Codifica XML e Gestione di Informazione Temporale in Fonti Storiche Digitalizzate di Grandi Dimensioni

XML Encoding and Manipulation of Temporal Information in Large Digitized Historical Text Sources

Fabio Grandi ♯, *Federica Mandreoli* ♭

♯ *C.S.I.TE.-C.N.R. e D.E.I.S. - Università di Bologna, fgrandi@deis.unibo.it*
♭ *D.I.I. - Università di Modena e Reggio Emilia, fmandreoli@dsi.unimo.it*

Sommario

Questo lavoro tratta dell'impiego di tecnologie legate all'XML per applicazioni nel campo dei Beni Culturali che sfruttino la codifica di semantica temporale nella gestione di documenti storici in forma elettronica. La ricerca è inserita nel contesto di un progetto mirato alla produzione di una versione digitale XML fruibile via Internet del dizionario *Repetti* (XIX secolo), di grande interesse per lo studio della storia e dell'archeologia medievale della Toscana. In particolare presentiamo una proposta di classificazione e codifica uniforme delle informazioni temporali contenute in fonti testuali caratterizzate da indeterminazione e uso di granularità e calendari multipli. Tale proposta si basa sull'estensione dell'approccio probabilistico (alla TSQL2) all'indeterminazione, con l'introduzione di distribuzioni di probabilità costanti a tratti, che risultano essere corrette da un punto di vista semantico e si prestano ad elaborazioni particolarmente efficienti. L'articolo contiene inoltre una breve descrizione di due strumenti i cui prototipi sono in avanzata fase di realizzazione: un *tool* di sviluppo, di uso amichevole, per l'introduzione assistita della marcatura temporale all'interno dei documenti e un sistema per la gestione della collezione di documenti XML che rende disponibile tramite il Web un efficiente motore di ricerca temporale.

The paper deals with the deployment of XML-related technologies in Cultural Heritage applications concerning the encoding of temporal semantics in the digital version of historical documents. The present research is included in the context of a project aimed at publishing on the Web an XML-based electronic edition of the Repetti dictionary (XIX century), extremely interesting for historical and archeological studies of the Tuscany Middle Ages. In particular, we introduce a proposal for the uniform encoding and classification of temporal information embedded in textual sources which are characterized by indeterminancy, multiple granularities and calendars. Our proposal is based on the extension of the probabilistic approach (à la TSQL2) to indeterminancy, with the introduction of piecewise-constant probability distributions, which are semantically correct and particularly efficient in their management. The paper also contains a brief description of two tools whose prototypes are carried on: a user-friendly tool for the computer-aided encoding of temporal XML documents and a system for the management of XML documents consisting in a temporal search engine accessible via standard Web browsers.

1 Introduzione

In campo informatico XML [18] si propone come lo standard emergente per la gestione e lo scambio di dati su Internet [1]. In particolare, vi è un grande interesse per la sua adozione come strumento comune per la rappresentazione e l'integrazione di dati strutturati e non. Una peculiarità di XML, che lo rende inoltre

estremamente interessante per applicazioni nel settore dei Beni Culturali, è la possibilità di codificare in modo semplice all'interno di documenti XML informazioni di tipo semantico sotto forma di *metadati* che possano poi essere interpretati ed utilizzati in modo automatico da strumenti informatici avanzati, quali motori di ricerca "intelligenti" (nella direzione di realizzare un cosiddetto *Semantic Web* [20]).

Il recente passato ci ha visto, come gruppo di ricerca, coinvolti nello sviluppo di un'infrastruttura XML/XSL per la gestione di dati e documenti temporali in ambiente Web, denominata "The Valid Web", mutuando ed adattando al contesto Web [11] concetti e tecniche sviluppate dalla ricerca sulle basi di dati temporali [17, 6]. Tale infrastruttura, descritta ad esempio in [8], è stata implementata su di un prototipo sperimentale [7] fruibile tramite un browser XML-compatibile ed è disponibile *on-line*.Una naturale estensione dell'approccio "The Valid Web" ci ha portato a collaborare con il gruppo di ricerca su informatica e storia coordinato da Franco Niccolucci presso l'Università di Firenze e, in particolare, a prendere parte al cosiddetto progetto "XML/Repetti" [14].

Il "*Dizionario geografico, fisico, storico della Toscana*" di Emanuele Repetti (nel seguito: il *Repetti*) è una raccolta enciclopedica di informazioni concernenti la Toscana edita in otto volumi fra il 1833 ed il 1846. Il *Repetti* ha subito nella sua storia diverse riproduzioni anastatiche ed è attualmente allo studio una sua riedizione in forma elettronica, eventualmente accessibile tramite Internet. Questo requisito è paricolarmente importante da un punto di vista scientifico, dato che in archeologia medievale le fonti scritte hanno spesso la stessa importanza dei reperti materiali e la condivisione delle informazioni tramite il Web ha già dimostrato un impatto rilevante sulla ricerca archeologica [12].

L'edizione elettronica del *Repetti* dovrà prevedere l'implementazione di un *motore di ricerca* basato sul contenuto delle voci del dizionario di immediato utilizzo da parte del ricercatore storico e archeologico. A tale scopo l'adozione del linguaggio di *markup* XML consente la classificazione uniforme dell'informazione di tipo testuale contenuta nel dizionario. In particolare l'applicazione ed estensione dell'approccio "The Valid Web" al progetto "XML/Repetti" [9] è mirata alla classificazione e codifica delle informazioni temporali contenute nel dizionario ed alla predisposizione di un corrispondente ausilio per la ricerca, e presuppone tre obiettivi fondamentali: **(a)** estensione dell'infrastruttura XML, in particolare della metodologia di codifica e della logica di elaborazione, per far fronte alla specificità dell'informazione temporale contenuta nel *Repetti* e in documenti analoghi; **(b)** ridefinizione dell'architettura globale di sistema, con una organizzazione efficiente del motore di ricerca temporale (basato su algoritmi di ricerca ottimizzati) e del *repository* XML (con eventuale utilizzo di indici temporali); **(c)** predisposizione di un *tool* di semplice utilizzo per l'introduzione assistita dei *tag* temporali all'interno dei documenti, che eviti il più possibile al ricercatore storico una codifica "manuale" della necessaria marcatura temporale.

Nello specifico, l'estensione dell'infrastruttura necessita di un arricchimento del sistema di marcatura e del "meccanismo" di ricerca in grado di catturare la semantica di espressioni temporali, assai diffuse nel *Repetti*, che coinvolgono: **indeterminazione** (come in: "verso la fine del XV secolo"), **diversi calendari** (come nell'uso del calendario Giuliano) e **diverse granularità** (es. mesi *versus* anni). In particolare, ampio spazio sarà dedicato al problema dell'indeterminazione che presenta i risvolti teorici più interessanti e per il quale è stato necessario introdurre le estensioni dell'infrastruttura più consistenti.

Il resto del lavoro è organizzato in tre sezioni principali: nella prima si trova una descrizione delle estensioni dell'infrastruttura che sono state studiate e che si intende implementare, mentre nella seconda è presente una descrizione preliminare del *tool* per il *markup* assistito che verrà ultimato in un prossimo futuro. La terza è dedicata ad una sommaria descrizione dell'implementazione della versione digitale del *Repetti* e del motore di ricerca in corso di realizzazione.

2 Indeterminazione Temporale nelle Fonti Storiche

È comune rinvenire nella lettura di un testo in linguaggio naturale espressioni temporali vaghe ed imprecise, quali date e periodi di tempo descritti attraverso locuzioni difficilmente formalizzabili, che riguardano la validità di un fatto storico. In questo contesto, l'unità base di riferimento sull'asse dei tempi è il *giorno*

e quindi ci occuperemo fondamentalmente di *date*. La validità di un fatto storico può essere istantanea, per un evento accaduto in una particolare data, oppure rappresentabile tramite un intervallo di tempo (un insieme di giorni consecutivi) per fatti con durata non nulla. Nell'interpretazione dell'espressione testuale, il vero problema è l'indeterminazione; la compresenza di diversi calendari e granularità non introduce (come diverrà anche più chiaro nel prosieguo del lavoro) complicazioni di sorta, avendo tutti i calendari in uso il giorno come granularità di base ed utilizzando apparentemente tutti lo stesso reticolo di granularità (in pratica, le uniche di interesse sono sempre: giorno, mese, anno e secolo), dimodoché le opportune funzioni di conversione possono essere facilmente predisposte.

Partendo dall'analisi di un vasto *corpus* di fonti storiche quale il *Repetti*, è possibile classificare le espressioni temporali che denotano eventi indeterminati in quattro categorie principali [10, 9]. Definendo "Espressione Temporale di Riferimento" (ETR) il valore letterale di tempo scritto nel testo, le quattro categorie corrispondono all'uso di espressioni aventi rispettivamente la seguente forma: "in ETR" (per riferirsi ad una validità di durata inferiore a quella dell'ETR) per la classe C_1, "all'inizio (fine) di ETR" per C_2 (C_3), "intorno a ETR" per C_4 come nei seguenti esempi:

- L'abbazia fu consacrata a S. Martino **nel 1276**. (C_1)
- La terza cinta delle mura cittadine fu aggiunta **all'inizio del XIV secolo**. (C_2)
- Il celebre pittore morì di peste **verso la fine del marzo 1532**. (C_3)
- La delegazione imperiale arrivo a Roma **intorno al Natale del 1467**. (C_4)

Si noti come nel caso C_1, peraltro assai frequente, ci si trovi di fronte ad un cosiddetto caso di *granularity mismatch* [5], in cui un'espressione determinata a granularità superiore è usata per denotare un'espressione indeterminata a granularità inferiore. È molto probabile infatti che l'esempio si riferisca ad un evento avvenuto in un giorno preciso dell'anno 1276, piuttosto che ad un'attività protrattasi per l'intero anno. Dato che poi non è possibile stabilire a priori di quale giorno eventualmente si tratti, non vi è ragione di preferire una data piuttosto di un'altra. L'esempio "Il castello fu ricostruito dopo l'incendio **tra il 1549 e il 1553**" riguarda invece un vero intervallo dato che l'azione di ricostruzione ha presumibilmente richiesto parecchi anni per essere completata. Se però non è dato sapere le esatte date di inizio e fine dei lavori, l'espressione denota un intervallo indeterminato, i cui estremi sono date di tipo C_1.

In definitiva qualsiasi espressione temporale rinvenuta nel testo può essere ricondotta ad una data indeterminata, o ad un intervallo i cui estremi siano date indeterminate, che ricadono in una delle quattro categorie sopracitate, della cui rappresentazione si occupa il Capitolo seguente.

2.1 Rappresentazione di date indeterminate

Nel campo delle basi di dati temporali esistono due approcci principali alla gestione dell'indeterminazione temporale: quello *probabilistico*, che è stato introdotto nel progetto del linguaggio TSQL2 [16] e ulteriormente sviluppato da Dyreson e Snodgrass in [5] e quello *fuzzy* proposto da Dutta [4]. Per una rassegna di altri approcci alternativi si rimanda alla discussione in [5] di cui si condividono le conclusioni. I due approcci sono profondamente differenti per quanto riguarda la rappresentazione di informazione temporalmente incompleta e, nel nostro caso, si preferisce il secondo dove una *validità indeterminata* è collegata all'occorrenza di un evento, che rimane comunque concettualmente singola, anche se di essa è nota soltanto una distribuzione di probabilità. Un istante indeterminato diviene quindi un insieme di possibili alternative, di cui una soltanto ne rappresenta la validità reale. Analogamente l'evento storico (es. la morte di un Re) da noi gestito deve essere accaduto in una data ben precisa, anche se dalle fonti non ne viene tramandata una designazione univoca e ben specificata. A ciò si aggiunga anche che nel caso di TSQL2 sono state proposte tecniche computazionalmente efficienti per la rappresentazione ed elaborazione di tempi indeterminati [5], cosa assolutamente necessaria per rendere fattibile il trattamento di lunghi testi contenenti migliaia di espressioni temporali indeterminate quali il *Repetti*.

Categoria	Forma	Densità associata	Nome	N
C_1	piatta	uniforme	DURING	1
C_2	asimmetrica decrescente	esponenziale	VERY_EARLY	3
			EARLY	4
C_3	asimmetrica crescente	esponenziale rovesciata	VERY_LATE	3
			LATE	4
C_4	simmetrica con picco centrale	normale	STRICTLY_AROUND	3
			AROUND	5
			WIDELY_AROUND	7

Tabella I: Distribuzioni di probabilità associate ad eventi indeterminati.

Nel modello probabilistico, un evento indeterminato t è rappresentato tramite la sua distribuzione di probabilità P, non nulla solamente all'interno di un intervallo di possibile occorrenza, i cui estremi (t^- e t^+) sono detti *supporto inferiore* e *supporto superiore*: $t = (t^- \sim t^+, P)$ ove $P(i) = \Pr[t = i]$ con $\sum_{i=t^-}^{t^+} P(i) = 1$ e $P(i) = 0$ se $i < t^-$ o $i > t^+$. Al fine di valutare le interrogazioni che li coinvolgono, due istanti indeterminati sono considerati equivalenti ($t_1 \equiv t_2$) se e solo se hanno i medesimi supporti e distribuzione. Allo stesso scopo TSQL2 definisce anche un'opportuna estensione della relazione d'ordine temporale, ovvero della primitiva "*Before()*" su cui tutti gli altri operatori di confronto temporale sono basati [16]. Nella semantica indeterminata, la primitiva "*Before()*" include un primo parametro addizionale per la specifica di una *plausibilità* di ordinamento, di valore compreso tra 0 e 100 (alta plausibilità significa elevata probabilità di precedenza tra i due istanti confrontati). La sua definizione completa diventa quindi: *Before*$(p, t_1, t_2) = \neg(t_1 \equiv t_2) \wedge \Pr[t_1 < t_2] \geq p/100$ ove la probabilità di precedenza è valutata come:

$$\Pr[t_1 < t_2] = \sum_{i<j} P_1(i) P_2(j) \tag{1}$$

essendo $P_k(x)$ la probabilità di occorrenza di t_k all'istante x. Si noti che poiché la valutazione del *Before()* si basa sull'indipendenza statistica delle occorrenze di t_1 e t_2, l'uso di "*Before()*" mal si presta ad una corretta valutazione probabilistica di un ordinamento $t_1 < t_2 < t_3$ in termini di "*Before*$(p, t_1, t_2) \wedge$ *Before*(p, t_2, t_3)" quando gli intervalli di supporto di t_1 e t_3 non siano disgiunti.

Per quanto riguarda le possibili distribuzioni di probabilità, sono state prescelte, per la loro semplicità analitica ma nel contempo correttezza dal punto di vista semantico, delle funzioni *costanti a tratti* su intervalli di eguale ampiezza (*intervalli base*) in cui può essere suddiviso l'intervallo fra i supporti. A questo punto introduciamo una corrispondenza fra tali funzioni e le quattro categorie di espressioni temporali individuate in precedenza. Le "forme" prescelte sono le più semplici compatibili con il significato "naturale" delle categorie, ovvero: piatta per C_1, asimmetrica decrescente (crescente) per C_2 (C_3), simmetrica con un solo picco centrale per C_4. Oltre che dalla distribuzione, una data indeterminata resta individuata, in alternativa ai supporti, dal suo *intervallo principale*, ovvero dall'intervallo base in cui assume il valore massimo. Per le categorie C_1 e C_4 l'intervallo principale corrisponde esattamente alla ETR scritta nel testo. Per C_2 (C_3) l'intervallo principale è il primo (ultimo) intervallo contenuto nell'ETR al livello di granularità immediatamente inferiore (es. è gennaio 1630 per l'espressione "all'inizio del 1630" con ETR 1630). I singoli valori assunti dalla probabilità sugli intervalli base sono stati calcolati tramite una densità continua *associata*. In assenza di ipotesi più ragionevoli, abbiamo scelto come densità associata la funzione più semplice avente le medesime caratteristiche di forma[1]. La Tab. I riassume i risultati; per le distribuzioni esclusa l'uniforme abbiamo anche considerato varianti consistenti in una minore o maggiore dispersione attorno al valor medio (es. VERY_LATE, WIDELY_AROUND, ecc.), che implicano corrispondentemente un differente numero N di intervalli base.

[1]Più formalmente, abbiamo operato le scelte che massimizzano l'entropia della variabile casuale quando non sono disponibili ulteriori informazioni (es. momenti di ordine superiore).

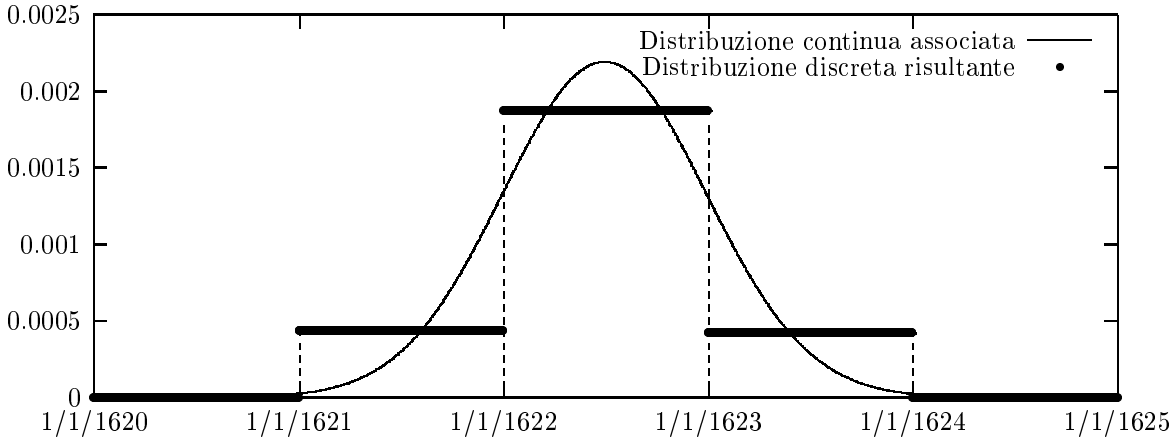

Figura 1: La distribuzione `STRICTLY_AROUND`.

Chiariamo la *ratio* di questo schema di codifica con un esempio. Supponiamo di voler interpretare e codificare l'espressione "intorno all'anno 1622" (C_4) e di poter escludere che l'evento in oggetto possa essere accaduto prima del 1621 o dopo il 1623 (N=3, `STRICTLY_AROUND`), mentre c'è una certa probabilità che sia accaduto nel 1621 o nel 1623 (in entrambi i casi con probabilità p'), pur essendo massima la probabilità (p''; ovviamente si ha che $2p' + p'' = 1$) che sia accaduto proprio nel 1622 (che è l'intervallo principale). Pertanto la distribuzione di probabilità è costante a tratti e risulta:

$$P(i) = \begin{cases} 0 & \text{se } i < \text{1/1/1621 o } i > \text{31/12/1623} \\ p'/365 & \text{se 1/1/1621} \leq i \leq \text{31/12/1621 o 1/1/1623} \leq i \leq \text{31/12/1623} \\ p''/365 & \text{se 1/1/1622} \leq i \leq \text{31/12/1622} \end{cases}$$

(essendo non bisestili i tre anni all'interno dei supporti). I valori p' e p'' possono essere determinati come valor medio assunto dalla distribuzione continua associata (normale) sui medesimi intervalli di tempo. La varianza della normale associata è scelta in modo tale che il 99,75% della probabilità sia contenuto fra i due supporti (essendo cioè trascurabile il contributo delle code escluse). Le funzioni di densità risultanti sono mostrate in Fig. 1 e si ha $p' \simeq 15,8\%$ e $p'' \simeq 68,4\%$.

2.2 Codifica XML di date indeterminate

Come estensione alla tecnica di *markup* adottata nell'approccio "The Valid Web", descriviamo qui la nostra proposta per la codifica di date indeterminate all'interno di documenti XML temporali che sarà applicata al *Repetti*. In questo caso, il "contesto" di validità di interesse per una ricerca all'interno del documento è individuabile nella *voce* del dizionario. Il contenuto di ciascuna voce sarà pertanto racchiuso in una coppia di *tag* `<ITEM>` ... `</ITEM>`, che costituirà il *target* principale per il funzionamento di un motore di ricerca (l'ambiente `<ITEM>` si sostituirà cioè all'ambiente `<valid>` impiegato nel più generale approccio "The Valid Web" [8]). Le frasi che compongono la voce possono contenere numerose espressioni temporali che verranno codificate per essere usate come *timestamp* della voce: se si è interessati ad un periodo storico si possono così ricercare tutte le voci che contengono almeno un *timestamp* che si sovrapponga al periodo stesso. Per la codifica possiamo introdurre un "tipo base" `DATE`, da utilizzare singolarmente o in coppia per la rappresentazione, rispettivamente, di eventi ed intervalli. Sulla base del tipo `DATE` sarà quindi possibile definire i *tag* `<EVENT>` e `<INTERVAL>`. Il *tag* `<EVENT>` conterrà un *element* XML `<AT>`, di tipo `DATE`, mentre il *tag* `<INTERVAL>` conterrà gli *element* `<FROM>` e `<TO>`, entrambi di tipo `DATE`. In questa maniera, eventi saranno rappresentati tramite strutture del tipo:

```
<EVENT>
   <AT ...  /> testo espressione temporale (evento)
</EVENT>
```

mentre per gli intervalli la marcatura sarà del tipo:

```
<INTERVAL>
   <FROM ...  /> <TO ...  /> testo espressione temporale (intervallo)
</INTERVAL>
```

Il tipo base `DATE` ha diversi attributi, alcuni dei quali specifici per l'espressione di date indeterminate: `GRANULARITY`, che permette di specificare la granularità con cui è espresso il valore della data come `"DAY"` (*default*), `"MONTH"`, `"YEAR"` o `"CENTURY"`; `VALUE`, che permette di specificare il valore dell'ETR (ovviamente in maniera consistente con la granularità assegnata); `CALENDAR`, che permette di far riferimento ad uno specifico calendario (con *default* `"GREGORIAN"`); `INDETETERMINATE`, con valori `"YES"` oppure `"NO"` (*default*), che consente di specificare se la data è espressa in forma indeterminata o meno. Nel caso l'attributo abbia valore `"YES"`, hanno significato gli ulteriori attributi: `DISTRIBUTION`, il cui valore può essere una delle sette distribuzioni di probabilità previste (con relative varianti, vedi Tab. I); `DURATION`, che esprime (con *default* `"1"`) in multipli della granularità, l'ampiezza degli intervalli base. Nel caso indeterminato, l'intervallo principale rimane così definito in modo *implicito* attraverso un intervallo avente come estremo inferiore il primo giorno dell'espressione temporale in `VALUE` (es. 1/1/1456 per `VALUE="1456"`; valutato anche in base al calendario specificato) ed ampiezza valutabile come dimensione in giorni della granularità specificata `GRANULARITY` moltiplicata per il valore dell'attributo `DURATION`.

La nostra espressione di esempio, con distribuzione in Fig. 1, potrà così essere semplicemente codificata come:

```
<EVENT>
   <AT VALUE="1622" GRANULARITY="YEAR"
       INDETERMINATE="YES" DISTRIBUTION="STRICTLY_AROUND" />
           intorno all'anno 1622
</EVENT>
```

piuttosto dell'espressione, se si preferisce usare la granularità del giorno, equivalente:

```
<EVENT>
   <AT VALUE="1622-01-01" DURATION="365"
       INDETERMINATE="YES" DISTRIBUTION="STRICTLY_AROUND" />
           intorno all'anno 1622
</EVENT>
```

Si noti come in ogni caso ci si possa così focalizzare sulla codifica dell'intervallo principale della distribuzione, che ha una corrispondenza diretta con l'ETR presente nel testo (il 1622 nell'esempio). Altrimenti l'espressione esplicita dei supporti presupporrebbe una conoscenza dettagliata della forma delle distribuzioni in gioco: nel nostro esempio, la distribuzione `STRICTLY_AROUND` ha tre intervalli base ("lobi" in Fig. 1) e pertanto i supporti risultano $1/1/1621$ e $31/12/1623$ ma, se fosse stata definita con cinque, essi sarebbero $1/1/1620$ e $31/12/1624$. La disponibilità di distribuzioni parametriche predefinite e lo schema di codifica implicita dei supporti rende la scelta del *markup* un po' più "trasparente" e amichevole, cosicché l'utente (ovvero il ricercatore storico) possa concentrarsi maggiormente sull'interpretazione del testo e sulla scelta di un "fattore di forma" intuitivo fra poche alternative disponibili piuttosto che sui dettagli matematici dei parametri della distribuzione quali il calcolo dei supporti o la varianza. I supporti e gli intervalli base possono comunque essere calcolati automaticamente in maniera elementare partendo dall'intervallo principale e dalla distribuzione.

Si noti anche come una scelta uniforme di codifica in cui valore e granularità specificate per l'intervallo base corrispondono *esattamente* all'ETR usata nel documento (regola di **codifica rigorosa**), costituisca di

```
<!DOCTYPE DICTIONARY[
<!ENTITY % DATE " CDATA #REQUIRED " >
<!ENTITY % DATE_ATTR
    " GRANULARITY (DAY|MONTH|YEAR|CENTURY) 'DAY'
      VALUE CDATA #REQUIRED
      INDETERMINATE (YES|NO) 'NO'
      DISTRIBUTION (DURING|EARLY|VERY_EARLY|LATE|VERY_LATE|
                    AROUND|STRICTLY_AROUND|WIDELY_AROUND) #IMPLIED
      DURATION CDATA '1'
      CALENDAR (GREGORIAN|ROMAN|JULIAN) 'GREGORIAN' " >
<!ELEMENT DICTIONARY (ITEM)* >
<!ELEMENT ITEM (#PCDATA|EVENT|INTERVAL)* >
<!ELEMENT EVENT (AT,#PCDATA) >
<!ELEMENT INTERVAL (FROM,TO,#PCDATA) >
<!ELEMENT AT %DATE; >
<!ATTLIST AT %DATE_ATTR; >
<!ELEMENT FROM %DATE; >
<!ATTLIST FROM %DATE_ATTR; >
<!ELEMENT TO %DATE; >
<!ATTLIST TO %DATE_ATTR; > ]>
```

Figura 2: DTD (parziale) per la codifica avanzata di informazioni temporali nel progetto "XML/Repetti". Le varianti di stile ai calendari sono state omesse per semplicità.

per sé *metainformazione* (sull'originale forma del contenuto del testo) da poter essere utilizzata per ricerche avanzate. Diverse ETR possono inoltre essere specificate a diversi livelli di granularità e in riferimento a calendari diversi. Nel nostro caso, questa possibilità ha conseguenze solamente sulla corretta conversione fra l'ETR e l'intervallo principale della distribuzione. Tutti gli intervalli principali sono infatti "ancorati" sull'asse dei tempi con base giorno e ogni altra granularità può essere facilmente convertita in giorni, per qualunque calendario di uso comune nelle fonti storiche. Ciononostante è possibile mantenere traccia della forma originaria (incluse granularità e calendario) dell'ETR nella codifica, costituendo metainformazione aggiuntiva. Ai fini del trattamento automatico, anche se espresse secondo diversi calendari, tutte le date in gioco possono comunque essere convertite secondo un calendario comune di riferimento (gregoriano) prima di ogni altra operazione (es. confronti). L'uso opzionale di speciali stili[2] può essere codificato attraverso varianti al calendario (es. `GREGORIAN_FLORENCE_STYLE` per lo stile "*Fiorentino*" del calendario gregoriano).

Per riferimento, riportiamo in Fig. 2 una Document Type Definition (DTD [15]) semplificata per la codifica XML delle espressioni temporali nel *Repetti*. Il tipo base `DATE`, con i suoi attributi, è stato qui codificato come "macro" di tipo `ENTITY`. Sottolineiamo il fatto che, una volta approvato, il nostro contributo diverrà parte del lavoro globale in svolgimento presso l'Univeristà di Firenze sull'edizione elettronica del *Repetti*. In questo contesto l'attività degli altri ricercatori [14, 2] è già orientata verso lo sviluppo di una DTD completa, ricavata anche adattando lavoro precedente sulla codifica in XML/SGML di documenti di interesse storico basata sulle DTD TEI o TEI Lite [19], che ha già ottenuto accettazione positiva dalla comunità interessata a biblioteche digitali e scienze umane. Pertanto la DTD in Fig. 2 sarà alla fine integrata nella DTD sviluppata per l'intero progetto "XML/Repetti".

[2] Lo *stile* di un calendario riguarda la data prescelta per l'inizio di un nuovo anno [3], che è il primo gennaio nello stile *moderno*. Nel medioevo diversi stili erano in uso in Toscana (es. nello stile Fiorentino l'anno iniziava il 25 marzo mentre nello stile Pisano iniziava il 25 marzo dell'anno precedente) e quindi le date presenti nel *Repetti* possono corrispondere a date differenti secondo lo stile moderno a seconda del contesto locale da cui sono tramandate; se non indicato chiaramente nel testo, l'ambiguità sul potenziale uso di diversi stili è una ulteriore fonte di indeterminazione.

3 Realizzazione di un *Tool* di Sviluppo

Scopo del nostro lavoro nel progetto “XML/Repetti” è anche il progetto e la realizzazione di un *tool* di sviluppo per la codifica assistita di documenti XML temporali. Partendo da una versione elettronica (in formato HTML) del dizionario, tale *tool* cerca di individuare, facendo uso di *espressioni regolari*, quante più determinazioni temporali possibili in esso contenute. Terminata questa prima “passata”, per ogni espressione temporale individuata, il *tool* guida l’utente nel processo di codifica della marcatura corrispondente proponendo una soluzione strettamente dipendente dal tipo dell’espressione stessa e dalla sua classificazione in una delle quattro categorie. Questo processo semiautomatico lascia comunque all’utente la libertà di scegliere se accettare la proposta avanzata dal *tool* di supporto oppure modificarla sulla base della propria interpretazione del testo. Inoltre l’utente può comunque e sempre selezionare parti del testo che ritiene debbano essere codificate perché contenenti informazione storica malgrado il *tool* non sia stato in grado di individuarle. Il sistema è utilizzabile facendo uso di un comune *browser* come Netscape o Explorer. Sfruttando quindi le capacità del client Web di interpretazione e visualizzazione di documenti HTML (e XML), esso consente all’utente di navigare all’interno del testo da codificare, che appare come il testo originale anche se già parzialmente codificato. I *tag* già inseriti non saranno quindi esplicitamente visibili come “interferenze” nel testo ma le corrispondenti parti saranno evidenziate (ad esempio visualizzate con diversi colori) al fine di poterle facilmente individuare. L’interfaccia grafica prevede finestre di dialogo per l’inserimento dei *tag*, basati eventualmente sulle proposte avanzate dal tool qualora la parte del testo in oggetto sia stata individuata dal tool stesso. La Fig. 3 mostra una versione prototipale preliminare del *tool* e, in particolare, una possibile organizzazione dell’interfaccia di interazione con l’utente. Nell’area principale della finestra (zona superiore) è visibile il documento che sta subendo il processo di marcatura. La fase di pre-elaborazione automatica ha evidenziato tutte le espressioni temporali che sono state rinvenute all’interno del documento (es. l’espressione “**nel 1404**” in figura). Per ciascuna espressione individuata, il *tool* ha aggiunto al documento due bottoni, ben visibili in figura, che permettono all’utente di eseguire due differenti azioni, a seconda che giudichi il risultato della ricerca proposto dal *tool* appropriato o meno: nel caso si tratti effettivamente di una espressione temporale da marcare potrà accedere alla funzionalità di inserimento dei *tag* tramite il bottone superiore oppure rifiutare la proposta di marcatura eliminandone ogni traccia dal documento tramite il bottone inferiore. Nel primo caso l’utente potrà accedere alle funzionalità che si attivano nella parte inferiore della finestra (vedi figura) che gli permetteranno di specificare struttura ed attributi dei *tag* da inserire. Nel caso l’utente non sia soddisfatto delle scelte proposte in automatico dal *tool*, potrà sempre operare scelte alternative usando le opzioni messe a disposizione dai *form* di inserimento dati. Una volta effettuate le scelte necessarie, l’utente potrà accedere (tramite il bottone in basso a destra in figura) alla funzionalità di *anteprima*, che consente di visionare la codifica XML

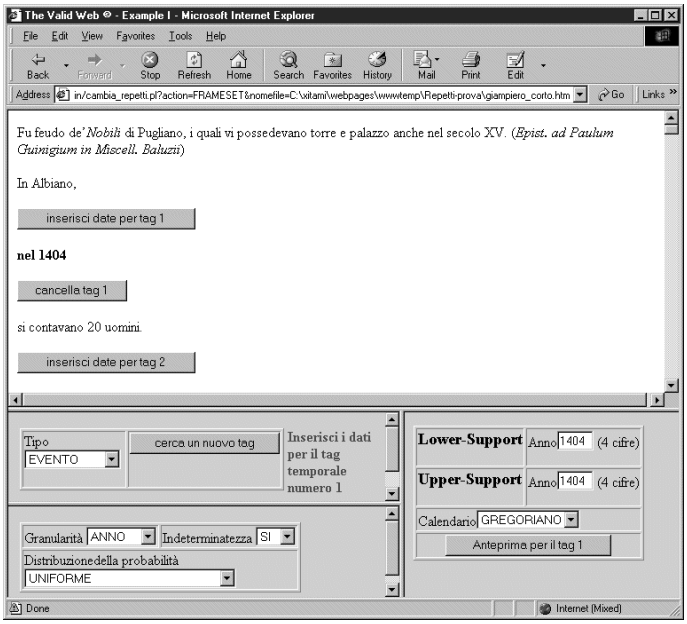

Figura 3: L’interfaccia grafica del *tool* di sviluppo. In questa prima versione del prototipo una distribuzione indeterminata richiede la specifica dei supporti mentre la versione finale permetterà di usare direttamente l’ETR.

completa del testo marcato e di rendere eventualmente definitivo il suo inserimento nel documento.

Una volta terminato l'inserimento, per comodità dell'utente il corrispondente testo può venire evidenziato dal *browser* in maniera differente, i relativi bottoni scompaiono, e si può passare al trattamento di un successivo *tag*. Nel caso che, scorrendo il testo del documento nella finestra del *browser*, l'utente identifichi una espressione temporale che non è stata riconosciuta automaticamente dal *tool*, egli potrà procedere ad una selezione manuale (tramite il *mouse*) della stessa ed alla sua successiva marcatura.

Ovviamente, le successive fasi di progetto delle funzionalità del *tool* e della sua interfaccia grafica saranno svolte in stretta collaborazione con i ricercatori storici che collaborano al progetto "XML/Repetti", che ne costituiscono gli utenti finali. Sarà invece nostro compito specifico implementare un prototipo per la fruizione del documento *Repetti* marcato, che consenta di realizzare funzionalità di ricerca avanzata delle voci in base ad un predicato temporale specificabile dall'utente, estendendo in definitiva le funzioni dell'approccio "The Valid Web" con la maggiore ricchezza semantica del *markup* qui prospettata. Sarà oggetto di particolare attenzione l'ottimizzazione degli algoritmi di confronto fra date indeterminate che sta alla base di ogni successiva elaborazione, la cui efficienza risulta particolarmente critica data la grande mole di dati da trattare (tutte le voci del *Repetti*, contenente migliaia di espressioni temporali).

4 Realizzazione di un Motore Temporale di Ricerca

Il sistema che gestisce la ricerca temporale avanzata all'interno del *Repetti* deve avere un'implementazione il più possibile ottimizzata per far fronte alle dimensioni del *Repetti* ed alla complessità delle procedure di ricerca. Questo requisito ha un diretto impatto sull'architettura generale del sistema, sull'organizzazione di memorizzazione del dizionario e sugli algoritmi per l'esecuzione delle interrogazioni. La soluzione di progetto qui prospettata, e su cui si basa il prototipo in via di sviluppo (per il momento limitato alle voci del dizionario che iniziano con la lettera "A"), si basa sull'impiego di una struttura ad indice temporale e di algoritmi di ricerca ottimizzati su un'architettura che prevede tutta l'elaborazione sul lato server.

In particolare, l'organizzazione fisica da noi prescelta prevede l'adozione di un indice temporale di tipo MAP21 [13], che si appoggia a tecnologie del tutto convenzionali (ovvero sull'uso "non standard" di un B$^+$-tree). L'indice MAP21 converte gli estremi di un intervallo temporale $I = [I^s, I^e]$ in un valore singolo (con la trasformazione lineare $\Phi(I) = 10^\alpha\, I^s + I^e$) che è poi usato come chiave di ricerca in un tradizionale B$^+$-tree. La costante α è il massimo numero di cifre necessarie a rappresentare un qualunque valore di tempo. Poiché in due millenni ci sono (485 anni bisestili) e 730.485 giorni, possiamo scegliere $\alpha = 6$ (sei cifre sono in realtà sufficienti per rappresentare circa 2.700 anni, che sembrano abbastanza per coprire tutte le date di interesse nel *Repetti*). Con tale scelta di α, la trasformazione è biiettiva (l'ordinamento lessicografico fra intervalli si mantiene nelle foglie) e gli estremi originari possono essere riestratti come $I^s = \Phi_s^{-1}(\Phi(I)) = \lfloor \Phi(I)/10^\alpha \rfloor$ e $I^e = \Phi_e^{-1}(\Phi(I)) = \Phi(I) \bmod 10^\alpha$. Nel nostro caso, le date indeterminate possono così essere indicizzate come intervalli fra i supporti (e le date determinate come intervalli di lunghezza unitaria). Dato che abbiamo a che fare con fatti storici non abbiamo il problema degli intervalli "senza limite destro", cioè con $I^e = NOW$, che richiedono un trattamento speciale nell'approccio MAP21 con l'uso di una struttura indice accessoria (Open Ended Tree). L'indice MAP21 consente di eseguire efficientemente[3] *query* di tipo *timeslice/overlap/containment*, che sono tutto quanto occorre per supportare le attività del nostro motore di ricerca. Le voci del *Repetti* codificate in XML possono essere memorizzate come *record* a lunghezza variable in un archivio sequenziale (che risulta ordinato alfabeticamente per nome della voce se queste sono memorizzate nello stesso ordine in cui compaiono nel dizionario) e indicizzate tramite l'indice MAP21 per mezzo delle espressioni temporali che contengono.

[3]Es. le prestazioni attese del MAP21 per *query* di tipo *timeslice* (ossia "trova tutti i fatti validi ad una data assegnata") sono $O(\log_B N + (A + K)/B)$, dove N è il numero complessivo di intervalli indicizzati, A è la dimensione della risposta e B è la capacità del blocco su disco. K è una costante pari nel caso peggiore a $O(\Delta^2)$, essendo Δ l'ampiezza massima degli intervalli indicizzati, ma molto più piccola nel caso medio ($K = 0$ rappresenta l'ottimo teorico).

Nell'architettura globale di sistema, il motore di ricerca lavora sul lato del Web server: nel nostro prototipo è implementato attraverso *script* PERL lanciati dal client tramite un meccanismo CGI standard. Il vero lavoro di *retrieval* è effettuato da programmi C^{++} compilati invocati dal programma PERL: un modulo C^{++} accede alle sole foglie dell'indice MAP21 che possono contenere date di interesse (ossia non cercano date che possono senz'altro essere escluse a priori in base al predicato di ricerca), estrae dalle foglie visitate l'esatta espressione delle date (indeterminate) che vengono poi passate al modulo che effettua i confronti probabilistici. Per ogni data che supera il *test* al livello di plausibilità assegnato, il programma usa il corrispondente puntatore trovato nelle foglie dell'indice per accedere su disco alle voci che soddisfano la *query*. I puntatori possono anche essere raccolti tutti e ordinati prima dell'accesso per evitare duplicati e recuperare le voci in ordine alfabetico. I risultati della ricerca vengono poi assemblati dallo *script* PERL e restituiti al client Web tramite Internet. A seconda di una qualche opzione utente, tali risultati possono essere una singola pagina XML/HTML contenente esplicitamente tutte le voci recuperate (con le espressioni temporali qualificanti evidenziate con uno stile speciale), oppure una pagina riassuntiva contente un'elenco di *link* alle voci selezionate che possono così essere scaricate individualmente dall'utente in un secondo momento. Si noti come l'indicizzazione temporale via MAP21 è un'organizzazione secondaria, che non impedisce di costruire anche altri indici (es. sui nomi, sui luoghi, ecc.). In questa maniera, usando algoritmi ad unione/intersezione di puntatori, si possono aggiungere funzionalità di ricerca multichiave senza modificare l'impianto dell'organizzazione temporale. La disponibilità di molti indici non soffre dei costi di manutenzione dato che il *Repetti*, una volta memorizzato in forma XML, non è naturalmente soggetto a modifiche.

Le funzionalità sul lato client richiedono solo l'esecuzione dei controlli per la gestione della formulazione della *query*, specifica completa del periodo storico di interesse inclusa, che può comportare la gestione di tutti i parametri di indeterminazione, granularità e calendario. In particolare, l'interazione con l'utente avverrà grazie ad un'interfaccia il più possibile amichevole, in modo da essere usata facilmente anche da utenti non esperti di strumenti informatici, che potrà essere simile a quella disegnata per "The Valid Web" (principalmente basata su un'*applet* Java2 e funzioni Javascript) [**?**]. Essendo il protocollo HTTP *stateless*, la gestione contestuale della formulazione della *query* e dei parametri temporali può sfruttare l'uso di *cookie* nel corso della sessione.

Una particolare cura è stata posta nell'implementazione ottimizzata del programma che effettua il confronto fra date indeterminate. La probabilità di precedenza fra date $\Pr[t_1 < t_2]$ definita dall'Eq. (1) è, come in TSQL2, la base per la definizione dei predicati *precedes*, *overlaps* e *contains* usati nel sistema di gestione per la formulazione delle *query*. Lavorando con date indeterminate essi possono infatti ritenersi soddisfatti quando la probabilità associata risulta maggiore o uguale della plausibilità assegnata. Le probabilità associate, dati due intervalli con estremi indeterminati I_1 e I_2 possono essere valutate con le stesse formule utilizzate per TSQL2 [16]. L'implementazione del prototipo prevede un algoritmo per la valutazione delle probabilità di precedenza ottimizzato rispetto ai costi di CPU. Sfruttando il fatto che le distribuzioni sono funzioni costanti a tratti, abbiamo mostrato in [10] come l'Eq. (1) possa essere valuata in $O(N)$ passi, dove N è il numero di intervalli base della distribuzione avente l'intervallo base di minore ampiezza. Per di più, essendo quelle elencate in Tab. I le uniche distribuzioni di interesse, N è sempre un numero molto piccolo, che non supera sette, e pertanto possiamo assumere un tempo costante e assai limitato di valutazione della (1). Il metodo presentato in [10] richiede la memorizzazione in una tabella di sistema dei valori precalcolati della funzione cumulativa associata alle varie distribuzioni; si noti che questo non rappresenta un *overhead* di memoria significativo in quanto richiede la memorizzazione di trenta valori in tutto (ogni distribuzione contribuisce per il suo N), indipendentemente dal numero effettivo di date indeterminate da rappresentare. Da rilevare il fatto che il metodo di Dyreson e Snodgrass [5] richiederebbe invece un tempo $O(P \log_2 P)$ per la valutazione nel caso peggiore della probabilità di precedenza, essendo P il numero di granuli base ("rod") usati per discretizzare le distribuzioni di probabilità da rappresentare. Come valutato in [10], rappresentare e gestire le nostre distribuzioni con sufficiente approssimazione richiederebbe $P = 1024$ e uno spazio di memoria variabile fra 6,5 e 13 KByte *per ogni singola data indeterminata presente nell'intero documento*,

quindi un *overhead* paragonabile all'intera dimensione del documento stesso, se non addirittura superiore.

In aggiunta a questo, l'implementazione dei nostri algoritmi sfrutta anche un'ulteriore ottimizzazione *intra-query*, giovandosi del fatto che molti confronti possono via via essere evitati in base ai risultati dei confronti già effettuati. Per esemplificare tale ottimizzazione, che viene gestita dallo *script* PERL che sovrintende l'esecuzione dell'interrogazione, consideriamo la seguente ricerca: *Trova nel* Repetti *tutte le voci che contengono date posteriori a t_Q (al livello di plausibilità p)*. Nel caso più generale anche l'*input* è una data indeterminata $t_Q = (t_Q^- \sim t_Q^+, P_Q)$ (es. precedentemente estratta da un'altra voce del *Repetti*, o specificata dall'utente tramite un'opportuna griglia di inserimento). Per rispondere alla *query*, il motore di ricerca deve scandire tutte le voci per verificare se una qualche data t_D in esse contenuta soddisfi il predicato *Before*(p, t_Q, t_D). Nella realtà la scansione è eseguita sulle foglie dell'indice MAP21 piuttosto che sulle voci memorizzate: se Δ è la massima lunghezza degli intervalli nell'indice, le date da ricercare devono avere un supporto che si sovrappone all'intervallo $[t_Q^- - \Delta, t_Q^-]$ o che lo segue; la foglia che contiene $\Phi(t_Q^- - \Delta, t_Q^-)$ è quindi acceduta assieme a tutte le foglie che la seguono (recuperate seguendo i puntatori fra foglie del B^+-tree). Per ogni *entry* presente nelle foglie vengono estratte le informazioni che riguardano la data codificata t_D (ossia supporti e distribuzione) che deve essere confrontata con t_Q. L'ulteriore ottimizzazione si basa sul mantenimento da parte dello *script* PERL, per ogni distribuzione P, di due *bound*:

- Upper bound $UB(P)$: definito come il massimo supporto superiore noto t_D^+ di una data t_D con distribuzione P che *sicuramente* segue la data t_Q (al livello di plausibilità p);
- Lower bound $LB(P)$: definito come il minimo supporto inferiore noto t_D^- di una data t_D con distribuzione P che *sicuramente* ***non*** segue la data t_Q (al livello di plausibilità p);

I due *bound* vengono inizializzati come $UB(P) = t_Q^-$, $LB(P) = t_Q^+$ prima dell'esecuzione della *query*. Durante la scansione dell'indice, il trattamento della data $t_D = (t_D^- \sim t_D^+, P_D)$ è effettuato come segue. Se $t_D^+ \leq UB(P_D)$ o $t_D^- \geq LB(P_D)$ non c'è bisogno di invocare il programma C^{++} che calcola la probabilità di precedenza: nel primo caso siamo certi che *Before*(p, t_Q, t_D) è vero e la data t_D soddisfa la *query*, nel secondo siamo certi che *Before*(p, t_Q, t_D) non vale e pertanto possiamo scartare t_D. Se nessuna delle precedenti condizioni è vera, il programma C^{++} viene richiamato per calcolare $\wp = \Pr[t_Q < t_D]$; se $\wp \geq p/100$, la data t_D soddisfa la *query* e l'*upper bound* viene aggiornato al nuovo valore $UB(P_D) = t_D^+$, altrimenti la data t_D può essere tranquillamente scartata e il *lower bound* aggiornato al nuovo valore $LB(P_D) = t_D^-$. In questo modo possiamo risparmiare molti confronti perfettamente inutili (basandoci sui risultati dei confronti precedenti con date aventi la stessa distribuzione). Ogni volta che invece un confronto deve essere effettuato, il risultato viene anche usato per rendere più stringenti i *bound* rendendo via via più stretto l'intervallo di date per cui la valutazione della probabilità di precedenza è da eseguirsi.

5 Conclusioni e Ringraziamenti

In questo lavoro abbiamo presentato un approccio per la codifica in XML e la gestione di espressioni temporali all'interno di fonti storiche testuali in forma digitale. Le tecniche di codifica proposte, basate su di un approccio probabilistico all'indeterminazione e sull'uso di distribuzioni costanti a tratti, sono anche state implementate su di un prototipo sperimentale nell'ambito del progetto "XML/Repetti". In particolare, per realizzare un motore di ricerca temporale avanzato che risultasse particolarmente efficiente sono state usate e/o messe a punto soluzioni implementative particolari che sono state descritte a grandi linee. È stato pure brevemente descritto il disegno di un *tool* di marcatura assistita dei documenti, che è in corso di realizzazione, e che verrà utilizzato dai ricercatori storici nella fase di produzione della versione XML del dizionario.

Vogliamo infine ringraziare l'Ing. Emanuele Luchetti e l'Ing. Alberto Olivieri per il loro contributo al progetto ed alla implementazione dei prototipi.

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, CA, 1999.

[2] A. Benvenuti, F. Niccolucci, S. Baragli, C. Carpini, "Advances in XML Treatment of Historical Documents", in *La Historia en una Nueva Frontera*, AHC, Toledo, Spagna, 2000.

[3] A. Cappelli, "Cronologia, Cronografia e Calendario Perpetuo", Hoepli, Milano, Italia, 1998.

[4] S. Dutta, "Generalized Events in Temporal Databases", Atti *Intl. Conf. on Data Engineering (ICDE)*, Los Angeles, CA, Febbraio 1989.

[5] C.E. Dyreson, R.T. Snodgrass, "Supporting Valid-time Indeterminacy", *ACM Trans. on Database Systems*, Vol. 23, No. 1, 1998.

[6] O. Etzion, S. Jajodia and S. Sripada (a cura di), *Temporal Databases - Research and Practice*, LNCS N. 1399, Springer-Verlag, Berlino, 1998.

[7] F. Grandi, F. Mandreoli, "The Valid Web ©", Atti *Software Demonstrations Track at the EDBT 2000 Intl. Conf.*, Costanza, Germania, Marzo 2000.

[8] F. Grandi, F. Mandreoli, "The Valid Web: an XML/XSL Infrastructure for Temporal Management of Web Documents", Atti *Intl. Conf. on Advances in Information Systems (ADVIS 2000)*, Izmir, Turchia, 2000, LNCS N. 1909, Springer-Verlag, Berlino, 2000.

[9] F. Grandi, F. Mandreoli, "The "XML/Repetti" Project: XML Encoding and Manipulation of Temporal Information in Historical Text Sources", Atti *Intl. Cultural Heritage Informatics Meeting* (ICHIM'01), Milano, Italia, 2001 *(in corso di stampa)*.

[10] F. Grandi, F. Mandreoli, "Effective Representation and Efficient Management of Indeterminate Dates", Atti *Intl. Symposium on Temporal Representation and Reasoning* (TIME'01), Cividale del Friuli, Italia, 2001 *(in corso di stampa)*.

[11] F. Grandi, M. R. Scalas, "Extending Temporal Database Concepts to the World Wide Web", Atti *Convegno Nazionale su Sistemi Evoluti per Basi di Dati* (SEBD'98), Ancona, Italia, 1998.

[12] S. Hermon, F. Niccolucci, "The Impact of Web-shared Knowledge on Archaeological Scientific Research", Atti *Intl. Conf. on Current Research on Information Systems (CRIS'00)*, Heksinky, Finland, 2000.

[13] M. Nascimento, M.H. Durham, "Indexing Valid Time Databases via B^+-trees", *IEEE Trans. on Knowledge and Data Engineering*, Vol. 11, No. 8, 1999.

[14] F. Niccolucci, A. Zorzi, M. Baldi, F. Carminati, P. Salvatori, T. Zoppi, "Historical Text Encoding: an Experiment with XML on Repetti's Historical Dictionary", Atti *Conf. of the Association for History and Computing - UK Branch (AHC-UK'99)*, Londra, UK, 1999.

[15] Prolog and Document Type Declaration, in *Extensible Markup Language (XML) 1.0*, W3C Recommendation, `http://www.w3.org/TR/REC-xml#sec-prolog-dtd`.

[16] R.T. Snodgrass (a cura di), 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, MA, 1995.

[17] A. Tansel, J. Clifford, V. Gadia, S. Jajodia, A. Segev, R.T. Snodgrass (a cura di), *Temporal Databases: Theory, Design and Implementation*, Benjamin/Cummings, Redwood City, CA, 1993.

[18] The eXtensible Markup Language (XML) Resource Page, W3C Consortium, `http://www.w3.org/XML/`.

[19] Text Encoding Initiative, TEI Consortium Home Page, `http://www.tei-c.org`.

[20] The Semantic Web Agreement Group Home Page, `http://swag.semanticweb.org`.