

Link in XML

Fabio Vitali

24 marzo 2000



Introduzione

Qui esaminiamo:

- ◆ HyTime
- ◆ XPath e XPointer
- ◆ XLink



HyTime: Introduzione

HyTime è uno standard abbastanza recente (ISO 10744, del 1992) per l'introduzione di caratteristiche ipertestuali e dinamiche nello standard SGML. Esso è cioè un'estensione di SGML.

L'origine di HyTime venne dalla descrizione di spartiti musicali nel progetto SDML (Standard Music Description Language).

HyTime permette di dare una sintassi standard per esprimere locazioni in vari tipi di dati.

Tra i più importanti, locazioni di partenza ed arrivo di salti ipertestuali, spazi di coordinate in oggetti grafici n-dimensionali, frame specifici in dati dipendenti dal tempo (animazioni, filmati, simulazioni scientifiche, etc.).



Descrizione

HyTime permette di creare hyperdocumenti, che possono essere o singoli documenti SGML che usano costrutti HyTime, o complessi insiemi di documenti altamente interconnessi in una grande varietà di media e formati. Spesso una collezione di documenti SGML può essere resa conforme a HyTime semplicemente modificandone il DTD.

Alla base di questo insieme c'è uno hub document, il documento da cui il motore HyTime inizia a processare le informazioni. Lo hub document determina anche lo scope dell'hyperdocumento in questione, ovvero l'insieme di documenti in qualche maniera raggiungibili dallo hub.



Le architectural form

Le architectural form sono una tecnica implementativa e definitoria usata per integrare HyTime ed altre estensioni nel corpo di SGML.

Permette alle applicazioni HyTime di rimanere generali e a non obbligarle ad un singolo DTD (così avviene per HTML, come vedremo).

Architectural form è il nome per una classe di elementi che fanno riferimento ad un unico concetto.

Il motore HyTime controlla la presenza e la natura dell'elemento HyTime osservando l'attributo HyTime nella sua lista di attributi.



Esempio di HyTime

```
<!ELEMENT      mylink      - - (p)*>  
<!ATTLIST     mylink  
  linkend      CDATA      #REQUIRED  
  HyTime       NAME       "clink">
```



Caratteristiche

Lo standard HyTime è diviso in varie clause, ognuna divisa in varie sub-clause, etc. Dopo la parte introduttiva, ogni clause seguente introduce un modulo, ovvero una serie di costrutti che identifica una comune funzionalità.

I moduli HyTime sono 6:

- ◆ Base module,
- ◆ measurement module,
- ◆ location address module,
- ◆ hyperlinks module,
- ◆ scheduling module,
- ◆ rendition module.



I moduli di HyTime (1)

Base module

- ◆ Questo modulo definisce i concetti di base, come le architectural form, i moduli HyTime, e una serie di opzioni e parametri generali.

Measurement module

- ◆ Questo modulo permette di definire ed esprimere unità di misura in qualunque formato, in spazio, tempo o altri domini. Questo viene usato in parte dal modulo ipertestuale.

Location address module

- ◆ Questo modulo usa misure, nomi formali ed altre tecniche per indicare locazioni all'interno di strutture come stringhe, alberi, e dati non-SGML. Fornisce cioè un meccanismo di puntamento utilizzato nel modulo ipertestuale per la creazione di link.



I moduli di HyTime (2)

Hyperlinks module

- ◆ Questo modulo facilita la creazione di link ipertestuali tra locazioni in documenti ed altri tipi di dati.
- ◆ Definisce i clink e gli ilink.
 - ◆ I clink (contextual link) sono link le cui istanze debbono essere presenti all'interno del contesto in cui il link esiste.
 - ◆ Gli ilink (independent link) sono invece link che possono essere posti fuori linea, cioè non risiedere negli estremi del link.



I moduli di HyTime (3)

Scheduling module

- ◆ Questo modulo fornisce il meccanismo per specificare eventi e schedule nello spazio e nel tempo.
- ◆ Vengono usati soprattutto per istanze di sincronizzazione tra elementi diversi in dati temporali (es. video, o musica).

Rendition module

- ◆ Questo modulo permette di descrivere aspetti di presentazione e rendering dei dati.



XPath, XPointer e XLink

XPath, XLink e XPointer sono tre documenti di W3C per la specifica di link ipertestuali sui documenti XML.

Originariamente erano un'unica proposta chiamata XLL (da cui la terna XML, XLL e XSL), poi divisa in tre per semplicità.

- ◆ **XPath** specifica i meccanismi per indicare percorsi all'interno di un documento XML. E' usato anche da XSLT. E' una raccomandazione W3C
- ◆ **XPointer** specifica i meccanismi per riferirsi a parti del documento XML (SGML permette di riferirsi solo ad elementi con l'attributo "ID", HTML solo ad elementi con l'attributo "NAME"). E' un Working Draft (6/12/99)
- ◆ **XLink** usa i meccanismi di indirizzamento di XPointer per descrivere link anche sofisticati tra documenti XML. E' un Working Draft (21/2/2000)



XPointer

Gli XPointer sono indirizzi di locazioni interne a documenti XML. Possono essere usati per indicare link da o a specifiche parti di documenti XML.

Gli XPointer sono una elaborazione dell'identificativo di frammento in un URL:

`http://www.site.com/dir/file.html#nome`

Gli XPointer sono dunque usati in un locatore, tipicamente un URI o URL, per indicare un frammento di quella risorsa.

Gli XPointer sono un'estensione degli XPath, di cui estendono leggermente la sintassi.



XPath

Gli XPath sono una sintassi comune per XSL e XPointer per esprimere locazioni all'interno di documenti XML.

XPath opera sulla struttura logica del documento, non su quella sintattica, usando una sintassi non XML accettabile all'interno di URI e attributi.

Un XPath è un'espressione che restituisce un oggetto di uno di questi quattro tipi:

- ◆ Un booleano
- ◆ Una stringa
- ◆ Un numero
- ◆ Un insieme di nodi (nodi elemento, nodi attributi, nodi testo)



Location Path

Il tipo più importante di XPath è il Location Path. Questo può essere o assoluto o relativo. Un Location Path assoluto inizia con '/'.
WWW

Un Location Path è composto di una sequenza di passi di locazione (Location Steps) separati da '/', e letti da sinistra a destra. Ogni termine individua più precisamente un frammento della risorsa individuata in precedenza.

Es.: `/child::chapter/descendant::para` identifica gli elementi "para" che discenda da un elemento "chapter" che sia figlio diretto della radice del documento XML.



Location Step

Un location step ha tre parti:

- ◆ Un **asse**, che individua la direzione di specifica del location step nell'albero e rispetto al contesto.
- ◆ Un **test**, che individua il tipo e il nome completo del nodo identificato dal location step
- ◆ Zero o più **predicati** che raffinano ulteriormente l'insieme di nodi selezionati dal location step

La sintassi è:

```
axis::test[pred1][pred2]...[pred N]
```



Assi

Gli assi identificano la direzione rispetto alla struttura del documento in cui andare a cercare l'oggetto da restituire rispetto al nodo contesto (NC). Tra gli assi possibili troviamo:

- ◆ **child, descendant**: figlio diretto e a qualunque livello del NC
- ◆ **parent, ancestor**: il genitore immediato e a qualunque livello del NC
- ◆ **self**: il NC;
- ◆ **attribute**: gli attributi del NC
- ◆ **preceding-sibling, following-sibling**: i nodi allo stesso livello ma precedenti o seguenti il NC.
- ◆ **preceding, following**: i nodi a qualunque livello (ma fuori al NC) che precedono o seguono il NC.



Test

Il test di un nodo identifica attraverso il nome o il tipo l'oggetto da restituire.

Se si specifica un nome solo i nodi con quel nome vengono selezionati. Altrimenti si possono utilizzare:

- ◆ `text()`: il nodo testo contenuto nel nodo indicato dall'asse
- ◆ `comment()`: il nodo commento
- ◆ `*`: tutto il contenuto del nodo indicato dall'asse

Esempi:

- ◆ `child::text()`: il testo di un elemento (il nodo testo figlio del nodo contesto)
- ◆ `attribute::*` tutti gli attributi del nodo contesto



Predicati

Un predicato filtra l'insieme dei nodi rispetto alla direzione indicata dall'asse per produrre un nuovo insieme di nodi.

Il filtro può essere attuato sulla posizione, o valutando un'espressione booleana

Ad esempio,

- ◆ `child::para[3]` individua il terzo nodo di nome “para” dentro al NC.
- ◆ `child::para[last()]` individua l'ultimo nodo di nome “para” nel NC.
- ◆ N.B.: `child::para[3]` è equivalente a `child::para[position()=3]`

Altre funzioni:

- ◆ `count()` - il numero di nodi
- ◆ `id()` - l'ID univoco di un nodo
- ◆ `name()` - il nome qualificato del nodo



Sintassi abbreviata

In molti casi esistono delle forme abbreviate usabili invece della sintassi completa:

- ◆ Child::x si può abbreviare con x
- ◆ Attribute::x si può abbreviare con @x
- ◆ Descendant si può abbreviare con '//', self con '.', parent con '..'

Esempi:

- ◆ /doc/chapter[5]/section[2]: la seconda sezione del quinto capitolo del documento.
- ◆ Chapter//para: tutti i para discendenti a qualunque livello del nodo chapter figlio del NC
- ◆ //para: tutti i para discendenti a qualunque livello della radice del documento.



Altri esempi di XPath

- `Para[@type="warning"]`: tutti i para figli del NC che abbiano l'attributo "warning".
- `Para[@type="warning"][5]`: il quinto para figlio di NC ad avere l'attributo type uguale a "warning".
- `Para[5][@type="warning"]`: il quinto para figlio di NC, ma solo se ha l'attributo type uguale a "warning".
- `Chapter[title]`: il "chapter" figlio del NC che contenga uno o più elementi "title"



XPointer

XPointer permettono di specificare la parte fragment di un URL. Essi sono di tre tipi:

- ◆ Un nome (che identifica l'elemento il cui ID è il nome dato)
 - ✦ `http://www.sito.com/file.xml#pippo`
 - ✦ Equivalente a: `...#xpointer(id("pippo"))`
- ◆ Una sequenza di attraversamento
 - ✦ `http://www.sito.com/file.xml#2/5/3`
 - ✦ Equivalente a: `...#xpointer(*[2]/*[5]/*[3])`
- ◆ Uno o più forme generali di frammento (General Fragment part)
 - ✦ `http://www.sito.com/file.xml#xpointer(a/b)`



Estensioni a XPath

XPointer estende il concetto di node in quello di locazione. Una locazione è un nodo, o un punto, o un range di un documento XML. Quindi XPointer definisce due nuovi location types:

- ◆ Point: è definito da un nodo ed un indice, e rappresenta una posizione descritto dall'indice all'interno del nodo. Se il nodo non ha nodi figli, allora l'indice si riferisce alla stringa contenuta nel nodo.
- ◆ Range: è definito come due punti, il primo precedente al secondo.

```
XPointer(id("sec2.1")/descendant::P[last()]) to  
id("sec2.2")/descendant::P[last()]
```



XLink

Gli XLink sono elementi di un documento XML che hanno significato e comportamento di link ipertestuale.

Un elemento XML è identificato come un XLink se possiede degli attributi riservati. Stiamo sfruttando il concetto di architectural form introdotto da HyTime

Questa è una soluzione di compromesso tra il riservare nomi specifici di elementi (che avrebbe limitato la libertà di creazione dei DTD) e il lasciare tutta la gestione dei link ai fogli di stile (che avrebbe negato di attribuire inequivocabilmente agli elementi la natura di link).

Gli XLink sono di due tipi:

- ◆ Link semplici: elementi inline e uni-direzionali
- ◆ Link estesi: inline o out-of-line, spesso multi-direzionali.



Funzionamento di XLink

XLink definisce un namespace e alcuni attributi globali in questo namespace. Includere il namespace equivale ad ammettere XLink nel proprio documento.

Qualunque namespace definito come "http://www.w3.org/1999/xlink" definisce degli XLink

Ad esempio:

```
<b:doc xmlns:a="http://www.w3.org/1999/xlink"
      xmlns:b="http://www.sito.com/mio-namespace">
  <b:para> Clicca
    <b:link a:type="simple" a:href="dest.xml">
      qui!
    </b:link>
  </b:para>
</b:doc>
```



XLink semplici

La presenza dell'attributo `xml:type` con valore "simple" identifica l'elemento come un XLink semplice. Un link semplice ha esattamente due risorse collegate, di cui una è locale ed una è remota. Dunque il link è sempre inline.

```
<A xl:type="simple" xl:href="http://.../"> W3C </A>
```

L'attributo `href` identifica il o i locatori coinvolti. Un locatore è un URL.

Il modo più semplice di definire elementi è specificare in un DTD (o nel subset inline di DTD) la presenza dell'attributo:

```
<!DOCTYPE mydoc [  
  <!ATTLIST A xl:type CDATA #FIXED "simple">  
>  
...  
<A xl:href="http://www.w3.org/"> W3C </A>
```



XLink estesi

Un XLink esteso associa un numero arbitrario di risorse, alcune locali, altre remote. Un link si dice inline se c'è almeno una risorsa locale.

Un link esteso deve contenere elementi tratti dai seguenti:

- ◆ L'elemento locator, che indica l'indirizzo di una risorsa remota)
- ◆ L'elemento arc che specifica regole di attraversamento tra le risorse del link
- ◆ L'elemento resource che indica una risorsa locale (e quindi inline) del documento.

I link estesi sono utili per:

- ◆ Creare link in uscita da documenti che non possono essere modificati (e quindi non permettono link inline)
- ◆ Creare link da o per documenti in formati non XML.
- ◆ Eseguire l'attivazione di collezioni di link a richiesta.
- ◆ Specificare link mutli-direzionali e multi-destinazione.



Link inline e out-of-line

Un link HTML è un link inline: il testo linkante o linkato è il contenuto dell'elemento A, e il link appartiene al documento in cui appare.

I link out-of-line sono link che vengono memorizzati in un documento, e appaiono in un altro. Questo è molto comodo per creare link a o da risorse read-only (CD-ROM) o su cui non si hanno permessi di modifica (documenti altrui).

XLink determina il tipo di link tramite la presenza dell'elemento resource



Un esempio di link esteso

```
<extlink xl:type="extended" xl:role="extlink"
  xl:title="prova">
  <ruolo xl:type="arc" xl:from="uno" xl:to="due"/>
  <ruolo xl:type="arc" xl:from="uno" xl:to="tre"/>
  <local xl:type="resource" xl:role="uno">
    Clicca qui
  </local>
  <url xl:type="locator" xl:role="due"
    xl:href="http://www.sitodue.com/" />
  <url xl:type="locator" xl:role="tre"
    xl:href="http://www.sitotre.com/" />
</extlink>
```



Lo stesso esempio

```
<!DOCTYPE [  
  <!ATTLIST extlink  
    xl:type CDATA #FIXED "extended">  
  <!ATTLIST ruolo  
    xl:type CDATA #FIXED "arc">  
  <!ATTLIST local  
    xl:type CDATA #FIXED "resource">  
  <!ATTLIST url  
    xl:type CDATA #FIXED "locator">  
>  
<extlink xl:role="extlink" xl:title="prova">  
  <ruolo xl:from="uno" xl:to="due"/>  
  <ruolo xl:from="uno" xl:to="tre"/>  
  <local xl:role="uno"> Clicca qui </local>  
  <url xl:role="due" xl:href="http://www.sito2.com/" />  
  <url xl:role="tre" xl:href="http://www.sito3.com/" />  
</extlink>
```



O perfino...

```
<!DOCTYPE [  
  <!ATTLIST extlink x1:type CDATA #FIXED "extended">  
  <!ATTLIST andata x1:type CDATA #FIXED "arc"  
    x1:from CDATA #FIXED "uno"  
    x1:to CDATA #FIXED "due">  
  <!ATTLIST ritorno x1:type CDATA #FIXED "arc"  
    x1:from CDATA #FIXED "due"  
    x1:to CDATA #FIXED "uno">  
  <!ATTLIST primo x1:type CDATA #FIXED "locator"  
    x1:role CDATA #FIXED "uno">  
  <!ATTLIST secondo x1:type CDATA #FIXED "locator"  
    x1:role CDATA #FIXED "due">  
>  
<extlink>  
  <andata/>  
  <primo x1:href="http://www.sito2.com/" />  
  <secondo x1:href="http://www.sito3.com/" />  
</extlink>
```



Linkset esterni

Visto che molti link finiscono con l'essere esterni al documento, come si fa ad associare un documento con l'elenco dei link definiti su di esso?

- ◆ Il metodo migliore è usando un metodo out-of-band: per esempio, un header della trasmissione HTTP
- ◆ Altrimenti, esiste un valore riservato per l'attributo role, external-linkset, per creare un link esteso che punti alla locazione del set di link relativo al documento:

```
<extendedlink
  xmlns:xl="http://www.w3.org/1999/xlink"
  xl:role="xl:external-linkset">
  <loc xl:href="http://..." />
</extendedlink>
```



Altri attributi

Altri importanti attributi sono:

- ◆ **title**: contiene una stringa intelligibile all'utente di spiegazione del senso della risorsa (usabile su simple, locator, resource e arc)
- ◆ **show** (valori possibili: "embed", "replace", "new"): specifica come visualizzare o elaborare la risorsa specificata. "new" indica che la risorsa va visualizzata in un contesto nuovo, come una finestra nuova; "replace" indica che la risorsa nuova sostituisce la vecchia nel contesto esistente. "embed" indica che il contesto della risorsa nuova è la risorsa locale, a cui va sostituita. Per esempio, rimpiazzando il testo del link con il testo della destinazione. (usabile su simple e arc)
- ◆ **actuate** (valori possibili: "onLoad" e "onRequest"): specifica quando l'attivazione del link debba avvenire. "user" indica che deve essere l'utente ad attivare l'azione, per esempio facendo click su un pulsante. "auto" significa che il link deve essere attivato appena la risorsa locale viene caricata. (usabile su simple e arc)



Conclusioni

Qui abbiamo parlato di

- ◆ HyTime
- ◆ XPath e XPointer
- ◆ XLink



Riferimenti

- S. DeRose, E. Maler, D. Orchard, B. Trafford, *XML Linking Language (XLink)*, W3C Working Draft, 21 February 2000, <http://www.w3.org/TR/xlink/>
- J. Clark, S. DeRose, *XML Path Language (XPath)*, Version 1.0, W3C Recommendation 16 November 1999, <http://www.w3.org/TR/xpath>
- S. DeRose, R. Daniel Jr., E. Maler, *XML Pointer Language (XPointer)*, W3C Working Draft 6 December 1999, <http://www.w3.org/TR/xptr>

