APPENDICE C

eXtensible Markup Language

C.1 eXtensible Markup Language, concetti di base

XML è un dialetto di SGML progettato per essere facilmente implementabile ed interoperabile con i suoi due predecessori (tra i quali c'è anche HTML). Per interoperabilità rispetto a SGML, si intende che, essendo un suo sottoinsieme tutti gli strumenti software che trattano documenti del primo tipo, dovrebbero essere in grado di manipolare anche documenti XML. Interoperabilità rispetto ad HTML invece significa che, ad esempio un utente potrebbe indifferentemente con lo stesso browser Web , leggere il contenuto di una pagina HTML, o di un'altra realizzata in XML, oppure durante l'esplorazione di un sito XML, potrebbe avere la possibilità attraverso hyperlink di "saltare" verso pagine HTML (e viceversa). Comunque Il filo conduttore che unisce i tre linguaggi, è il protocollo http, che ne permette la diffusione sul Web.

C.2 Vincoli

Un oggetto è considerato un documento XML se rispetta due vincoli fondamentali:

Well-formedness (ben formatezza): si dice che un documento è ben formato quando è sintatticamente corretto, cioè rispetta alcune le regole imposte dalla grammatica definita nello standard, queste regole sono:

- Tutti gli elementi devono avere rispettivamente un tag di apertura e un tag di chiusura

- I valori degli attributi devono essere racchiusi tra apici
- Elementi possono contenere altri elementi o stringhe, e nient'altro.

Vediamo un esempio. Dato un piccolo documento relativo a informazioni su un libro:

```
<?xml version="1.0"?>
 <book>
    <title>XML</title>
    <author>Richard Smith</author>
 </Books>
```

si può constatare che non è ben formato, perché al tag di apertura `<book>` non corrisponde nessun tag di chiusura `</book>` (al suo posto c'è Books)

Validity (validità): un documento XML è valido quando:

1. è ben formato
2. rispetta le regole imposte dal suo DTD

Questo vincolo deve essere rispettato in tutti quei casi in cui un documento fa riferimento ad un certo DTD. Si prenda l'esempio precedente e si supponga che il documento sia ben formato, e debba rispettare il DTD contenuto nel file "book.dtd":

```
<!ELEMENT book (title,author,publisher)>
<!ELEMENT book (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
```

il documento è il seguente:

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

```
<book>
   <title>XML</title>
   <author>Richard Smith</author>
</book>
```

Nel file XML, non compare l'istanza dell'elemento publisher, di conseguenza non essendo presente la sequenza dei tre elementi: "*title,author,publisher*", il vincolo di validità è violato, questo equivale a dire che il documento XML non è valido per il DTD "book.dtd".

C.3 Documenti XML

XML è caratterizzato da due componenti fondamentali: i dati contenuti all'interno di documenti, e una definizione di tipo opzionale chiamata Document Type Definition (DTD). Il DTD può essere *esterno*: cioè un file con estensione "*.DTD* " localizzato sul sistema stesso o sulla rete (ed identificato tramite un URL); oppure *interno*: contenuto all'interno del documento stesso che in tal caso si dice "*autodescrittivo*". Nel caso in cui sono presenti entrambi, il DTD interno non può ridefinire oggetti già presenti nel DTD esterno, mentre può ridefinire attributi entità e notazioni. In sostanza il parser XML lascia sempre "l'ultima parola" alla definizione di tipo interna.
I componenti fondamentali di un documento XML sono:

1. PROLOG
2. PROCESSING INSTRUCTION
3. DOCUMENT TYPE DECLARATION
4. ELEMENT
5. ATTRIBUTE
6. ENTITY
7. NOTATION
8. CONDITIONAL SECTION

Nei paragrafi seguenti ci occuperemo di descrivere solo i componenti 1.,3. e 4.

C.3.1 Prolog

E' presente all'inizio del documento XML, include:

- la dichiarazione del tipo di documento XML document type declaration
- eventuali commenti (possono apparire in qualunque altra parte del documento)

Un esempio:

```
<?xml version='1.0'?>
<!--declaration-->
<!--PROCESSING INSTRUCTIONS: -->
<?xml-stylesheet href="stile.xsl"  type="text/xsl"?>
<!DOCTYPE DIV SYSTEM 'teisams.dtd' [
  <!ENTITY chapter8.fig1 SYSTEM '08px01.pcx'>
  <!ENTITY chapter8.fig2 SYSTEM '08px02.pcx'>
  <!--PROCESSING INSTRUCTIONS: -->
    <?STYLESHEET href="stile.dsl"  type="text/dsl"?>
    <?PCX compression='standard' type='256 color'>
    <!- - other declarations - ->
 ]>
```

C.3.2 Document Type Declaration

Specifica dove trovare le regole che governano il documento. Si può far riferimento ad un DTD esterno con un modello del tipo:

```
<!DOCTYPE Doc_element  SYSTEM "SUBSET-REFERENCE">
<!DOCTYPE Doc_element  PUBLIC "EXTERNAL-SUBSET-REFERENCE"
                          "OTHER REFERENCE">
```

Dove la parola chiave SYSTEM, indica al processore che il DTD è nell'URL specificato, mentre PUBLIC indica che il DTD è di dominio pubblico ed è individuato dal path `EXTERNAL-SUBSET-REFERENCE`, altrimenti è rintracciato da `OTHER REFERENCE` (che è un URL). Ad esempio volendo indirizzare una definizione di tipo su file system all'indirizzo "`file://DTD_set/books.dtd`" si potrebbe scrivere:

```
<!DOCTYPE  book  SYSTEM "file://DTD_SET/BOOKS.DTD ">
```

Mentre se il DTD si trova sul World Wide Web all'indirizzo "`http://www.w3c.org/XML/DTD/BOOKS.DTD`" allora il doctype è:

```
<!DOCTYPE book SYSTEM " http://www.w3c.org/XML/DTD/BOOKS.DTD" >
```

Quando è presente un DTD interno il doctype assumeme la forma:

```
<!DOCTYPE  Doc_element  [
  INTERNAL-DTD-SUBSET
]>
```

Le due forme dichiarative possono coesistere nel seguente modo:

```
<!DOCTYPE  Doc_Element EXTERNAL-SUBSET-REFERENCE
 [
  INTERNAL-DTD-SUBSET
 ]>
```

In tal caso, se il DTD esterno specifica la struttura del documento, allora in quello interno si potranno dichiarare soltanto entità notazioni e liste di attributi per gli elementi già esistenti (nuove dichiarazioni o dichiarazioni che sovrascrivono quelle già esistenti).

C.3.3 Element

E' il componente fondamentale di un documento XML, e la parte del codice che identifica l'informazione. Nel DTD la dichiarazione di un elemento stabilisce il nome, ne definisce il contenuto ed eventuali attributi associati. La sintassi di una dichiarazione di elemento è la seguente:

```
<!ELEMENT ElementName Content>
```

Un esempio è più esplicativo, prendiamo un DTD di un documento, relativo ad una posta elettronica:

```
<!ELEMENT memo(from,to,cc)>
<!ELEMENT from(#PCDATA)>
<!ELEMENT to(#PCDATA)>
<!ELEMENT cc(#PCDATA)>
```

L'elemento che include tutti gli elementi del documento XML è detto *document-element* (l'elemento memo), senza questo un documento XML non è valido (non è ben formato nel caso che il DTD non ci sia).Un documento valido per il DTD sopraindicato può essere:

```
<?xml version='1.0'?>
<!DOCTYPE refry SYSTEM "../dtd/email.dtd">
<memo>
  <from>smith@flash.com</from>
  <to>webmaster@flash.com</to>
  <cc>spok@net.com</cc>
</memo>
```

Dove il document-element <memo>, è detto *start-tag* mentre </memo> è detto

end-tag, questa nomenclatura vale per tutti gli altri elementi.
Una delle potenzialità della definizione di un elemento è sicuramente la ricorsione:

```
<!ELEMENT node(desc,node*)>
<!ELEMENT desc(#PCDATA)>
```

questa dichiarazione indica che l'elemento *node* contiene un elemento *desc* seguito da zero o più occorrenze di se stesso, grazie a questa si possono realizzare frammenti di codice XML del tipo:

```
<node>
 <desc>top node</desc>
 <node>
  <desc>level 1</desc>
   <node>
     <desc>level 2.1</desc>
   </node>
   <node>
     <desc>level 2.2</desc>
   </node>
 </node>
</node>
```