

Sommario

- 1 Concetti generali del linguaggio XML
- 2 XPath
- 3 XQuery
- 4 Editor di YAWL: funzionalità avanzate

Sommario

- 1 Concetti generali del linguaggio XML
- 2 XPath
- 3 XQuery
- 4 Editor di YAWL: funzionalità avanzate

Perché XML

XML (eXtensible Markup Language)

- Metalinguaggio per creare documenti arricchiti da informazioni addizionali
- Formato standard e flessibile per lo scambio dei dati
- Favorisce l'interoperabilità:
 - utilissimo per lo scambio di informazioni tra applicazioni diverse

Gli elementi costitutivi nei documenti XML



Regole sintattiche

- Struttura del documento rigorosamente ad albero:

- Deve esistere uno ed un solo elemento radice
- Tutti i tag devono essere chiusi
- I tag devono essere correttamente nestati:

`bold<i>bold and italicitalic</i>` ⇒ illegale

- I valori degli attributi devono sempre essere inclusi tra apici singoli o doppi
- case sensitive:

`<to>Luca</To>` ⇒ scorretto

- gli spazi vengono preservati

Regole sintattiche

Tipi di relazioni:

- padre-figlio (<note>-<from>)
- fratello-fratello (<from>-<message>)
 - Non conta l'ordine con cui sono definiti i fratelli nel documento

Tipi di contenuti:

- Simple content:

```
<to>Luca</to>
```

- Empty content:

```
<exceptions></exceptions>  
oppure <exceptions/>
```

- Element content:

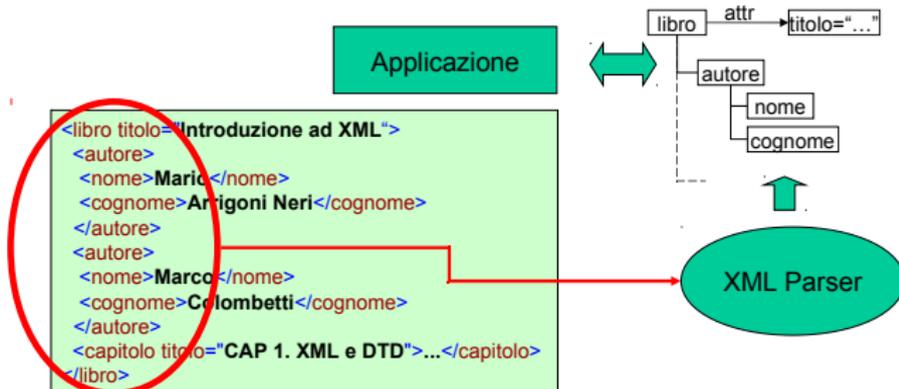
```
<note>  
<from></from>  
</note>
```

```
<note id="12345">  
  <!-- esempio -->  
  <from>Carlo</from>  
  <to>Luca</to>  
  <to>Michele</to>  
  <exceptions></exceptions>  
  <message>  
    Ricorda la riunione  
  </message>  
</note>
```

XML Parser

Parser XML:

- Legge il documento XML e ne interpreta la sintassi
- Estrae tutte le informazioni e le fornisce all'applicazione
- Le applicazioni non devono preoccuparsi di gestire regole particolari
- Può leggere qualsiasi documento XML (indipendentemente dall'applicazione che lo ha creato)



PCDATA

NB: Ci sono caratteri non validi nei PCDATA

```
<comparison> A < B & A > C </comparison>
```

Alcuni caratteri sono riservati ⇒ escaping chars

- `&` ⇒ `&`
- `<` ⇒ `<`
- `>` ⇒ `>`
- `'` ⇒ `'`
- `"` ⇒ `"`

Sommario

- 1 Concetti generali del linguaggio XML
- 2 XPath
- 3 XQuery
- 4 Editor di YAWL: funzionalità avanzate

XPath

XPath è un linguaggio tramite il quale è possibile esprimere delle espressioni per indirizzare parti di un documento XML.

XPath opera su una rappresentazione logica del documento XML, che viene modellato con una struttura ad albero, ed XPath definisce una sintassi per accedere ai nodi di tale albero

Le espressioni definite da XPath per accedere ai nodi dell'albero prendono il nome di Location Path (percorsi di localizzazione).

Regole sintattiche

`/catalogo/Libro`
**seleziona tutti gli
 elementi libro nel
 catalogo**

`/catalogo/Libro/Autore`
**seleziona tutti gli
 elementi Autore di tutti
 i libri nel catalogo**

```
<?xml version="2.0"?>
<catalogo>
  <Libro edizione="Sperling_e_Kupfer">
    <Titolo>Il Delfino</Titolo>
    <Autore>Sergio Bambaren</Autore>
    <prezzo>12.00</prezzo>
  </Libro>
  <Libro>
    <Titolo>La Casa Dipinta</Titolo>
    <Autore>John Grisham</Autore>
    <prezzo>15.00</prezzo>
  </Libro>
  <Libro edizione="Mondadori">
    <Titolo>Svegliatevi dormienti</Titolo>
    <Autore>Philip K.Dick</Autore>
    <prezzo>11.50</prezzo>
  </Libro>
</catalogo>
```


Selezione di rami

Mediante l'operatore | si possono selezionare diversi cammini:

- /catalogo/Libro/Titolo | /catalogo/Libro/Autore

Per specificare gli attributi di un elemento, XPath utilizza il prefisso @:

- //@edizione
Seleziona tutti gli attributi che si chiamano edizione
- //Libro[@edizione]
Seleziona tutti gli elementi Libro che hanno un attributo che si chiama edizione
- //Libro[@*]
Seleziona tutti gli elementi Libro che hanno attributi (qualsiasi sia l'attributo presente)
- //Libro[@edizione='Mondadori']
Seleziona tutti gli elementi Libro che hanno come attributo edizione uguale a Mondadori.

Sommario

- 1 Concetti generali del linguaggio XML
- 2 XPath
- 3 XQuery**
- 4 Editor di YAWL: funzionalità avanzate

Interrogazioni

Il risultato della query sarà:

```
<Autore>Sergio Bambaren</Autore>
```

```
<Autore>John Grisham</Autore>
```

```
<Autore>Philip K. Dick</Autore>
```

Per racchiudere il risultato in una nuova coppia di TAG basta specificare:

```
return <Autori> $x </Autori>
```

questa espressione viene detta **costruttore di elementi**

Interrogazioni

Le espressioni `for` possono essere annidate.

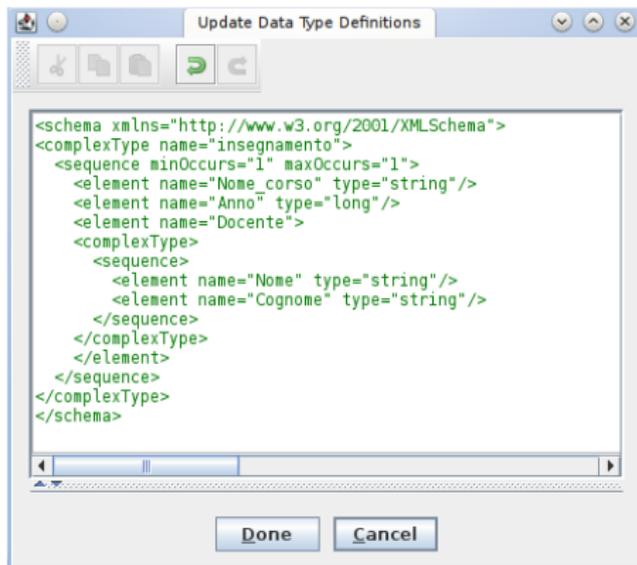
```
for $x
IN document(catalogo.xml)//Libro
  for $y in $x/Autore
  return $y
```


Sommario

- 1 Concetti generali del linguaggio XML
- 2 XPath
- 3 XQuery
- 4 Editor di YAWL: funzionalità avanzate**

Data visibility

L'editor permette di creare tipi di dati complessi attraverso XML Schema.



Data visibility

Le variabili del linguaggio hanno i seguenti tipi di utilizzo :

input only: riceve i dati da una variabile della net;

output only: fornisce i dati a una variabile della net;

input & output: riceve e fornisce i dati da una variabile della net;

local variable: applicabili solo a variabili della net utilizzate per manipolazioni interne alla net (quindi nel suo scope).

Data transfer

Il passaggio di dati può avvenire attraverso:

- variabili (passaggio interno di dati)
- interazione fra processo e ambiente operativo (passaggio esterno)



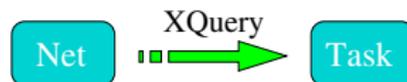
Attenzione!

Il passaggio dati direttamente tra task non è permesso!!!

Data transfer: passaggio interno

Tutti i passaggi interni dei dati avvengono tra il singolo task e la net, utilizzando i parametri input e output del task.

Parametri di input del task



Parametri di output del task



Attenzione!

Non esiste il trasferimento interno dei dati direttamente tra task!

Data transfer: passaggio esterno

I dati richiesti dall'ambiente esterno (variabili input della net) sono gestiti da YAWL con la generazione automatica di form per richiedere informazioni a utenti o servizi associati.

Edit Work Item: 1.1

register

customer:

registrinfo

startDate: 

endDate: 

want_flight:

want_hotel:

want_car:

payAccNumber:

Attenzione!

Non applicabile a variabili locali della net o alle variabili dei task !!!

Data transfer: passaggio esterno

YAWL fornisce alcuni servizi per supportare la modellazione del passaggio dei dati:

- segnalazione automatica della mancanza di variabili necessarie per il trasferimento dati;
- generazione automatica di stringhe XQuery corrette e compatibili con lo YAWL Engine.

Data related: DB conditional routing

Negli split OR e XOR sono presenti condizioni associate ai rami: sono espressioni booleane che, basandosi sui dati del processo, ne determinano il flusso di esecuzione.

Queste condizioni (branching condition) sono condizioni booleane XPath impostate nei dettagli di flusso per split OR e XOR.

Il motore eseguirà il/i ramo/i la cui condizione risulterà vera o quella di default se tutte risultassero false.

Attenzione!

Solo variabili della net sono interessate nella specifica delle condizioni di flusso!!!

