

# Un'infrastruttura XML/XSL per la gestione temporale di documenti e dati in ambiente Web

Fabio Grandi    Federica Mandreoli

C.S.I.TE. – C.N.R. e Dipartimento di Elettronica, Informatica e Sistemistica  
Università di Bologna, Viale Risorgimento 2, 40136 Bologna  
Tel: 051 209.3548, Fax: 051 209.3540, Email: {fgrandi,fmandreoli}@deis.unibo.it

**Sommario** In questo lavoro presentiamo una estensione temporale del Web per il supporto e la gestione del tempo di validità, definita attraverso una infrastruttura XML/XSL. Tale estensione consente la definizione esplicita di informazione temporale all'interno di pagine Web (documenti HTML o XML), i cui contenuti possono così essere acceduti e fruiti selettivamente in base alla loro validità. Con la soluzione proposta, agendo su di un contesto di navigazione temporale, è possibile “viaggiare nel tempo” in un ambiente virtuale dato, attraverso un qualunque *browser* che riconosca codice XML. Dal punto di vista dell'utente tale funzionalità consente, ad esempio, di ritagliare percorsi di visita personalizzata, circoscritti ad una particolare epoca, all'interno di un museo virtuale o di una biblioteca storica digitale, oppure di visualizzare l'evoluzione attraverso epoche successive di un sito archeologico, oppure anche di accedere selettivamente a serie storiche di dati (es. quotazioni di borsa), edizioni passate di pubblicazioni *on-line* e quanto altro possa essere organizzato secondo la dimensione temporale. In aggiunta a tali funzionalità di navigazione, l'infrastruttura proposta può anche essere impiegata, in maniera immediata, per la gestione di dati semistrutturati codificati in XML, ponendo le basi per la gestione di dati temporali e lo sviluppo di linguaggi di interrogazione temporale per XML.

Le estensioni del Web proposte sono state sperimentate su di un prototipo software che mostra due esempi applicativi: la realizzazione di un sito Web temporale (museo virtuale) e la gestione di dati XML temporali con funzionalità di query di tipo TSQL2.

## 1 Introduzione

Molto lavoro di ricerca è stato dedicato negli ultimi anni allo studio delle basi di dati temporali [24,10,18]. Frutto di questo sforzo è lo sviluppo di una vasta infrastruttura per la gestione di dati variabili nel tempo e di cui è necessario mantenere tutte le versioni successive, così come sono prodotte dalle modifiche. Questa infrastruttura comprende modelli dei dati, linguaggi di interrogazione, strutture di indicizzazione, ecc. Nella letteratura specifica sono state considerate principalmente due dimensioni temporali [10]: il *tempo di transazione* ed il *tempo di validità*. Il primo riguarda l'evoluzione dei dati rispetto al sistema in cui sono memorizzati mentre il secondo riguarda l'evoluzione dei dati rispetto alla realtà applicativa che essi descrivono. D'altra parte, gli interessi scientifici intorno all'informazione temporale si sono per il momento concentrati su tipi di dati ben strutturati (relazionali od orientati agli oggetti), mentre dati testuali o documenti multimediali meno strutturati non sono stati sinora adeguatamente studiati.

---

Il presente lavoro è stato in parte finanziato dal progetto M.U.R.S.T. “INTERDATA” (Tema 6: metodologie di progettazione di siti WWW) e dal Progetto Finalizzato C.N.R. “Beni Culturali”.

Il World Wide Web (o Web) è invece una enorme collezione distribuita di documenti ipertestuali e multimediali semi-strutturati, redatti allo stato attuale principalmente in conformità allo standard HTML e disponibili *on-line* tramite Internet. Tuttavia HTML è probabilmente destinato ad essere presto soppiantato dall'eXtensible Markup Language (XML [20]) che sta diventando il nuovo standard per la pubblicazione di documenti sul Web. XML è stato progettato per superare le principali limitazioni intrinseche all'HTML ed è stato recentemente raccomandato dal Consorzio W3C [23], essendo tale raccomandazione il passo più importante per la sua standardizzazione. La navigazione di pagine XML può essere al meglio fruita tramite l'impiego di appositi fogli di stile (*stylesheet*). In particolare, l'eXtensible Stylesheet Language (XSL [21]) consente la definizione di regole di trasformazione per documenti XML [26] e la specifica di una semantica di formattazione per la visualizzazione finale dei documenti. In aggiunta a questo, XML può anche essere considerato, in una prospettiva di ricerca sulle basi di dati, un linguaggio di modellazione dei dati [13], adatto a descrivere dati *semistrutturati*, ovvero dati che possono rivelarsi irregolari o incompleti e la cui struttura può cambiare in maniera rapida ed inattesa. Per tale motivo XML è un probabile candidato per divenire il futuro standard dominante per la rappresentazione e lo scambio di dati attraverso Internet [1].

Nonostante l'enorme e continuo sviluppo (sull'onda anche della crescente popolarità) delle tecnologie sottostanti al Web, non è stata finora prestata la dovuta attenzione ai possibili aspetti *temporali* dell'informazione disponibile, ed è questo che ci si è ripromessi di fare come gruppo di ricerca da tre anni a questa parte. I risultati ottenuti consistono nello sviluppo di metodologie per la gestione di risorse Web temporali e nella realizzazione di prototipi, a partire dall'integrazione del tempo di transazione [3] e per finire con l'integrazione del *tempo di validità* che è l'oggetto di questo articolo (di cui [6,5] sono versioni preliminari). L'uso di tale dimensione temporale permette la definizione di *documenti intrinsecamente temporali*, ovvero che contengono al loro interno informazione *storica*, in cui la validità è parte costituente dell'informazione stessa. Per un inquadramento più completo del presente lavoro e delle problematiche dell'integrazione di concetti temporali nelle funzionalità WWW, con una rassegna più estesa dei settori applicativi di interesse e dei lavori correlati, si veda comunque anche [9].

Il contributo del presente lavoro consiste pertanto nell'introduzione del tempo di validità all'interno del Web. Saranno considerate due applicazioni di riferimento che verranno presentate nella Sezione 2. La Sezione 3 è dedicata agli aspetti più tecnici della nostra proposta e descrive un'infrastruttura completa che definisce una possibile soluzione per l'incorporazione di informazione temporale all'interno dei documenti Web e rende disponibili all'utente finale strumenti per la navigazione temporale. La soluzione che proporremo non richiede cambiamenti nella tecnologia Web corrente e si basa invece su XML. L'adozione di un opportuno schema XML per introdurre marche temporali (*timestamp*) all'interno dei documenti e l'uso di un foglio di stile XSL per consentire un trattamento temporalmente selettivo dei documenti metteranno un qualsiasi *browser* XML-compatibile, come Microsoft Internet Explorer 5 (Ie5 [22]), nelle condizioni di gestire documenti temporali. Le estensioni del Web proposte sono anche state sperimentate su di un prototipo software che sarà descritto brevemente nella Sezione 4. L'implementazione sfrutta diffusamente anche un'altra tecnologia Web supportata da Ie5: il Document Object Model (DOM [19]), che è un'*application programming interface* (API) che consente la manipolazione dinamica di documenti HTML e XML (attraverso linguaggi di scripting quale JavaScript). La Sezione 5 contiene alcune

considerazioni finali.

## 2 Due applicazioni di riferimento

La nostra proposta consiste nell'aggiunta della dimensione temporale di validità a documenti Web. Tipiche classi di documenti che qui consideriamo sono i classici documenti ipermediali basati su HTML (pagine Web) contenenti cioè dati fortemente irregolari (perlopiù testi), oppure documenti basati su XML contenenti dati strutturati o semi-strutturati. Nel primo caso l'estensione proposta è finalizzata a consentire la navigazione temporale in ambienti virtuali che sono fonti di informazione storica. Un esempio estremamente adeguato di un tale ambiente è un tipico museo Web, che sarà quindi utilizzato come prima applicazione di riferimento nel resto dell'articolo. Nel secondo caso invece l'estensione è mirata a consentire la gestione di dati temporali che siano rappresentati e negoziati sotto forma di documenti XML. Come esempio di riferimento per questo secondo tipo di applicazione considereremo la rappresentazione di dati relazionali riguardanti impiegati, ossia l'esempio più comunemente usato nella letteratura concernente i *database* temporali.

### 2.1 Il museo Web temporale

L'interesse per i siti museali è in continua crescita: lo dimostrano l'elevato numero di siti museali ormai disponibili sul Web (e la loro analisi come applicazione di metodologie progettuali [4]), e la nascita di una disciplina specifica per l'informatica museale [2], con riviste specialistiche (es. *Archives and Museum Informatics*) e conferenze internazionali (es. *Museums and the Web*).

Il "Web Museum" [12], uno dei primissimi e più popolari siti museali presenti sul Web, conosciuto anche come "il Louvre del Web", rappresenta il primo esempio di riferimento. Abbiamo infatti realizzato una versione temporale di un sottoinsieme delle pagine costituenti il sito originale ed abbiamo sviluppato un ambiente Web per il *browsing* temporale delle sue collezioni. In un museo Web, il *browsing* temporalmente selettivo è finalizzato alla definizione di cammini personalizzati di visita virtuale attraverso i secoli ed attraverso i periodi storico/artistici all'interno delle collezioni museali. L'utente per pianificare una visita può agire sulla selezione del tempo di validità (contesto di navigazione) per modificare il periodo storico di interesse. Ad esempio, possiamo scegliere l'Alto Rinascimento selezionando l'intervallo di validità 1495–1520. Possiamo quindi iniziare la nostra visita entrando in qualche sala o galleria virtuale: troveremo presenti in esse soltanto i dipinti e le sculture temporalmente rilevanti; cambiando il contesto di validità potremo vedere alcune di essi sparire ed altre opere apparire in tempo reale. In una sala dedicata all'Alto Rinascimento Italiano potremo così facilmente prendere visione, per esempio, dell'evoluzione degli stili pittorici di Leonardo da Vinci, Raffaello, Michelangelo e Tiziano, nonché poi studiare, diciamo, tutte le opere contemporanee alla Gioconda.

### 2.2 Dati XML temporali (tabella IMPIEGATI)

Per quanto riguarda la gestione di dati riguardanti impiegati, considereremo le funzionalità di base di un *database* relazionale temporale. A tal fine consideriamo una tabella temporale quale quella riportata in Tab. I, che adotta un *timestamping* a livello di tupla (modello omogeneo) con l'utilizzo di *temporal element* (unione disgiunta di intervalli), in accordo al modello temporale dei dati BCDM

[11]. Ciascuna tupla rappresenta un fatto riguardante un impiegato e l'attributo *Valid Time* ci dice quando il fatto è (era o sarà) vero. L'attributo *ID* è da considerarsi un identificatore tempo-invariante gestito dal sistema (un *surrogato* nel modello dei dati di TSQL2 [17]), che garantisce l'identificazione degli impiegati anche a fronte dei cambiamenti del valore della chiave primaria (come accade nell'esempio per il nome di Pippo). Su questi dati possono essere eseguite *query*

**IMPIEGATI**

<i>ID</i>	<i>Nome</i>	<i>DataNascita</i>	<i>Dipartimento</i>	<i>Stipendio</i>	<i>Valid Time</i>
P1	Pippo	15/8/51	Marketing	2,7M	[ 1996-04-15 – 1998-09-19 ]
P1	Filippo	15/8/51	Contabilità	2,8M	[ 1998-09-20 – 1999-05-31 ] ∪ [ 2000-01-01 – <i>now</i> ]
A1	Anita	25/5/59	Progetto	3,0M	[ 1994-01-01 – 1995-12-31 ] ∪ [ 1997-03-01 – 1999-11-30 ]
A1	Anita	25/5/59	Finanza	3,5M	[ 1996-01-01 – 1997-02-28 ]

**Tabella I.** La tabella temporale **IMPIEGATI**.

temporali per estrarne informazione temporale, ad esempio, selezionando le tuple sulla base della loro validità (selezione temporale). Adotteremo, senza per questo ledere la generalità, TSQL2 come linguaggio di riferimento, essendo esso un linguaggio ampiamente conosciuto derivante da un cospicuo sforzo consensuale e che, quindi, sintetizza la maggior parte delle funzionalità di tutti gli altri linguaggi temporali di interrogazione [17].

### 3 Integrazione del Tempo di Validità nel Web

In questa Sezione presentiamo la nostra proposta di infrastruttura XML/XSL per la definizione ed utilizzo di risorse Web temporali. In generale, l'adozione del tempo di validità è mirata alla gestione di informazione storica (passata, presente o futura) che deve essere esplicitamente codificata all'interno dei documenti, per permetterne quindi la selezione temporale durante il *browsing*. A tale scopo esamineremo separatamente le due principali problematiche sottostanti: la codifica dei documenti e la loro fruizione temporale.

#### 3.1 Definizione di documenti temporali

Un documento temporale è un documento che contiene informazioni storiche con diverse validità. La soluzione da noi prospettata per la definizione di documenti temporali consiste nell'introduzione di un nuovo *tag* XML, denominato `<valid>`, per la definizione di *contesti di validità*. Un contesto di validità viene usato per isolare parti del documento temporale considerate storicamente omogenee ed attribuirgli una pertinenza temporale o *timestamp* che ne rappresenta la validità. Il contenuto racchiuso in un contesto di validità può essere di un qualunque tipo consentito (es. testo, grafica, o un qualunque altro elemento XML compresi altri elementi `valid` annidati). I *timestamp* di un contesto di validità sono codificati per mezzo di *tag* denominati `validity`, che consentono la definizione di un intervallo temporale attraverso la specifica dei suoi estremi (ovvero i valori degli attributi `from` e `to` dell'elemento XML `validity`). In generale è possibile specificare intervalli

multipli aggiungendo altri elementi di tipo `validity`: in tal caso, il *timestamp* è l'unione di tutti gli intervalli specificati all'interno del contesto di validità (un *temporal element* nel modello dei dati BCDM [11]). Ad esempio, il seguente codice XML:

```
<!-- definizione di un contesto di validita' -->
<valid>
  <!-- valid-time timestamping -->
  <validity from="1980-01-01" to="1985-12-31"/>
  <validity from="1995-01-01" to="2000-12-31"/>

```

Questo testo e' `<b>valido dal 1980 al 1985</b>`  
ma anche `<b>valido dal 1995 al 2000</b>...`

```
</valid>
```

definisce un contesto di validità la cui pertinenza temporale risulta essere  $[1980-1985] \cup [1995-2000]$ . Le costanti di tempo sono espresse in conformità al formato ISO 8601, corrispondente al tipo di dati `date` di XML. L'introduzione del contesto di validità, inclusa la definizione dei nuovi *tag* richiesti, è effettuata per mezzo dello schema XML denominato `ValidSchema.xml` visualizzato in Fig. 1.

---

```
<?xml version="1.0" ?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
  xmlns:dt="urn:schemas-microsoft-com:datatypes">

  <AttributeType name="from" required="yes" dt:type="date" />
  <AttributeType name="to" required="yes" dt:type="date" />

  <ElementType name="validity">
    <attribute type='from' />
    <attribute type='to' />
  </ElementType>

  <ElementType name="valid" content="mixed">
    <element type="validity" />
  </ElementType>

</Schema>
```

---

**Figura 1.** Lo schema XML `ValidSchema.xml`.

---

Tale schema deve essere referenziato in ogni documento temporale XML per poter abilitare l'uso dei contesti temporali, nella maniera usuale per gli schemi di tipo XML-Data [25]:

<T-Doc xmlns:dt="x-schema:ValidSchema.xml">

*tutto il resto del documento (con contesti di validità)...*

</T-Doc>

Si noti che l'adozione di uno schema XML assicura l'esecuzione automatica, da parte di un *browser* abilitato XML, di controlli sintattici sulla correttezza (*well-formedness*) dei documenti temporali<sup>1</sup>. L'adozione di uno schema XML piuttosto che di un Document Type Definition (DTD [14]) va ascritta alla sua maggiore flessibilità ed estendibilità. A differenza di un DTD infatti uno schema XML è basato su di un modello di contenuto *aperto* e può pertanto essere applicato ad un qualunque tipo di documento (*irregular data*), anche contenente altri elementi XML definiti altrove o addirittura non definiti. La nostra proposta consente, quindi, di trasformare un qualsiasi sito Web esistente, anche basato sul tradizionale HTML, in un documento temporale semplicemente aggiungendo *timestamp* XML <valid> in conformità con lo schema ValidSchema.xml. In definitiva, come dimostrato dal prototipo software, il nostro schema di marcatura temporale ha (almeno) una duplice funzione:

- può essere impiegato per rendere temporali siti Web “tradizionali” (siti *legacy*, basati su documenti multimediali HTML), al fine di supportare una navigazione temporalmente selettiva sulla base della validità dell'informazione stessa;
- può anche essere usato per rendere temporale l'impiego emergente di XML per la rappresentazione e lo scambio di dati tramite Web, al fine di supportare la gestione di dati temporali strutturati o semistrutturati e abilitare l'utilizzo di funzionalità di trattamento così come sviluppate dalla ricerca sulle basi di dati temporali (es. uso di linguaggi di interrogazione temporale di tipo TSQL2).

Ovviamente anche qualsiasi altro tipo di applicazione innovativa basata su XML potrà essere resa temporale allo stesso modo e quindi beneficiare delle facoltà di rappresentazione di informazione temporale e di gestione temporale dei dati.

Per quanto riguarda il primo esempio di riferimento, le pagine HTML *legacy* che compongono il sito del Web Museum devono essere convertite in documenti XML in conformità allo schema ValidSchema.xml. In questa fase, è richiesto un attento intervento da parte di un progettista umano per l'aggiunta dei *timestamp*, in quanto l'informazione da racchiudere negli ambienti <valid> deve essere attentamente prescelta e devono essere assegnati valori di tempo appropriati ai *timestamp* <validity>.

Per quanto riguarda il secondo esempio di riferimento, i dati riguardanti l'impiegato Pippo (che diventa Filippo) in Tab. I possono essere convertiti nella prescritta rappresentazione XML temporale come spiegato appresso. L'attributo *ID*, che è un identificatore tempo-invariante, è stato trasformato in attributo proprio dell'elemento XML <employee>. È stato poi introdotto un “ambiente” di tipo <valid> per ciascuna delle sue tuple relative a Pippo in Tab. I onde rappresentare una versione temporale omogenea dei dati dell'impiegato stesso. I *timestamp* <validity> corrispondono ai valori dell'attributo *Valid Time* in Tab. I. La rappresentazione che se ne ottiene è la seguente:

---

<sup>1</sup> I controlli sintattici vengono automaticamente effettuati anche sulle costanti di tipo data specificate come valori degli attributi *from* e *to*. Tuttavia Internet Explorer (benché previsto) non supporta attualmente il tipo *date* e pertanto non effettua tali controlli.

```

<impiegato ID="P1">
  <valid>
    <Nome>Pippo</Nome> <DataNascita>15/8/51</DataNascita>
    <Dipartimento>Marketing</Dipartimento> <Stipendio>2,7M</Stipendio>
    <validity from="1996-04-15" to="1998-09-19" />
  </valid>
  <valid>
    <Nome>Filippo</Nome> <DataNascita>15/8/51</DataNascita>
    <Dipartimento>Contabilita'</Dipartimento> <Stipendio>2,8M</Stipendio>
    <validity from="1998-09-20" to="1999-05-31" />
    <validity from="2000-01-01" to="9999-99-99" />
  </valid>
</impiegato>

```

### 3.2 Navigazione e browsing temporale

L'uso dei documenti temporali mira a consentire il *browsing* selettivo dei documenti, ovvero la navigazione temporale del Web e l'interrogazione dei dati temporali. Supponendo che il tempo di validità usato per *default* nella navigazione sia l'intero asse del tempo, che consente la visualizzazione degli interi documenti (corrispondendo così all'uso di base dei documenti non temporali), per una navigazione temporalmente selettiva il tempo di validità adottato deve essere adeguatamente delimitato dall'utente (es. tramite qualche strumento messo a disposizione dal *browser*). Non appare troppo restrittivo supporre che esso sia un *intervallo* temporale, dato che anche la maggior parte delle *query* eseguite in una base di dati di tipo *valid-time* si basano proprio sulla sovrapposizione con un intervallo. Una volta fissato dall'utente, tale intervallo di validità è conosciuto dal *browser* come *contesto di validità di navigazione*. Quando i documenti temporali sono elaborati, solamente quelle parti i cui *timestamp* si sovrappongono (con intersezione non nulla) al contesto di navigazione sono effettivamente presi in considerazione e visualizzati. Per gli altri si va soltanto alla ricerca di un eventuale ulteriore contesto temporale annidato, che potrebbe risultare da visualizzare in quanto pertinente al contesto di navigazione<sup>2</sup>. Lo stesso procedimento viene automaticamente applicato quando nuovi documenti sono reperiti seguendo un collegamento ipertestuale, rendendo così possibile una navigazione temporale vera e propria. Nel caso dei dati XML temporali (ad esempio la tabella temporale *Employee*), il contesto di navigazione assume il significato di *contesto di validità di interrogazione* e può essere usato da una applicazione basata sul Web per la selezione di dati sulla base di una condizione di confronto (non soltanto di *overlap*).

La selezione temporale nella nostra proposta si basa sull'adozione di un foglio di stile (*stylesheet*) XSL [21], denominato *Valid.xsl*, implementato in Ie5 per eseguire il filtraggio dinamico dei contenuti di un documento sulla base del contesto di navigazione. Fig. 2 mostra la definizione di tale foglio di stile: la prima parte è costituita da un semplice *template* che realizza una trasformazione di tipo identità, mentre la seconda parte è dedicata alla selezione temporale vera e propria del contenuto dei contesti di validità. L'elaborazione di elementi XML *valid* causa la copia nel risultato del contenuto del contesto di validità qualora una *condizione di selezione di validità* (che coinvolge i valori degli attributi *from* e *to*) sia verificata. Ad esempio, se tale condizione ha la forma:

```
@from[.le$'1999-12-31'] and @to[.ge$'1999-01-01'] ,
```

<sup>2</sup> Si noti che per non ledere la generalità non si impone a priori nessuna relazione (es. di contenimento) fra i *timestamp* di contesti temporali annidati.

---

```

<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl" >

<!-- identity transformation template -->
<xsl:template>
  <xsl:copy>
    <xsl:apply-templates select="@*|*|comment()|pi()|text()" />
  </xsl:copy>
</xsl:template>

<!-- recursive valid-time selection template -->
<xsl:template match="valid">
  <xsl:choose>
    <xsl:when test="validity[condizione sui valori degli attributi from e to ]">
      <xsl:copy>
        <xsl:apply-templates select="@*|*|comment()|pi()|text()" />
      </xsl:copy>
    </xsl:when>
    <xsl:otherwise>
      <xsl:apply-templates select="*//valid" />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>

</xsl:stylesheet>

```

---

**Figura 2.** Il foglio di stile XSL `Valid.xsl`.

---

ogni elemento `valid` i cui *timestamp* di `validity` si sovrappongono almeno in parte all'anno 1999 vengono inclusi nell'*output* generato dal foglio di stile.

La particolare struttura del *template* di selezione causa l'esecuzione di un *test* che coinvolge il contesto di navigazione e tutti i *timestamp* (elementi `<validity>`) trovati nell'elemento `<valid>` corrente. L'elaborazione condizionale impiega l'istruzione `xsl:choose` che dispone di un caso `xsl:otherwise`, al fine di ricercare ricorsivamente contesti di validità annidati qualora nessuno dei *timestamp* dell'ambiente `<valid>` corrente soddisfi la condizione di selezione, mentre l'istruzione `xsl:when` viene attivata se almeno uno dei *timestamp* soddisfa la condizione di selezione.

Si noti che se il contesto di validità di navigazione viene cambiato dall'utente nel corso della navigazione stessa, la condizione di selezione deve essere modificata dinamicamente all'interno del foglio di stile. Quando il foglio di stile sarà riapplicato, anche la visualizzazione del documento cambierà in accordo alla nuova condizione di selezione, riflettendo così la modifica del contesto di navigazione. Nella nostra implementazione prototipale, il foglio di stile viene modificato in maniera dinamica e riapplicato al documento per mezzo delle funzionalità rese disponibili dal DOM che agisce su una copia del foglio di stile caricata nello spazio di memoria principale gestito dal *browser* (o su una sua copia mantenuta nella *cache* del *browser* su un disco locale alla macchina sulla quale il *browser* è in esecuzione). Ad esempio, la condizione di sovrapp-



posizione con l'anno 1999 descritta sopra può essere applicata dinamicamente al documento in corso di visualizzazione attraverso lo *script* mostrato in Fig. 3.

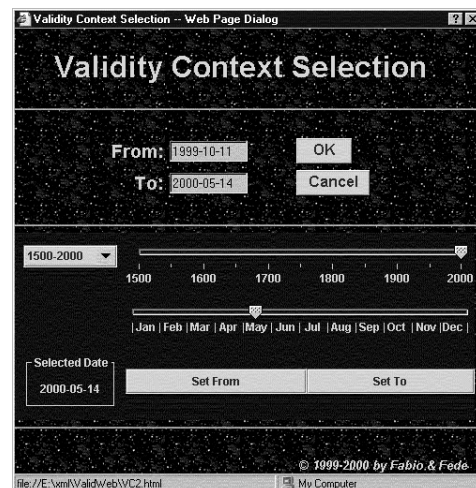
```
// Trova la condizione del test di validita' all'interno del foglio di stile.  
var sel = document.XSLDocument.selectSingleNode("*/xsl:when/@test");  
// Sostituiscine il valore con la condizione di sovrapposizione al 1999.  
sel.value = "validity[ @from[.<='1999-12-31'] and @to[.>='1999-01-01'] ]"  
// Applica lo stile modificato al documento e aggiorna la visualizzazione.  
document.body.innerHTML = document.XMLDocument.transformNode(document.XSLDocument)
```

**Figura 3.** Uno script DOM per cambiare dinamicamente il contesto di validità di navigazione.

Si noti come condizioni temporali ben più complesse della semplice sovrapposizione con un intervallo possano essere facilmente realizzate applicando una opportuna condizione di selezione all'interno del foglio di stile. Pertanto, ad un sito temporale possono anche essere aggiunte sofisticate funzionalità di interrogazione temporale, grazie ad una gestione diretta della condizione di selezione temporale all'interno di `Valid.xsl`. Essendo infatti basato sull'implementazione XSL di `Ie5`, la nostra proposta rappresenta una estensione temporale del linguaggio di interrogazione per il Web XQL [15]; estensione progettata e realizzata come applicazione dello stesso XQL. In particolare, nel nostro prototipo descritto nella prossima Sezione, l'adozione di opportuni filtri XSL permette di implementare tutti i predicati di selezione temporale del linguaggio di interrogazione TSQL2 [17] come mostrato in [6].

## 4 L'implementazione del prototipo

In questa sezione descriviamo l'implementazione del prototipo software realizzato per sperimentare e validare le nostre estensioni XML/XSL temporali. Il prototipo, chiamato "The Valid Web" [7,8], consiste di un sito Web che può essere visitato con `Ie5` (vedi Fig. 5). Le pagine Web del sito sono organizzate in due *frame*. Nella parte bassa della finestra del *browser* c'è un piccolo *frame* di servizio che contiene tutti i controlli necessari alla gestione interattiva del contesto di validità e che viene usato per la navigazione e interrogazione temporale, inclusa la visualizzazione del contesto corrente. Tutti i controlli sono implementati tramite funzioni JavaScript. Un *frame* più grande è invece deputato alla visualizzazione dei documenti temporali, ossia alla presentazione dei risultati del filtraggio temporalmente selettivo effettuato dallo stile `Valid.xsl`



**Figura 4.** Applet per la selezione del contesto di validità.

sui documenti XML dotati di *timestamp*. Il prodotto di tale filtraggio è nella nostra prima applicazione un semplice documento HTML che viene pertanto reso dal *browser* nel modo “solito”, mentre nella seconda applicazione di riferimento i risultati del filtraggio temporale dei dati XML sono poi sottoposti ad un secondo filtraggio di formattazione (con un secondo stile XSL) che li rende in forma di tabella HTML. La selezione del contesto di validità di navigazione consiste nella scelta di un intervallo, ovvero dei due istanti di tempo che ne rappresentano gli estremi. Nel nostro prototipo, la scelta di ciascun istante può essere effettuata agendo su due *slider*, implementati all'interno di un Applet, il primo per l'anno e il secondo per il giorno dell'anno (vedi Fig. 4), che consentono una selezione fine analogica di un data sull'asse dei tempi. In ogni caso sono sempre disponibili, nella finestra di dialogo che contiene l'Applet in esecuzione, dei campi di *input* in cui è possibile digitare direttamente il valore di data designato (in formato stringa “YYYY-MM-DD”).

#### 4.1 Visita del Museo Web temporale

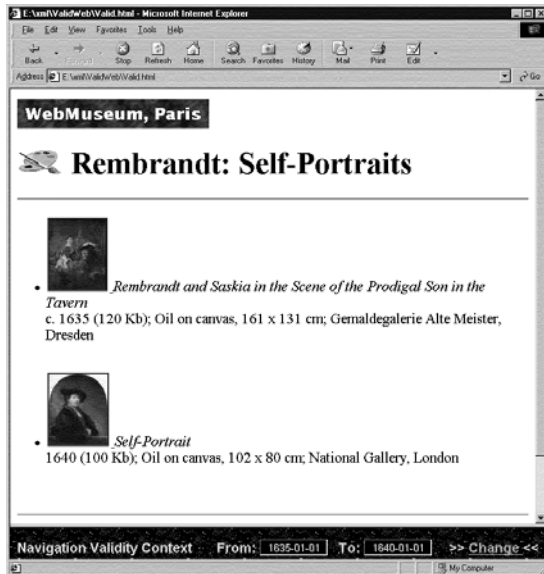
Nell'applicazione museale, ogni volta che l'utente cambia il contesto di validità di navigazione viene rieseguita la selezione temporale del materiale da visualizzare attraverso il meccanismo basato sul DOM descritto nella sezione 3. Inoltre, per consentire una vera e propria navigazione temporale, ogni volta che l'utente cambia il documento visualizzato nel modo solito (es. seguendo un *link*), il contesto di navigazione corrente deve essere automaticamente “ereditato” anche dalla nuova pagina caricata (naturalmente se si tratta ancora di una pagina XML temporale). Ciò è stato realizzato impiegando un foglio di stile XSL leggermente modificato rispetto a quello descritto in figura 3, che attua un meccanismo di *callback* dinamico tramite l'inserzione di codice JavaScript nel preambolo della pagina Web processata. Tale *script* causa, all'atto del caricamento del documento da parte del browser, l'attivazione immediata della funzione di selezione temporale: come quando si cambia il contesto di validità, il filtro XSL è modificato “al volo” (per includervi come condizione di selezione la condizione di sovrapposizione con l'attuale contesto di navigazione) e poi riapplicato al documento visualizzato per mezzo di chiamate ai metodi DOM. In questo modo la condizione di selezione di validità viene propagata automaticamente, in maniera del tutto trasparente, anche ai documenti appena caricati. Ad esempio, la metà superiore della Fig. 5 mostra due istantanee della navigazione di una pagina di esempio contenente alcuni autoritratti di Rembrandt. La pagina completa contiene sette dipinti, che datano dal 1629 al 1669. La Fig. 5.a mostra la pagina quando il contesto di validità è stato fissato a [1635–1640] e solo due immagini sono pertanto visibili (la terza e la quarta). La Fig. 5.b mostra la stessa pagina con contesto di validità cambiato in [1660–1665] dove quindi sono state visualizzate altre due immagini (la quinta e la sesta). Il contesto di navigazione corrente è sempre visibile nel *frame* di servizio in basso. Il comando “Change” sulla destra è un *link* che attiva l'Applet di Fig. 4.

#### 4.2 Interrogazione temporale dei dati di impiegati

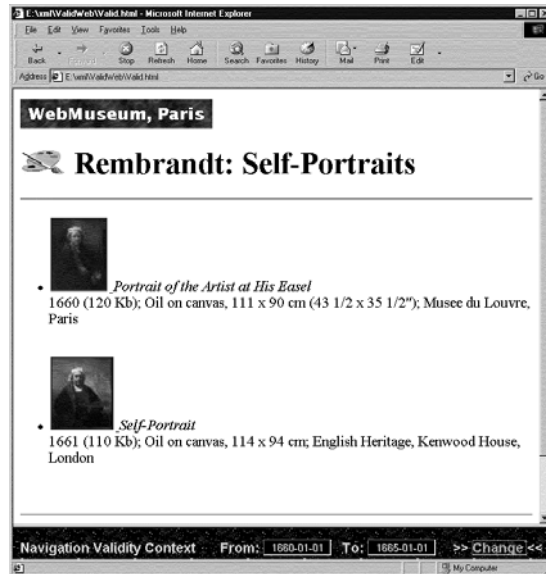
Nell'applicazione di gestione di dati temporali codificati in XML, il contesto di validità è utilizzato come argomento di predicati di selezione temporale di tipo TSQL2 per eseguire interrogazioni sui dati temporali degli impiegati contenuti in Tab. I. La classe di *query* semplici (su una sola relazione) che considereremo è del seguente tipo:

```
SELECT < target-list >
FROM Employee
WHERE VALID(Employee) < temporal-predicate > Validity_Context
```

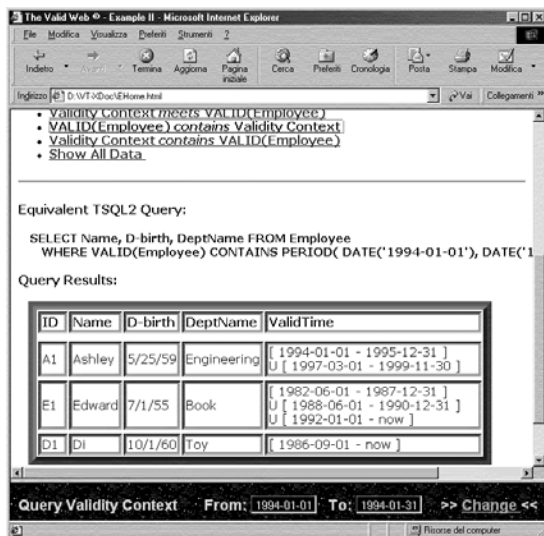
ove *Validity\_Context* è un intervallo temporale definito dall'utente e *<temporal-predicate>* rappresenta uno dei *predicati di selezione* in TSQL2 per effettuare un confronto fra un intervallo e un *temporal element*



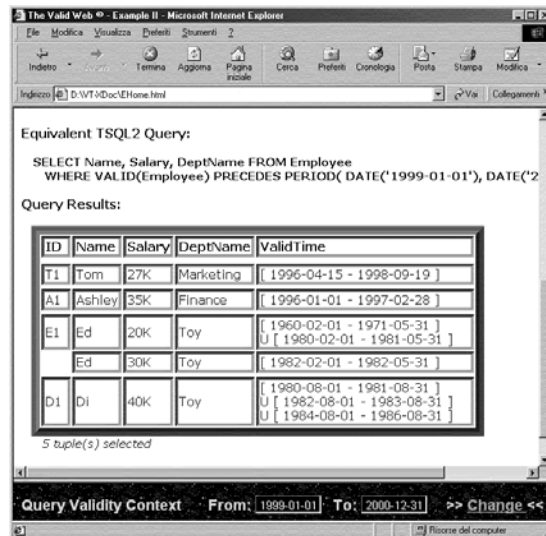
(a)



(b)



(c)



(d)

Figura 5. Il prototipo Valid Web: (a),(b) navigazione temporale del Museo Web (vedi Sez. 4.1); (c),(d) interrogazione temporale dei dati XML riguardanti impiegati (vedi Sez. 4.2).

(cioè: OVERLAPS, EQUALS, PRECEDES, CONTAINS, MEETS<sup>3</sup>). Considereremo anche ovviamente le *query* duali, in cui l'ordine degli operandi nei predicati temporali è invertito, quando non si tratti di operatori simmetrici (ossia nel caso di PRECEDES, CONTAINS, MEETS). Per esempio, la *query*:

```
SELECT * FROM Employee
WHERE PERIOD( DATE('1994-01-01'), DATE('1994-12-31') )
PRECEDES VALID(Employee)
```

seleziona tutti i dati degli impiegati la cui validità segue l'intervallo di tempo [1994] (cioè tutte le tuple tranne la terz'ultima in Tab. I).

La finestra principale dell'applicazione contiene un insieme di *link* che consentono la scelta del predicato temporale desiderato da parte dell'utente. Ciascun *link* attiva la chiamata ad una funzione JavaScript che modifica dinamicamente il foglio di stile `Valid.xsl` aggiornando la condizione di selezione ed esegue la *query*. Per esempio, il predicato di eguaglianza corrisponde alla seguente condizione di selezione XSL

```
validity[ index()=0 and end() ] and @from[.= VC from ] and @to[.= VC to ] ]
```

che controlla se l'intervallo in esame è allo stesso tempo il primo e l'ultimo dell'ambiente `<valid>` che lo contiene (prima condizione) e se i suoi estremi coincidono con quelli del contesto di validità di interrogazione (ultime due condizioni).

Anche la *target list* della *query* può essere definita dall'utente specificando gli attributi desiderati in uscita per mezzo di *checkbox* predisposti in un apposito *form*. Tale *form* è poi letto dall'applicazione prima dell'esecuzione della *query* per definire le specifiche di proiezione. La proiezione è implementata attraverso una sorta di *stylesheet* XSL parametrico per la traduzione dei risultati della *query* in formato tabellare HTML. Anche questo *stylesheet* è modificato dinamicamente dall'applicazione attraverso il meccanismo del DOM per abilitare la visualizzazione selettiva dei soli attributi prescelti dall'utente nella *target list*. Al termine dell'applicazione dello *stylesheet* XSL, il risultato finale è incluso dinamicamente nella pagina attiva (all'interno di un elemento HTML `<PRE>` utilizzando la *property* `innerHTML`) e visualizzato come una tabella.

Per esempio, la metà inferiore della Fig. 5 mostra due schermate corrispondenti all'esecuzione del prototipo. Le Figure 5.c e 5.d mostrano i risultati dell'esecuzione di due *query* sui dati XML degli impiegati. La *query* equivalente in forma TSQL2 è (parzialmente) visibile nelle immagini catturate, assieme al contesto di validità di interrogazione usato, che è presente nel *frame* inferiore delle finestre raffigurate.

Come esemplificato dall'applicazione dimostrativa, l'infrastruttura da noi proposta può essere usata, così com'è, per supportare *query* di selezione temporale di tipo TSQL2 su di una sorgente dati (ossia una "relazione" memorizzata come file XML). Il suo potere espressivo è in definitiva lo stesso del linguaggio XQL sottostante supportato dal Ie5. Pertanto non è relazionalmente completo, in quanto non comprende l'operazione di *join* (ossia *query* agenti su sorgenti XML multiple, cosa invece possibile con il linguaggio XML-QL [16]). Comunque, in aggiunta al prototipo qui presentato, abbiamo anche sviluppato una applicazione di prova nella quale si possono eseguire *join* temporali [10] fra sorgenti XML temporali tramite una apposita funzione JavaScript che implementa l'algoritmo *nested-loops* [6].

---

<sup>3</sup> In realtà, TSQL2 definisce MEETS solamente come operatore di confronto fra due *intervalli*. Perciò la clausola "VALID(Employee) MEETS Validity\_Context" dovrebbe più propriamente essere riscritta come "LAST(VALID(Employee)) MEETS Validity\_Context", in cui l'operatore LAST estrae l'ultimo intervallo componente da un *temporal element*. Di conseguenza intenderemo nel testo un operatore MEETS dalla semantica leggermente estesa in modo da operare nello stesso modo con un *element*. Ci scusiamo con il lettore per il veniale abuso.

## 5 Conclusioni e Lavoro Futuro

In questo lavoro abbiamo delineato un approccio per l'incorporazione del tempo di validità all'interno di documenti Web, considerando le estensioni richieste alla struttura e al formato dei documenti, e studiando le modalità di elaborazione specifiche richieste da essi (*browsing* e navigazione temporale). La soluzione da noi prospettata e realizzata si fonda su semplici estensioni basate sull'adozione di XML, ovvero sulla definizione del nuovo *tag* di *markup valid* con l'introduzione di uno schema XML per la definizione di documenti temporali e sull'uso di un foglio di stile XSL per il filtraggio selettivo dei documenti temporali in accordo ad un contesto di navigazione definito dall'utente. Abbiamo inoltre sviluppato un prototipo di sito Web temporale, accessibile con Microsoft Internet Explorer 5, che realizza le soluzioni proposte e fornisce strumenti per la gestione del contesto di validità di navigazione.

Nel nostro approccio, l'uso di documenti temporali di tipo *valid time* è perfettamente compatibile con l'odierna tecnologia XML, come dimostra il nostro prototipo di sito Web. Le soluzioni proposte sono inoltre compatibili anche con *client* Web non temporali, nel senso che tutta l'informazione contenuta in un documento temporale è accessibile (essendo i nuovi *tag* semplicemente ignorati), pur non essendo ovviamente disponibili le nuove funzionalità di navigazione temporale e visualizzazione selettiva.

Il nostro lavoro futuro sarà dedicato all'estensione dell'infrastruttura proposta. In particolare vorremmo consentire la selezione per mezzo di un qualunque tipo di contesto temporale, rilassando il vincolo attuale dell'uso di un intervallo. Ad esempio, il contesto di navigazione potrebbe anche essere "catturato" dal tempo di validità connesso ad un oggetto selezionato all'interno del documento in corso di visualizzazione. Si noti che, in generale, il contesto di validità così catturato sarebbe un *temporal element* piuttosto che un intervallo. Dal punto di vista teorico, verrà anche studiata la definizione di un modello generalizzato di rappresentazione di informazione temporale all'interno di documenti Web memorizzati tramite XML, cui possano essere riconducibili modelli "classici" adottati per le basi di dati temporali relazionali e orientate agli oggetti, ma anche in grado di catturare le caratteristiche tempo-varianti di informazioni multimediali anche non strutturate. Per la definizione di un tale modello generalizzato, lo schema di *timestamping* qui proposto non rappresenta che una soluzione parziale e preliminare. Una ulteriore estensione dell'infrastruttura potrà anche comprendere l'introduzione di un vero e proprio linguaggio di interrogazione temporale per consentire *query* temporali complesse sul Web. A tale proposito intendiamo considerare l'estensione di un linguaggio di interrogazione quale XML-QL [15] per supportare generiche condizioni di selezione temporale sui documenti XML dotati di *timestamp*. Le estensioni temporali richieste verranno basate sulle stesse funzionalità del linguaggio temporale TSQL2 [17] in modo da eguagliarne anche il medesimo potere espressivo in quanto a selezione e proiezione temporale. Ulteriori estensioni riguarderanno l'aggiunta di altre funzionalità di tipo TSQL2, quali il trattamento di tempi indeterminati e l'uso di diversi calendari; funzionalità di estremo interesse per la gestione di documenti finalizzata alla ricerca storica vera e propria.

In parallelo all'estensione dell'approccio sarà anche condotta un'analisi critica dell'approccio stesso, tendente a porre in evidenza la sua specificità rispetto al trattamento delle informazioni effettuato nelle basi di dati temporali con particolare riferimento al contesto Web/XML, ovvero a scenari applicativi soggetti a rapida e continua evoluzione ed alla necessità di gestire informazioni multimediali non strutturate.

## Riferimenti bibliografici

1. S. Abiteboul, P. Buneman, D. Suciu, *Data on the Web: From Relations to Semistructured Data and XML*, Morgan Kaufmann Publishers, San Francisco, California, 1999.
2. Archives & Museums Informatics, URL: <http://www.archimuse.com>.
3. C. Cristofori, F. Grandi, F. Mandreoli, M.R. Scalas, "Gestione di versioni temporali di risorse nel World Wide Web secondo il tempo di transazione", Rapporto tecnico INTERDATA, T6-R03, luglio 1998, URL: <http://www.difa.unibas.it/users/gmecca/interdata/prodotti/T6-R03.ps>.

4. A. Discenza, F. Garzotto, "Design Patterns for Museum Web Sites", Rapporto tecnico INTERDATA, T6-R10, dicembre 1999, URL: <http://www.difa.unibas.it/users/gmecca/interdata/prodotti/T6-R10.ps>.
5. F. Grandi, F. Mandreoli, M.R. Scalas, "Gestione di versioni temporali di risorse nel World Wide Web secondo il tempo di validità", Rapporto tecnico INTERDATA, T6-R13, agosto 1999, URL: <http://www.difa.unibas.it/users/gmecca/interdata/prodotti/T6-R13.ps>.
6. F. Grandi, F. Mandreoli, "The Valid Web: it's Time to Go...", Rapporto tecnico TIMECENTER, TR-46, dicembre 1999, URL: <http://www.cs.auc.dk/research/DP/tdb/TimeCenter/TimeCenterPublications/TR-46.pdf>.
7. F. Grandi, F. Mandreoli, "Prototipo di client Web con supporto di versioni secondo il tempo di validità", Prototipo software INTERDATA, T6-S20, febbraio 2000, URL: <http://www.difa.unibas.it/users/gmecca/interdata/prodotti/T6-S20.html>.
8. F. Grandi, F. Mandreoli, "The Valid Web ©", *Proc. of Software Demonstrations Track at the EDBT'2000 Intl. Conference*, Konstanz, Germania, marzo 2000.
9. F. Grandi, M. R. Scalas, "Extending Temporal Database Concepts to the World Wide Web", *Atti del Sesto Convegno Nazionale su Sistemi Evoluti per Basi di Dati*, Ancona, Italia, 1998.
10. C.S. Jensen, J. Clifford, R. Elmasri, S.K. Gadia, P. Hayes, S. Jajodia (eds.) et al., "A Consensus Glossary of Temporal Database Concepts - February 1998 Version", in O. Etzion, S. Jajodia and S. Sripada (eds.), *Temporal Databases - Research and practice*, LNCS N. 1399, Springer-Verlag, 1998.
11. C.S. Jensen, M.D. Soo, R.T. Snodgrass, "Unifying Temporal Data Models via a Conceptual Model", *Information Systems*, Vol. 19, N. 7, 1994.
12. N. Pioch, "The Web Museum," URL: <http://www.cnam.fr/wm/>.
13. *Proc. of QL'98 - The W3C Query Languages Workshop*, Boston, MA, Dec. 1998, URL: <http://www.w3.org/TandS/QL/QL98/>.
14. Prolog and Document Type Declaration, in *Extensible Markup Language (XML) 1.0*, W3C Consortium Recommendation, URL: <http://www.w3.org/TR/REC-xml#sec-prolog-dtd>.
15. J. Robie, J. Lapp, D. Schach, "XML Query Language (XQL)", in [13], URL: <http://www.w3.org/TandS/QL/QL98/pp/xql.html>.
16. D. Schach, J. Lapp, J. Robie, "Querying and Transforming XML", in [13], URL: <http://www.w3.org/TandS/QL/QL98/pp/query-transform.html>.
17. R.T. Snodgrass (ed.), I. Ahn, G. Ariav, D. Batory, J. Clifford, C.E. Dyreson, R. Elmasri, F. Grandi, C.S. Jensen, W. Käfer, N. Kline, K. Kulkarni, T.Y. Cliff Leung, N. Lorentzos, R. Ramakrishnan, J.F. Roddick, A. Segev, M.D. Soo, S.M. Sripada, *The TSQL2 Temporal Query Language*, Kluwer Academic Publishers, Boston, Massachusetts, 1995.
18. A. Tansel, J. Clifford, V. Gadia, S. Jajodia, A. Segev, R.T. Snodgrass (eds.), *Temporal Databases: Theory, Design and Implementation*, Benjamin/Cummings Publishing Company, Redwood City, California, 1993.
19. The Document Object Model (DOM) Home Page, W3C Consortium, URL: <http://www.w3.org/DOM/>.
20. The Extensible Markup Language (XML) Resource Page, W3C Consortium, URL: <http://www.w3.org/XML/>.
21. The Extensible Stylesheet Language (XSL) Resource Page, W3C Consortium, URL: <http://www.w3.org/Style/XSL/>.
22. The Microsoft Internet Explorer Home Page, Microsoft, URL: <http://microsoft.com/windows/Ie/>.
23. The World Wide Web Consortium (W3C) Home Page, URL: <http://www.w3.org/>.
24. V.J. Tsostras, A. Kumar, "Temporal Database Bibliography Update", *ACM SIGMOD Record*, Vol. 25, N. 1, 1996.
25. XML-Data, W3C Consortium Note, URL: <http://www.w3.org/TR/1998/NOTE-XML-data/>.
26. XSL Transformations (XSLT) Version 1.0, W3C Consortium Working Draft, URL: <http://www.w3.org/TR/xslt>.