Uso dell'estensione XSLT in archivi eXtraWay

Il tema di cui si occupa questo capitolo affonda le radici nella storia del server eXtraWay ed in particolare nella sua eredità che viene dal server HighWay.
In quest'ultimo il concetto di titolo di un documento era quanto mai corrispondente ad una semplice stringa di testo, più o meno arricchita da separatori, simboli, altre stringhe e così via, usata prevalentemente per una visualizzazione di sintesi del documento stesso nella prima presentazione dell'esito delle selezioni.
Al termine di una ricerca, infatti, a meno che la stessa non identifichi un solo documento, le applicazioni sono solite mostrare in una pagina la visione di sintesi dei primi documenti selezionati consentendo di passare alle pagine successive ed a qualsiasi documento indicato selezionandolo.
In un simile scenario è intuitivo comprendere lo scopo che animava la creazione dei titoli dei documenti.
Col passare del tempo, ancora nel server HighWay, il titolo ha assunto ruoli diversi venendo utilizzato come strumento per ottenere parti del documenti utili a generare altri titoli o in fasi di ordinamento delle selezioni. Il concetto di parti del documento ha presto piede anche presso le applicazioni che hanno provveduto a fare uso dei titoli non più come stringhe da mostrare ma sempre più spesso come un sottoinsieme semplificato del documento dal quale trarre informazioni che non per forza troveranno rappresentazione a video per l'utente finale. Nel titolo, quindi, hanno man mano visto la luce dei separatori il cui scopo era esclusivamente applicativo: riconoscere le singole parti del documento per poterne fare l'uso preferito o isolare una parte del titolo il cui unico fine è quello di essere utilizzato in fase di ordinamento.
Quello che era nato come una stringa è quindi divenuto un oggetto sottoposto ad interpretazione al cui forma s'è fatta man mano sempre più complessa al crescere delle esigenze delle applicazioni: a volte in un titolo hanno finito con l'essere presenti così tante parti di un documento da farlo divenire quasi una sua diversa forma.
Il titolo aveva quindi esaurito il suo scopo originario per divenire una vera e propria sorta di *cache* del documento ad uso e consumo tanto del server stesso[1)] quanto delle applicazioni.

Dal momento che l'attuale natura del titolo è effettivamente quella di una *cache* del documento, variamente utilizzata dal server e dalle applicazioni, è naturale che al crescere delle esigenze delle applicazioni stesse sia cresciuto il bisogno di flessibilità nella generazione di questa *cache* e le regole di generazione dei titoli esistenti, ereditate da HighWay ed adattate alla nuova natura XML dei documenti, sono risultate rapidamente inadeguate.

Quest'osservazione, unitamente all'esigenza sempre più sentita di rappresentare nelle applicazioni i documenti in una forma differente rispetto alla loro reale natura XML, ha condotto a studiare un'estensione XSLT dei documenti, estensione che è il fulcro di questo capitolo.

Come si ottiene la lavorazione

In primo luogo è indispensabile fare uso di un Server eXtraWay di classe 21. Solo da questa versione del server, infatti, è stata implementata questa funzionalità.
La versione 22.1.3.5, d'altro canto, introduce un differente comportamento che verrà meglio dettagliato in seguito.

Gli esempi finali mostreranno più chiaramente come si compie la configurazione ma intanto vediamo quali potenzialità si hanno.
In primo luogo si può scegliere di definire un foglio di stile XSLT per l'intero archivio ovvero uno differente per ciascuna unità informativa[2)].
Il foglio di stile XSLT generico si dichiara aggiungendo un elemento *profile* avente l'attributo *type* valorizzato col valore “cache.xslt”. Il nome del foglio di stile XSLT sarà presente nell'attributo *value*.
Il foglio di stile XSLT specifico di unità informativa si dichiara aggiungendo un attributo *ud_xslt* all'elemento *primary_node*.
In ambo i casi il valore dell'attributo che identifica il foglio di stile (ud_xslt o value) sarà un nome di file assoluto o relativo. Se relativo la directory di origine sarà considerata la stessa directory che ospita l'archivio.

In seguito la dichiarazione degli indici da produrre ricalcherà quella inerente le chiavi di ogni *primary_node* ma il prefisso di chiave cambierà da XML a XSL. Le chiavi prodotte per un *primary_node* hanno come prefisso il nome dell'elemento che identifica il documento quindi saranno `XSL,/<primary_node/…`

Obiettivo Principale

Obiettivo principale di questa estensione funzionale del Server eXtraWay è quella di produrre, a partire da ogni documento, un XML che sia il frutto di una specifica lavorazione.
Data la ricchezza e la potenza del linguaggio XSLT è possibile realizzare dal XML di un documento un nuovo XML che può essere utilizzato per diversi motivi.
Vediamo quali in che modo realizzare ed utilizzare questo nuovo XML descrivendo la teoria ed esponendo alcuni esempi chiarificatori.

Generazione di Indici Speciali

E' pratica comune, in eXtraWay, produrre indici che non afferiscono semplicemente al contenuto dei documenti, espresso nel contenuto degli attributi e degli elementi. Le esigenze applicative, solitamente, richiedono la generazione di indici compositi che possono derivare dalla composizione degli attributi di un elemento oppure dalla composizione di più componenti del documento che vengono semplicemente concatenate. Per ottenere questi risultati è stata realizzata una sintassi che, se pure risponde a gran parte delle esigenze, ha rapidamente mostrato i propri limiti e sarebbe forse irragionevole cercare di estenderla ulteriormente.
La dove la sintassi di concatenazione non si è dimostrata efficace ovvero in quei casi in cui un indice speciale era il frutto di un'elaborazione più complessa e completa di una semplice concatenazione di parti del documento, la risposta all'esigenza di

produrre indici speciali è stata data ricorrendo ad apposite librerie d' archivio, evocate durante le fasi di indicizzazione dei documenti, per mezzo delle quali c'è ampia libertà di creare gli indici più elaborati. Questa pratica, però, richiede un programmatore esperto ed ha tempi di realizzazione e manutenzione piuttosto considerevoli. Si pensi poi ad un'applicazione che possa cambiare nel tempo e variare di conseguenza le esigenze in materia di indici: un intervento in una libreria realizzata appositamente è molto più complesso di una variazione ad un foglio di stile XSLT che può rapidamente cambiare l'XML che da esso deriva ponendo l'archivio, con una semplice rigenerazione di questi XML ed una reindicizzazione, in condizione d'evolvere con uno sforzo del tutto contenuto.

Le chiavi che vengono generate appartengono ad un diverso sottoinsieme del documento. Come è noto le chiavi dei documenti eXtraWay appartengono a rami distinti. Essi sono il ramo UD per le chiavi di servizio (utente che ha compiuto inserimento e modifica, data ed ora di inserimento e modifica, file XML di destinazione e così via) ed il ramo XML (collezione degli indici prodotti grazie alle regole esplicite ed implicite sul contenuto del documento). Nel ramo XML trovano naturalmente collocazione anche le chiavi prodotte da un'eventuale libreria d' archivio. A questi due rami se ne aggiunge un terzo, il ramo XSL. Il comportamento del server in merito a questo rame è del tutto analogo al ramo XML. La dichiarazione delle modalità con le quali si desidera che le chiavi vengano indicizzate è in tutto e per tutto equivalente alle dichiarazioni inerenti il ramo XML e quindi trovano posto nel file *nomearchivio*.conf.xml nella sezione inerente il *primary_node* pertinente. Al pari degli indici per il ramo XML, anche quelli del corrispondente ramo XSL si rifanno alle modalità di default[3)].

Palesemente, al pari degli altri indici, questi possono essere consultati sotto forma di vocabolario, può essere associato loro un *search_alias* per semplificarne l'uso in selezione e possono comunemente essere espressi nelle frasi di ricerca seguendo lo stesso stile tipico dei rami XML.

Attenzione:
Dal momento che tutti gli indici calcolati per una certa *primary_node* hanno in comune il prefisso che li distingue, ciò vale anche per il ramo XSL. Per meglio comprendere supponiamo di avere un *primary_node* identificato dall'elemento `document`. Tutte le chiavi calcolate per il ramo XML saranno identificabili per la loro radice comune: `XML,/document/…` Analogamente a questo, per mantenere invariato il comportamento del server, anche gli indici calcolati sul ramo XSL avranno un prefisso corrispondente: `XSL,/document/…` Perché questo sia possibile il server impone che l'XML prodotto dalla lavorazione XSLT abbia l'elemento radice corrispondente alla *primary_node* del documento dal quale si deriva. Se il foglio di stile XSLT non lo prevede, tale elemento viene posto come radice d'ufficio.

Uso del semilavorato XSLT per gli ordinamenti

L'ordinamento dei documenti può essere effettuato sulle parti dei documenti stessi secondo una sintassi anch'essa ereditata dal progetto HighWay ed adeguata alle esigenze della piattaforma eXtraWay.
Per quanto questa sintassi possa risultare versatile, però, non si può e non si deve escludere che speciali ordinamenti possano essere ottenuti solo grazie ad altrettanto speciali combinazioni di parti di un documento con un pizzico di logica applicativa.
Per questo scopo o per poter produrre una chiave di ordinamento unica applicabile in tutti i casi e magari frutto di molteplici parti provenienti dal documento, l'uso del foglio di stile XSLT può risultare appropriato.

Con la stessa logica che ci conduce a comporre canali speciali per la costituzione di indici altrimenti complicati da realizzare possiamo pensare a chiavi di ordinamento composite e quindi più potenti e veloci. Queste chiavi potrebbero quindi essere escluse dall'indicizzazione.

Uso del foglio di stile per produrre una particolare cache del documento

Come precedentemente accennato l'idea di potenziare il titolo, divenuto *cache* del documento,è quella che ha condotto a questo sviluppo che poi ha trovato particolare applicazione per scopi ben diversi. Ciò non toglie che, in ultima analisi, si possa richiedere al server, tramite questa lavorazione, di produrre una *cache* del documento direttamente per scopi applicativi legati alla sua rappresentazione sia per la forma sintetica che per quella finale.

Osservazioni:
L'uso di questo strumento per la realizzazione di una *cache* del documento per scopi applicativi va necessariamente armonizzata con le esigenze in termini di indici ed ordinamenti. Qualora lo scopo fosse esclusivamente applicativo e l'XML prodotto dalla lavorazione XSLT risultasse di imponenti dimensioni, la progettazione dell'applicazione sarebbe andata sensibilmente al di là dello scopo effettivo di questa funzionalità.

Non dimentichiamo, però, che il ramo XSL è in tutto e per tutto simile al ramo XML quindi nulla ci vieta, nel dichiarare una regola di composizione titolo, di avvalerci di parti provenienti da quel ramo.
Gli esempi finali chiariranno meglio la questione.

Per richiedere, applicativamente, l'intero XML corrispondente alla lavorazione XSL del documento originale si può evocare una speciale funzione del server, non documentata in questa sede, ovvero richiedere semplicemente che il titolo del documento venga fatto sull base della regola…

```
XSL
```

… che esplicita al server di non voler realizzare un titolo per composizione di parti del documento bensì ottenere l'intero XML derivato.

Nota: Questa sintassi non può essere usata come regola di default di composizione del titolo dei documenti espressa nel file *nomearchivio*.conf.xml

L'evoluzione dalla versione 21 alla 22

Come indicato precedentemente, la possibilità di associare ad un'unità informativa un suo elaborato per mezzo di un foglio di stile XSLT è stata introdotta dalla versione 21 del server.
Sin dalla sua introduzione il suo utilizzo è stato particolarmente sporadico, solo poche applicazioni ne hanno realmente fatto uso e nella maggior parte dei casi per comporre sequenze complesse da utilizzare in sede di ordinamento.
In seguito, quando alcuni utilizzatori hanno iniziato a farne un uso più articolato, i limiti iniziali dell'implementazione hanno portato ad evidenziare serie difficoltà nel creare, ad esempio, vocabolari combinati da contenuti della componente XML e della componente XSL. Questi ostacoli hanno portato ad una variazione comportamentale del server dalla versione 22.1.3.5.

Esempi di configurazione ed uso

Prendiamo ad esempio un documento che riporti i dati di una notizia secondo il formato internazionale NewsML.

Nota: Una parte delle cose indicate nel seguente esempio potrebbe essere realizzata con altri accorgimenti, anche a livello di configurazione di archivio, ma serve a mantenere l'esempio semplice e comprensibile.

Nel documento in formato NewsML ci sono alcune parti composite. Le date sono complete di ore, minuti e secondi, il numero progressivo della notizia e la rete in cui essa viene distribuita (nazionale, regionale ed altre) sono invece parti di componenti più complesse.
Nel nostro esempio estrarremo dalla notizia queste componenti, assieme al titolo della stessa, e produrremo un semplice XML che si può prestare per diversi scopi.

Vediamo l'esempio del foglio di stile che chiameremo *small.xslt*

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

<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:output method="xml" indent="yes" />

  <xsl:template match="/">
   <xsl:apply-templates select="//NewsML"/>
  </xsl:template>

  <xsl:template match="text()"/>

  <xsl:template match="*">
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="NewsML">
    <xsl:variable name="TheDate" select="substring(NewsEnvelope/DateAndTime,1,8)"/>
    <xsl:variable name="TheNet" select="substring(NewsEnvelope/NewsProduct/@FormalName,1,3)"/>
    <xsl:variable name="TheId"
select="substring(NewsItem/Identification/NewsIdentifier/NewsItemId,1,4)"/>
     <NewsMLExtension date="{$TheDate}" net="{$TheNet}" number="{$TheId}"
                      composed="{$TheDate}{$ThsId}{$TheNet}">
        <xsl:value-of select="NewsItem/NewsComponent/NewsLines/HeadLine"/>
     </NewsMLExtension>
  </xsl:template>

</xsl:stylesheet>
```

Senza neppure conoscere la struttura del documento NewsML risulta evidente che vengono da esso estratte:

- I primi 8 caratteri di /NewsML/NewsEnvelope/DateAndTime che sarebbe la data della notizia
- I primi 3 caratteri di /NewsML/NewsProduct/ che sarebbe la rete della notizia
- I primi 4 caratteri di /NewsML/NewsItem/Identification/NewsIdentifier/NewsItemId che sarebbe il numero della notizia
- Il contenuto di /NewsML/NewsItem/NewsComponent/NewsLines/HeadLine che sarebbe il titolo della notizia

Le diverse parti vengono combinate in un XML che prevede attributi ed elementi per i valori estratti ed uno specifico attributo che rappresenta la composizione di alcuni di essi.

Vediamo ora come si configurano le chiavi per quanto fatto.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xway_cfg cfg_ver="001.000.001">
   <primary_node ud_name="NewsML" ud_container="listof_NewsML" ud_xslt="small.xslt">
```

```
   ...
      <key name="XSL,/NewsML/NewsMLExtension/@date" value_type="date"/>
      <key name="XSL,/NewsML/NewsMLExtension/@number" value_type="num"/>
      <key name="XSL,/NewsML/NewsMLExtension/@composed" key_style="skip"/>
   ...
   </primary_node>
   <primary_node ud_name="Dossier" ud_container="listof_NewsML">
   ...
   </primary_node>
   ...
   <profile type="cache.xslt" value="global.xslt"/>
   <profile type="title" value='UD,/xw/@UdType "|" XSL,/NewsML/NewsExtension/@date
XSL,/NewsML/NewsExtension'/>
   <profile type="sort.defaultrule" value='XSL(xpart:/NewsML/NewsExtension/@composed)'/>
</xway_cfg>
```

L'esempio precedente mostra un tratto del file *nomearchivio*.conf.xml. In esso si dichiara per il *primary_node* NewsML un particolare foglio di stile chiamato *small.xslt* mentre per il *primary_node* `Dossier` non viene dichiarato alcun foglio di stile specifico quindi vige quello generico dichiarato come global.xslt.
Per la primary_node NewsML sono state dichiarate le caratteristiche di chiave che esulano dal default per i canali necessari ed è stato richiesto di non compiere indici per l'attributo `composed` il cui ruolo risulta evidente nella dichiarazione della regola di ordinamento di default.
Per comporre il reale titolo del documento ci si avvale poi del semilavorato XSL, anche se si potrebbe fare un uso misto che comprenda anche l'XML reale, per motivi di performance e per usare parti già espressamente isolate.

In fine, potremo ricercare tutti i documenti appartenenti ad un certo lasso di tempo ed aventi un termine nel titolo con una sintassi come la seguente:

[XSL,/NewsML/NewsExtension/@date]={20071001	20071031} and [XSL,/NewsML/NewsExtension]=economia

Nota:

C'è una cosa molto importante da sottolineare. Il foglio di stile disegna un XML e ne definisce la radice con l'elemento `NewsExtension`. In tutti gli altri punti, però, sia nel file di configurazione che nella ricerca, questo elemento è a sua volta incapsulato in un ulteriore elemento NewsML che ha lo stesso nome del *primary_node* che abbiamo usato nella nostra esemplificazione.
Questo serve principalmente per distinguere l'esito della lavorazione XSL per ogni diverso tipo di documento, sia nei contenuti quando si fa un titolo o un ordinamento, sia nelle ricerche. In questo modo lo stile adottato per gli XML veri e pripri rimane uniforme.
In sostanza, se il foglio di stile prevede come elemento radice lo stesso del *primary_node* cui verrà applicato il server non compie interventi, se invece non c'è corrispondenza tra i due il server impone che il risultato della lavorazione XSL sia posto all'interno di un elemento dal nome noto.

1)
composizione di alti titoli, ordinamenti
2)
Per quelle per le quali non è stato definito un foglio di stile specifico vale quello generico d'archivio
3)
mono valore alfanumerico per gli attributi e multi valore alfanumerico per gli elementi