello di parser XML → istruzioni per una applicazione specifica

Empty tag

- Elemento Vuoto: un elemento che non ha contenuto.
- Può essere rappresentato attraverso:
  - Uno start.tag seguito immediatamente da un end tag
  - Un **empty-element tag** :

```
EmptyElemTag ::='<'Name (S attribute)* S? '/>'
```

- WFC:
  - Anche in questo caso il nome di attributo specificato deve essere unico all'interno del tag.

Documento Ben Formato

- Contiene uno o più elementi
- Esiste un unico elemento, chiamato radice, che non appare come content di qualche altro elemento.
- Per tutti gli altri elementi, se lo start-tag è nel content di un altro elemento, allora anche l'end tag lo è → Struttura ad albero
- Valgono tutti I Well-Formed Constraint sui singoli elementi.