

# XBase, XPath e XPointer

---

Fabio Vitali



# Introduzione

Qui esaminiamo:

- ◆ XBase, XPath e XPointer



# XBase, XPath, XPointer e XLink

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

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

- ◆ **XBase** specifica un meccanismo per esprimere URI di base per parti di documenti XML. E' simile all'elemento BASE di HTML.
- ◆ **XPath** specifica i meccanismi per indicare percorsi all'interno di un documento XML. E' usato anche da XSLT. E' una raccomandazione W3C del 16/11/99
- ◆ **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' una candidate recommendation del 7/6/00
- ◆ **XLink** usa i meccanismi di indirizzamento di XPointer per descrivere link anche sofisticati tra documenti XML. E' una Proposed Recommendation del 20/12/00



# 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)



# 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.



# 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 di nodo**, 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 / qualunque livello del NC
- ◆ **self, namespace**: il NC; il nodo namespace del 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.
- ◆ **descendant-or-self, ancestor-or-self**: come descendant e ancestor, ma considerando anche il NC.



# Test di nodo (node test) (1)

Il test di un nodo identifica attraverso il nome o il tipo l'oggetto da restituire. Se un asse contiene un nodo, questo può essere verificato attraverso un test sul nome (se è un elemento) o attraverso altri meccanismi:

- ◆ Se l'asse è `attribute`, il nodo è un attributo
- ◆ Se l'asse è `namespace`, il nodo è il namespace
- ◆ Altrimenti, il nodo è un elemento.

Il test può essere:

- ◆ Un nome: vero se il nodo (elemento o attributo) ha quel nome
- ◆ `text()`, `processing-instruction()`, `comment()`: vero se il nodo è di tipo testo, processing instruction o commento.
- ◆ `node()`: vero sempre
- ◆ `*`: vero per tutti i nodi del tipo definito dall'asse



# 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. In questo secondo caso il risultato è quel sottoinsieme di nodi, tra quelli individuati finora, per cui l'espressione booleana è vera.

L'espressione booleana può essere (ovviamente complicata da operatori booleani come or, and e not), un'espressione relazionale ( $a=b$ ,  $a \neq b$ ,  $a < b$ , ecc.), e/o un Location Path, e/o una funzione pre-definita.

- ◆ **Predicato: [a] con a Espressione Booleana**
- ◆ **Espressione Booleana: a or b, a and b, c, con a e b espressioni booleane e c espressione relazionale**
- ◆ **Espressione Relazionale:  $a = b$ ,  $a \neq b$ ,  $a < b$ ,  $a > b$ ,  $a \leq b$ ,  $a \geq b$ , c, dove c espressione aritmetica**
- ◆ **Espressioni Aritmetiche:  $a + b$ ,  $a - b$ ,  $a * b$ ,  $a \text{ div } b$ ,  $a \text{ mod } b$ , c, -c, con c espressione di disgiunzione**
- ◆ **Espressioni Disgiunzione:  $a | b$ , c, con c espressione di path**
- ◆ **Espressione di Path: LocationPath o Espressione primaria**
- ◆ **Espressione Primaria: numero, lettera o chiamata funzione**



# Funzioni predefinite

XPath non definisce un elenco completo di funzioni, ma un elenco fondamentale.

- ◆ Funzioni sull'insieme di nodi: `last()`, `position()`, `count()`, `id()`, `local-name()`, `namespace-uri()`, `name()`
  - ✦ `child::para[position()=3]` individua il terzo nodo di nome "para" dentro al NC.
  - ✦ `child::para[last()]` individua l'ultimo nodo "para" nel NC.
  - ✦ N.B.: `child::para[3]` è equivalente a `child::para[position()=3]`
- ◆ Funzioni stringa: `string()`, `concat()`, `starts-with()`, `contains()`, `substring-before()`, `substring-after()`, `substring()`, `string-length()`, `normalize-space()`, `translate()`,
- ◆ Funzioni booleane e numeriche



# Sintassi abbreviata

In alcuni 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 (1)

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.



# Estensioni a XPath (2)

Inoltre XPointer definisce alcune funzioni aggiuntive, tra cui:

- ◆ **range-to(nodo)**: dall'inizio di NC al nodo parametro
- ◆ **string-range()**: una sottostringa del NC
- ◆ **range(node-set)**: tutti i range espressi dai parametri
- ◆ **start-point()**, **end-point()**: l'inizio e la fine del NC
- ◆ **here()**, **origin()**: il punto contesto, e l'origine del link.



# Conclusioni

Qui abbiamo parlato di

- ◆ XPath e XPointer



# Riferimenti

- 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>

