

# XML: una introduzione

*Nozioni di base e possibili aree di sviluppo  
nel contesto bibliotecario*

di Valdo Pasqui

L'acronimo XML, che sta per eXtensible Markup Language, è diventato uno standard di fatto per la rappresentazione del contenuto informativo nella comunità di Internet. XML ha dato luogo a una serie di specifiche e iniziative che ormai formano un contesto autonomo in continua evoluzione e sta cominciando a diffondersi anche nell'ambito bibliotecario per la rappresentazione di documenti elettronici e metadati e per l'interscambio di dati tra applicazioni. È pertanto utile apprendere le caratteristiche principali e riconoscerne la giusta collocazione nel panorama tecnologico attuale. Questo articolo ha lo scopo di fornire gli elementi base per l'interpretazione dei documenti XML e di proporre una rassegna sintetica di alcune aree di sviluppo connesse a XML nel contesto bibliotecario.

## 1. Che cosa è XML?

XML è un linguaggio semantico che permette di annotare il testo attraverso l'uso di elementi di marcatura comunemente chiamati tag. Lo sviluppo di XML è stato larga-

mente influenzato da SGML (Structured Generalized Markup Language)<sup>1</sup> che permette di definire la struttura dei documenti descrivendone le proprietà formali e le relazioni tra componenti. In pratica XML è una variante semplificata, un sottoinsieme, di SGML.

XML evoca immediatamente anche HTML (Hypertext Markup Language) che è il linguaggio utilizzato per scrivere le pagine web. HTML è un'applicazione di SGML che definisce una specifica tipologia di documenti che fanno uso di un insieme predefinito di marcatori (tag). La differenza più significativa con XML consiste nel fatto che quest'ultimo permette di definire documenti con struttura arbitraria, usando un insieme di elementi di marcatura e di attributi opportunamente definiti attraverso la specifica di una Document Type Definition (DTD), come vedremo più avanti. Inoltre, mentre HTML è principalmente orientato alla presentazione del testo (visualizzazione e formattazione), XML si occupa esclusivamente della struttura del testo e del significato delle entità che lo compongono.

XML è definito da una Recom-

mendation<sup>2</sup> del World Wide Web Consortium (W3C)<sup>3</sup> che descrive come sono fatti i documenti XML e come si devono comportare i programmi che accedono alla loro struttura e contenuto.

## 2. Documenti XML

Logicamente un documento XML ben formato (*well-formed*) è costituito da:

- un prologo che comprende una serie di dichiarazioni;
- un elemento, detto radice, che a sua volta contiene tutti gli elementi secondo una struttura gerarchica ad albero rovesciato in cui gli elementi sono annidati l'uno dentro l'altro; gli elementi possono avere attributi;
- eventuali commenti e istruzioni per l'elaborazione.

Tutti questi componenti sono indicati nel documento attraverso l'uso di marcatori (*markup*) espliciti costituiti dai tag che denotano l'inizio e la fine del contenuto degli elementi, i riferimenti alle entità, gli elementi vuoti, i riferimenti ai caratteri, i commenti, la Document Type Definition, le sezioni CDATA e le istruzioni di elaborazione. In particolare:

- gli elementi sono delimitati da un *tag di inizio*, costituito da un nome racchiuso tra i caratteri '<' e '>', e un *tag di fine*, costituito dallo stesso nome racchiuso tra i caratteri '</' e '>', il tag d'inizio può anche contenere degli attributi;
- gli elementi vuoti, cioè privi di contenuto, consistono di un nome, eventualmente seguito da una lista di attributi, racchiuso tra i caratteri '<' e '/>';
- un nome è tipicamente una sequenza di lettere e cifre in cui possono comparire anche i caratteri '.', '-', '\_', ':', '&#x2D;', '&#x2D;';
- gli attributi sono espressi nella forma nome="valore";
- i commenti sono sequenze di te-

sto delimitate da “<!--” e “-->”;  
 – le informazioni di elaborazione sono delimitate da “<?” e “>?”.  
 Tutto il testo che non è markup costituisce il contenuto del documento, detto “character data”. La figura 1 mostra un esempio di documento XML che rappresenta una semplice bibliografia, il grassetto è usato solo al fine di evidenziare meglio i tag.  
 La riga

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

fa parte del prologo ed è chiamata “dichiarazione XML”: contiene la specifica della versione XML e della codifica dell’insieme di caratteri usati. XML utilizza l’insieme dei caratteri Unicode e i processori di documenti XML devono accettare almeno la codifica UTF-8 e UTF-16.<sup>4</sup> Caratteri che non appaiono

sulla tastiera possono essere inseriti in un documento XML per mezzo di un character reference che può assumere due formati: decimale e esadecimale. Per esempio il carattere ‘©’ (simbolo del copyright) è rappresentato in notazione decimale come &#169; e in notazione esadecimale come &#xA9;. Il prologo continua con la specifica della Document Type Definition che per chiarezza è stata riportata separatamente in figura 2. In alternativa, al posto della sequenza di dichiarazioni che formano la DTD, è possibile usare la seguente forma:

```
<!DOCTYPE bibliografia SYSTEM "http://some.host/bib-xml-dtd.dtd">
```

In questo caso le dichiarazioni della DTD sono contenute esternamente (in un file) e riferite attraverso un URI.<sup>5</sup>

### 3. La Document Type Definition (DTD)

Come abbiamo visto la DTD può essere definita esternamente (ossia contenuta in un file separato) o internamente al documento XML. In ogni caso la sua dichiarazione deve comparire nel prologo mediante il tag <!DOCTYPE Nome .....> e “Nome” deve corrispondere all’elemento radice del documento. Quest’ultimo è un vincolo di validità, ovvero un documento XML ben formato per essere valido deve soddisfare questa condizione.<sup>6</sup> La DTD consiste di un insieme di dichiarazioni che formano la grammatica dei documenti, cioè ne definiscono la struttura, e sono di tre tipi: elementi, attributi ed entità.  
 1) Le Element Type Declaration sono usate per dichiarare i tipi di elementi che possono comparire nel documento secondo il formato:

```
<!ELEMENT Nome specifica_del_contenuto>
```

Nome è il tipo dell’elemento dichiarato e ogni tipo di elemento deve essere dichiarato una sola volta. La specifica\_del\_contenuto definisce e vincola il contenuto dell’elemento indicando quali sotto-elementi, detti figli, possono essere contenuti nell’elemento dichiarato. La specifica del contenuto può essere:  
 – EMPTY, per dichiarare un elemento privo di contenuto;  
 – un misto di caratteri eventualmente mescolati con elementi figli, in questo caso si usa la parola chiave #PCDATA (*parsed character data*) per indicare che il contenuto è una sequenza di caratteri;  
 – un modello del contenuto che consiste di nomi di sotto-elementi organizzati in “sequenze” e “scelte alternative” mediante una semplice grammatica in cui le parentesi tonde ‘(e)’ sono usate per raggruppare le varie porzioni, la virgola ➤

Fig. 1 - Esempio di bibliografia

```
<?xml version="1.0" encoding="UTF-8"?>
<!doctype bibliografia [
.....
]>
<!-- esempio di bibliografia in XML -->
<bibliografia>
  <citazione isbn="0138876479">
    <autore id="ullman">Ullman, Jeffrey D.</autore>
    <autore>Windom, Jennifer</autore>
    <titolo>A first course in database systems</titolo>
    <editore>Prentice Hall</editore>
    <anno>1997</anno>
    <classificazione schema="DDC" valore="5.74"/>
  </citazione>

  <citazione isbn="0201000237">
    <autore>Aho, Alfred V.</autore>
    <autore>Hopcroft, John E.</autore>
    <autore idref="ullman"/>
    <titolo>Data structures and algorithms</titolo>
    <editore>Addison-Wesley</editore>
    <anno>1983</anno>
  </citazione>

  .....
  <nota>Bibliografia redatta da: &estensore;</nota>
</bibliografia>
```

Fig. 2 - DTD per il documento della figura 1

```
<!DOCTYPE bibliografia [
  <ELEMENT bibliografia ((citazione+),nota?)>
  <ELEMENT citazione (autore+, titolo, editore?, anno?, classificazione?)>
  <ATTLIST citazione
    isbn CDATA #IMPLIED>
  <ELEMENT autore (#PCDATA)>
  <ATTLIST autore
    id ID #IMPLIED
    idref IDREF #IMPLIED>
  <ELEMENT titolo (#PCDATA)>
  <ELEMENT editore (#PCDATA)>
  <ELEMENT anno (#PCDATA)>
  <ELEMENT classificazione EMPTY>
  <ATTLIST classificazione
    schema (DDC|LCC|JUDC|MeSH) "DDC"
    valore CDATA #IMPLIED>
  <ELEMENT nota (#PCDATA)>
  <ENTITY estensore "Valdo Pasqui, e-mail: valdo@biblio.unifi.it">
]
```

(;') denota una sequenza, il simbolo ' | ' denota una scelta; una lista di tipo sequenza fissa anche l'ordine in cui gli elementi devono comparire nel contenuto; per ciascuna delle porzioni è possibile dichiarare se può comparire una o più volte ('+'), zero o più volte ('\*'), zero o una volta ('?').

Nell'esempio di figura 2 si definisce che **bibliografia** può contenere uno o più elementi **citazione** ed eventualmente una **nota**. Una citazione a sua volta può contenere nell'ordine uno o più elementi **autore**, un elemento **titolo** (obbligatorio) e poi un **editore**, un **anno** e una **classificazione**, ma questi ultimi tre elementi sono opzionali.

Si possono definire modelli di contenuto più complessi come il seguente:

```
<!ELEMENT sezione ((titolo, (paragrafo+)) | (titolo, (paragrafo*), (sottosezione+)))>
```

in cui si specifica che una sezione può contenere un titolo seguito da almeno un paragrafo, oppure un titolo seguito da zero o più paragrafi e da almeno una sottosezione.

2) Le Attribute List Declaration per-

mettono di definire insiemi di attributi associati a un determinato elemento, stabilendone vincoli e valori per difetto. Gli attributi possono comparire solo nei tag d'inizio degli elementi o negli elementi vuoti e sono formati da coppie nome="valore" (in alternativa il valore può essere delimitato dal simbolo ' invece che da '). L'ordine in cui gli attributi vengono specificati nella dichiarazione e quello in cui compaiono nel tag dell'elemento non sono significativi. Il formato di una dichiarazione di attributi è:

```
<!ATTLIST Nome lista_di_definizioni_attributo>
```

Nome è il tipo di elemento per cui vengono definiti gli attributi. La lista\_di\_definizioni\_attributo è costituita da una o più

Tab. 1 - Principali tipi di attributo

Tipo attributo	Definizione
CDATA	Una qualunque sequenza di testo (character data) che non sia markup, eccetto i caratteri '<' e '&'
enumerazione	Una lista di valori (nmtoken) separati da ' '
ID	Un nome che identifica univocamente un elemento
IDREF	Un riferimento ad un elemento (per mezzo del suo ID)
ENTITY	Il nome di una entità dichiarata
NMTOKEN	Simile a CDATA ma gli spazi e la stringa vuota non sono ammessi

triple, Nome tipo\_attributo valore\_per\_difetto, separate tra loro da uno spazio bianco, in cui:

- Nome è il nome dell'attributo;
- tipo\_attributo ne definisce il tipo (vedi tabella 1);
- valore\_per\_difetto può essere il valore per difetto dell'attributo, delimitato da "'", o una delle parole chiave #REQUIRED (attributo obbligatorio) o #IMPLIED (non viene fornito un valore per difetto).

3) Le Entity Declarations consentono di definire vari tipi di entità. Tra queste le più semplici sono quelle generali, interne di tipo *parsed*. Si tratta di entità la cui occorrenza viene rimpiazzata con il testo definito nella dichiarazione (*parsed*), dichiarate nel documento (interne) e usate solo nel contenuto del documento (generali). In pratica sono delle abbreviazioni usate al posto di porzioni di testo più lunghe. Nel documento XML il riferimento ad una entità avviene con &Nome; dove Nome è il nome assegnato all'entità nella dichiarazione, per esempio **estensore** in figura 1 e 2. XML prevede un insieme di entità predefinite usate per rappresentare i caratteri usati nei tag di marcatura, riportate in tabella 2.

Allo scopo di parametrizzare e semplificare la scrittura della DTD è possibile dichiarare entità, dette *parameter entity*, che hanno effetto solo nella DTD. Supponiamo di voler aggiungere un attributo **nazione** agli elementi **autore** e **editore**, è possibile specificare:

**Tab. 2 - Entità predefinite**

Riferimento all'entità	Carattere
&amp;	&
&lt;	<
&gt;	>
&apos;	'
&quot;	"

```
<ENTITY % AttNazione "nazione CDA-
TA #IMPLIED">
```

```
.....
<!ATTLIST autore
%AttNazione;
.....>
<!ATTLIST editore
%AttNazione;>
```

Notare l'uso del carattere '%' sia nella dichiarazione che nel riferimento. Le entità di tipo parametro possono essere referenziate (ossia usate) solo all'interno della DTD.

Occorre osservare che mentre un documento XML ha una struttura strettamente gerarchica, la DTD che la definisce è invece lineare e non fa risaltare in modo sempre chiaro il concetto di contenimento degli elementi che formano i documenti XML. Inoltre la DTD ha anche alcuni limiti come l'impossibilità di definire due elementi figli diversi, aventi lo stesso nome e contenuti in elementi distinti, e l'impossibilità di definire criteri per la validazione del contenuto degli elementi. Per ovviare a questi problemi è in corso di definizione la specifica XML Schema (vedi anche tabella 3) che prevede l'uso di XML stesso per definire gli schemi. Una definizione di schema è un documento XML composto secondo certe regole il quale a sua volta definisce le regole per articolare e vincolare la struttura di altri documenti XML detti istanze. Lo schema è composto da elementi di tipo semplice e complesso, le loro relazioni, il loro contenuto, gli attributi e i loro valori e le entità. La specifica ha mutuato concetti dall'area dei database e dei linguaggi a oggetti e oramai è in fase avanzata di

assestamento (*candidate recommendation*).

#### 4. Gli spazi di nomi

Come corollario ed estensione di XML sono stati specificati o sono in corso di definizione altri standard, dei quali i più significativi sono riportati nella tabella 3. Tra questi riveste particolare importanza XML Namespaces che fornisce un metodo per qualificare i nomi degli elementi e degli attributi usati nei documenti XML. Il proliferare di specifiche basate su XML rende opportuno poter sfruttare nomi definiti altrove, purché questo avvenga senza determinare conflitti e ambiguità all'interno dei documenti. A tal fine si fa uso degli spazi di nomi e dei nomi qualificati.

Uno spazio di nomi (*namespace*) è una collezione di nomi identificata da un URI che è il nome del namespace e che deve essere univoco e persistente. Un namespace può essere definito in ogni elemento o attributo. Nel caso degli elementi la sua portata si estende a tutti gli elementi figli di quello in cui è definito. L'attributo 'xmlns' è riservato per dichiarare gli spazi di nomi come segue:

```
<bibliografia xmlns:edi="http://ecom-
merce.org/schema/">
```

In questo esempio nell'elemento 'bibliografia' viene dichiarato un namespace il cui nome è 'http://ecommerce.org/schema/' e che è legato al prefisso 'edi'. In base a questa dichiarazione:

```
<edi:price units="Euro">32.18</edi:price>
```

è un elemento il cui tag 'edi:price' è un nome qualificato formato da un prefisso ('edi') associato allo spazio di nomi 'http://ecommerce.org/schema/' e da una parte locale ('price') che è un nome ap-

partenente a quel namespace. È possibile dichiarare anche più namespace usando prefissi diversi:

```
<bibliografia xmlns:edi="http://ecom-
merce.org/schema/"xmlns:dc="http://
purl.org/dc/elements/1.1/">
```

e si può dichiarare un spazio di nomi implicito (per difetto) come segue:

```
<bibliografia xmlns="http://purl.org/
dc/elements/1.1/">
```

in questo caso gli elementi contenuti in 'bibliografia' senza prefisso sono considerati appartenenti al namespace implicito (nell'esempio è l'insieme dei metadati Dublin Core). Un uso più avanzato degli spazi di nomi in XML sarà possibile con il consolidamento della specifica XML Schema. I namespace infatti consentono di verificare che istanze di documenti sono conformi a uno o più schemi e questa modalità, per esempio, offre la possibilità di descrivere le risorse in XML combinando tra loro diversi schemi di metadati (per es. descrittivi, amministrativi, di preservazione).

#### 5. Applicazioni di XML di rilevante interesse per le biblioteche

Di fatto XML sta diventando lo standard per la rappresentazione dei dati ed esistono due categorie principali di applicazione:

- a) markup semantico di documenti e pagine web, unitamente all'uso degli stili (XSL) per la visualizzazione;
- b) formato uniforme per l'interoperabilità tra applicazioni e per la serializzazione di strutture dati complesse trattabili come oggetti.

In prospettiva è facile prevedere un uso pervasivo di XML in Internet, specificamente nelle applicazioni e-commerce, e-busi- ➤

Tab. 3 - Altri standard in ambito XML

<b>XML Schema</b>	Comprende: – XML Schema Part 0: primer – XML Schema Part 1: Structures – XML Schema Part 2: Datatypes che descrivono il linguaggio XML Schema che permette di definire una classe di documenti XML. I documenti XML "istanza" sono conformi a un particolare schema definito.
<b>Extensible Stylesheet Language (XSL)</b>	Si occupa della presentazione del contenuto dei documenti XML, comprende: – XSL Transformations (XSLT) fornisce un insieme di regole per selezionare ed elaborare il contenuto dei documenti – XML Path Language (XPath) un modello le cui espressioni permettono a XSLT di navigare nella struttura ad albero del documento – XSL Formatting Objects (XSL Fos) permette di assegnare proprietà di formattazione al risultato di una trasformazione XSLT ai fini della presentazione grafica o su stampa Per esempio con XSLT è possibile trasformare un documento XML in HTML.
<b>XML Link Language (XLL)</b>	Consiste di due specifiche che permettono la gestione di legami (link) tra documenti XML: – XML Linking Language (XLink) specifica come definire legami tra documenti XML di tipo bidirezionale, con più documenti, con la possibilità che la risorsa designata sia integrata nel corpo di quella di partenza, oppure rimpiazzata o mostrata in una finestra separata. – XML Pointer Language (XPointer) definisce meccanismi per "indirizzare dentro le strutture interne dei documenti XML" facendo uso anche delle espressioni XPath.
<b>XHTML</b>	Si tratta di una riformulazione dello standard HTML 4.0 in modo che i documenti scritti in accordo a questa specifica siano anche documenti XML ben formati. Per esempio tutti gli elementi devono avere l'appropriato tag di fine o essere elementi vuoti.
<b>Document Object Model (DOM)</b>	Fornisce un'interfaccia programmatica (API) per la manipolazione del contenuto dei documenti XML secondo un approccio che trasforma il documento in una collezione di oggetti con struttura gerarchica e fornisce dei metodi per visitare, modificare, inserire e rimuovere ogni parte dell'albero.
<b>Simple API for XML (SAX)</b>	Fornisce una API per l'analisi ( <i>parsing</i> ) dei documenti XML basata sugli eventi. In questo caso si usa un parser XML nel quale si registra il proprio <i>handler</i> , un programma i cui metodi, appositamente scritti, vengono invocati dal parser ogni volta che nella scansione seriale del documento viene trovato un nuovo tag.
<b>Simple Object Access Protocol (SOAP)</b>	Un protocollo basato su XML e pensato per lo scambio di dati (oggetti) in ambito decentralizzato e distribuito, tipicamente con trasporto su HTTP.

ness e in genere per l'organizzazione delle informazioni in rete. Analogamente, anche il settore bibliotecario si avvarrà sempre di più delle tecnologie XML. Una prima area di applicazione è quella della gestione delle collezioni digitali nelle quali XML può essere utilizzato per definire formati interni e uniformi dei documenti, per strutturarne le componenti, per descrivere i metadati e per delimitare le porzioni di contenuto che li alimentano e sui quali vengono applicate le indicizzazioni necessarie per le funzioni di ricerca e browse. Nel contesto della biblioteca digitale XLL consente di organizzare i legami tra i vari oggetti (testi, immagini, sequenze audio e video), mentre XSL permette lo sviluppo di moduli specializzati per la visualizzazione e riproduzione del contenuto. La seconda area di applicazione prevede l'uso di XML come linguaggio (sintassi) di rappresentazione di strutture dati ai fini di assicurare la interoperabilità tra applicazioni di rete distribuite per mezzo di formati di interscambio dei dati e di protocolli applicativi. A conclusione di questa panoramica su XML si propone una sintetica rassegna di alcune rilevanti applicazioni in ambito bibliotecario.

#### *Disseminazione (export/import) di dati bibliografici*

- Conversione dei formati MARC e UNIMARC
- a) Una serie di iniziative e strumenti è censita nel portale XML della Organization for the Advancement of Structured Information Standards (OASIS).
- b) La Lane Medical Library della Stanford University ha sviluppato XMLMARC un convertitore da MARC a XML, scritto in Java, gratuito per usi non commerciali.
- c) Un altro convertitore MARC-XML-MARC è stato sviluppato dalla Logos Research Systems.

d) In Francia il Ministero della cultura e della comunicazione ha sviluppato BiblioML, un convertitore UNIMARC-XML.

e) La National Library of Medicine esporta le citazioni bibliografiche di MEDLINE in formato XML.

f) La Open Archives Initiative(OAI) ha definito OAI\_MARC, una DTD XML per il formato MARC21.

g) BiblioML ha definito una DTD per UNIMARC.

#### *Rappresentazione di schemi di metadati*

– Dublin Core

a) La Open Archives Initiative ha definito una DTD XML per il formato di metadati Dublin Core non qualificato.

b) Nell'ambito della Dublin Core Metadata Initiative (DCMI) sono state formulate due specifiche: Using Dublin Core in XML e XML Encoding of Simple Dublin Core Metadata, anche in questo caso senza l'uso dei qualificatori. La seconda specifica fornisce sia una DTD che un XML Schema.

– Resource Description Framework (RDF)

RDF si basa su un modello che prevede la descrizione delle risorse in termini di proprietà (attributi) e loro valori attraverso la definizione di triple oggetto-attributo-valore dette comandi. I valori delle proprietà possono essere atomici (es. stringhe, numeri) oppure a loro volta possono essere altre risorse dotate delle loro proprietà. RDF utilizza XML per rappresentare in modo testuale i "comandi"; si tratta di una sintassi di serializzazione che consente l'interscambio dei dati RDF.

– Preservation Metadata per oggetti digitali

Il Digital Repository Services (DRS) della Harvard University ha definito una DTD XML per specificare il file di istruzioni che serve a caricare nell'archivio un nuovo oggetto digitale.

– Preservazione e scambio del contenuto e dei metadati

Il progetto Publishing Requirements for Industry Standard Metadata (PRISM) sta sviluppando un insieme di metadati per la descrizione delle risorse, la specifica delle relazioni tra le risorse e la definizione dei diritti di proprietà intellettuale nel contesto delle riviste, giornali, libri e cataloghi. I metadati di PRISM sono espressi sotto forma di documenti XML.

– Il progetto SCHEMAS

Ha lo scopo di creare un forum per tutti i progettisti di schemi di metadati in ambito europeo.

#### *XML come sintassi di trasporto di protocolli applicativi*

– Inter Library Loan (ILL 10160&10161)

a) Il progetto per la realizzazione del Servizio ILL SBN ha specificato e implementato i messaggi (*protocol data unit*) del protocollo ILL in XML mediante la definizione di una specifica DTD.

b) Il progetto europeo ONE2 ha definito uno schema XML per il protocollo ILL.

– Open Archives Initiative (OAI) Protocol for Metadata Harvesting è un protocollo applicativo (trasportato su HTTP) per la cattura (importazione) dei metadati relativi ai documenti messi a disposizione dai "data provider" nell'ambito dell'architettura definita da OAI. I messaggi di risposta del protocollo sono in formato XML e per ogni messaggio è stato definito uno specifico schema XML di validazione.

– La National Information Standards Organization (NISO) sta definendo un protocollo per i servizi di circolazione, denominato Circulation Interchange Protocol (ANSI/NISO-Z39.83-2000x), i cui messaggi sono strutturati e codificati in XML, usando inizialmente una DTD, ma prevedendo il successivo uso di schemi XML.

– È previsto che il futuro Indice SBN, del quale l'ICCU sta avviando il progetto, il protocollo applicativo Indice-NuoviPoli usi XML come sintassi di rappresentazione.

#### *Utilizzo in sistemi di gestione bibliotecaria commerciali*

– La Endeavor Information Systems ha recentemente annunciato ENCompass, un prodotto concepito per la gestione di collezioni nell'ambito della biblioteca digitale. ENCompass combina l'utilizzo di XML e XSL con il RDBMS Oracle e con le funzioni di ricerca e visualizzazione di Voyager per la realizzazione di repository che oltre ai tradizionali cataloghi di biblioteca possono essere collezioni di oggetti digitali (immagini, full-text, sequenze video ecc.). In particolare i dati descrittivi di ciascun repository sono definiti tramite una specifica DTD, generata mediante un'apposita interfaccia grafica. ■

#### **Note**

<sup>1</sup> Vedi ISO 8879:1986(E). *Information processing – Text and Office Systems – Standard Generalized Markup Language (SGML)*, International Organization for standardisation. First edition, 1986, Geneve.

<sup>2</sup> Le W3C Recommendation sono la forma finale delle specifiche emesse nell'ambito del W3C e vanno intese come specifiche cui attenersi per essere conformi allo standard.

<sup>3</sup> T.BRAY – J.PAOLI – C.M.SPERBERG-McQUEEN, *Extensible Markup Language (XML) 1.0 (Second Edition)*, W3C Recommendation 6 October 2000. URL: <<http://www.w3.org/TR/2000/REC-xml-20001006>>.

<sup>4</sup>Unicode prevede una tabella di 1.114.112 caratteri ciascuno dei quali può essere codificato con una o due parole a 16 bit. Universal Character Set Transformation Format (UTF) definisce le modalità di rappresentazione (codifica) dei caratteri Unicode: UTF-8 usa uno o più byte da 8 bit e i primi 256 caratteri sono codificati usan- ➤

do un solo byte. UTF-16 adotta un meccanismo di escape.

<sup>5</sup> Gli Uniform Resource Identifiers (URI) sono sequenze di caratteri che, in accordo con la sintassi specificata in IETF RFC 2396, identificano le risorse per localizzazione o per nome. Gli Uniform Resource Locator (URL), come quello usato nell'esempio, sono un sottoinsieme di URI; T. BERNERS-LEE – R. FIELDING – L. MASINTER, *RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax*, Internet Engineering Task Force (IETF), 1998. URL: <<http://www.ietf.org/rfc/rfc2396.txt>>.

<sup>6</sup> La specifica di XML (vedi nota 3) contiene regole che devono essere rispettate affinché un documento XML sia ben formato (*well-formed constraint*) e altre regole che permettono di stabilire se un documento è anche valido (*validity constraint*).

### Risorse su XML

BiblioML, URL: <<http://www.culture.fr/BiblioML>>.

CIP. NISO, URL: <[http://www.niso.org/NCIP\\_v0\\_1a.dtd](http://www.niso.org/NCIP_v0_1a.dtd)>.

DCMI. Using Dublin Core in XML, URL: <<http://dublincore.org/documents/2000/07/14/dcmes-xml/>>.

DCMI. XML Encoding of Simple Dublin Core Metadata, URL: <<http://dublincore.org/documents/2000/11/dcmes-xml/>>.

DRS, URL: <<http://hul.harvard.edu/ois/systems/drs/>>.

DTD\_MARC, URL: <[http://www.openarchives.org/OAI/oai\\_marc.xs](http://www.openarchives.org/OAI/oai_marc.xs)>.

DTD\_UNIMARC, URL: <<http://www.culture.fr/BiblioML/en/dtds.html>>.

ENCompass, URL: <<http://endinfosys.com/prods/encompass.html>>.

ILL. ICCU-SBN, URL: <<http://prestito.iccu.sbn.it/ILLXm1/ILL-APDU.dtd>>.

ILL. ONE2, URL: <<http://www.portia.dk/pubs/ill/schema/illv2/illv2.xml>>.

MARC-XML-MARC, URL: <<http://www.logos.com/marc>>.

MEDLINE in formato XML, URL: <[http://www.nlm.nih.gov/bsd/xml\\_announce.html](http://www.nlm.nih.gov/bsd/xml_announce.html)>.

OAI\_DC, URL: <<http://www.openarchives.org/OAI/dc.xsd>>.

OAI. Harvesting Protocol., URL: <<http://www.openarchives.org/OAI/openarchivesprotocol.htm>>.

OASIS. Portale su XML, URL: <<http://www.xml.org/>>.

OASIS, *The XML Cover Pages*, URL: <<http://www.oasis-open.org/cover/sgml-xml.html>>.

OASIS. Risorse MARC-XML, URL: <<http://www.oasis-open.org/cover/marc.html>>.

PRISM, URL: <<http://www.prism-standard.org/about/>>.

RDF, URL: <<http://www.w3.org/TR/REC-rdf-syntax/>>.

SCHEMAS, URL: <<http://www.schemas-forum.org/>>.

XMLMARC, URL: <<http://xmlmarc.stanford.edu>>.

W3C. Home Page XML, URL: <<http://www.w3c.org/XML/>>.