6

STRUMENTI

XML per i beni culturali

Esperienze e prospettive per il trattamento di dati strutturati e semistrutturati

a cura di
Sonia Maffei

EDIZIONI
DELLA
NORMALE

ISBN 978-88-7642-221-8

Indice

Introduzione

Sonia Maffei

Dal 1998, quando XML è nato sviluppandosi da una costola di SGML, la sua duttilità si è imposta all'attenzione del pubblico internazionale con un inno che potremmo chiamare 'futurista' alla velocità: SPEED, la parola che riepiloga i vantaggi di XML, è infatti acronimo di Storing (memorizzazione); Publishing (pubblicazione); Exchanging Electronic Documents (scambio di documenti elettronici). Questi tre settori, l'archiviazione, la pubblicazione e l'interscambio rappresentano infatti i campi in cui l'XML ha espresso tutta la sua efficacia[1].

Desidero ringraziare i partecipanti alla giornata e quanti hanno contribuito alla ricca discussione degli interventi. Un ringraziamento particolare per la stesura degli atti va a Maria Vittoria Benelli, Bruna Parra e Antonella Pascucci. Dedico questo lavoro a Michele Ciliberto a cui devo molte cose, tra cui una calda adesione alla mia idea di questa giornata di studio e un incoraggiamento alla pubblicazione di questi atti.

[1] XML è un linguaggio definito dal World Wide WEB Consortium <http://www.w3.org/XML>; *The XML 1.0 specification* <http://www.w3.org/TR/REC-xml>; *The XML 1.1 specification* <http://www.w3.org/TR/xml11>; *Annotated XML Specification* <http://www.xml.com/axml/testaxml.htm>; *The XML FAQ Frequently-Asked Questions about the Extensible Markup Language* <http://xml.silmaril.ie>; una traduzione in italiano della Raccomandazione del W3C *Extensible Markup Language (XML) 1.0* di Andrea Marchetti si può consultare all'indirizzo: http://www.xml.it:23456/XML/REC-xml-19980210-it.html. Cfr. inoltre *XML. From Wikipedia, the free encyclopedia* <http://en.wikipedia.org/wiki/XML>; *Comparison of SGML and XML* <http://www.w3.org/TR/NOTE-sgml-xml-971215>; *CIMI Briefing Paper: Introducing XML* <http://www.cimi.org/public_docs/xmlbrief.html>. Utili sintesi sul problema si trovano anche in Signore 2001; Signore 2002c; Burnard, Sperberg-McQueen 1995.

Quanto all'interscambio di dati, la scelta di XML come metalinguaggio è stata promossa nell'ambito dei beni culturali da importanti istituzioni, come il CIMI[2] (Consortium of the Interchange of Museum Information) e l'MDA[3] (Museum Documentation Association), ed è stata attuata in moltissimi progetti sia museali che storico-culturali: le caratteristiche di XML lo identificano come il mezzo ideale per far dialogare banche dati diverse e permettere l'interscambio tra dati gestiti con software e con schemi catalografici differenti. In questo senso XML può costituire un valido aiuto contro l'obsolescenza dei software e per la conservazione degli archivi digitali[4]. Un progetto campione in questo campo è stato XML-Spectrum Schema[5] promosso dal CIMI e dall'MDA per sperimentare appunto la possibilità di interscambio tra i dati. Il progetto propone una versione XML del modello anglossassone per la gestione museale SPECTRUM[6], interessante standard che permette di definire e implementare le procedure indispensabili alla gestione dei beni conservati nei musei nel loro intero ciclo di vita, dall'annessione nel museo alla loro uscita, e consente quindi anche di catalogare oggetti molto diversi tra loro, abbracciando più settori, dall'archeologia alla biologia, dalla paleontologia alla pittura. Più complesso il progetto COVAX (Contemporary Culture Virtual Archives in XML)[7] proposto da diversi partners tra cui l'ENEA per promuovere l'interscambio tra biblioteche (utilizzando lo standard MARC), archivi (EAD), musei (AMICO) e banche dati testuali (TEI). XML è utilizzato come lo strumento principe per consentire la standardizzazione, l'interoperabilità e l'interconnettività fra biblioteche, archivi e musei durante ogni fase di interrogazione, ricerca e reperimento di tutti i tipi di documenti, resi accessibili come fonti primarie su Internet per tutti gli utenti.

---

[2] <http://www.cimi.org/>.

[3] <http://www.mda.org.uk/>.

[4] Cfr. Tola, Castellani 2004.

[5] <http://www.cimi.org.uk/spectrum.htm>.

[6] <http://www.mda.org.uk/spectrum.htm>.

[7] <http://www.covax.org/>; *Project Deliverable Report, D1.2 – State of the Art. COVAX* <http://www.covax.org/public_docum/pdf.zip>; Project Deliverable Report, D8.1 – Market Studies. COVAX <http://www.covax.org/public_docum/pdfM.zip>.

L'interesse di XML relativamente alla pubblicazione dei dati è dovuto al fatto che XML si preoccupa precipuamente della definizione dei tag da usare e della loro strutturazione, separando dunque la struttura e il contenuto dalla presentazione. Di conseguenza in XML uno stesso documento può essere mostrato in modi diversi senza che esso venga alterato: può essere divulgato via cellulare o via sms, mostrato attraverso un monitor ma anche diffuso attraverso una voce generata sinteticamente. Questo significa che un documento XML può essere letto e visualizzato secondo un progetto non necessariamente preso in considerazione all'atto della sua creazione. La duttilità di XML in fase di pubblicazione viene già ampiamente utilizzata in applicazioni concrete: prova è che il linguaggio XSL[8] per la trasformazione dei documenti XML in vari formati (HTML per il web, PDF per la stampa, ecc.) è oggi uno standard affermato.

La duttilità di XML nei modi di pubblicazione trova riscontro anche nella sua espressività nell'archiviazione dei dati. XML è ampiamente sperimentato nel settore delle banche dati testuali, grazie anche ad un'iniziativa internazionale chiamata TEI (Text Encoding Initiative[9]), linguaggio di codifica che offre agli studiosi una vasta gamma di strumenti per la descrizione e rappresentazione di fenomeni linguistici[10]. Accanto a queste metodologie si sta affermando anche il nuovo campo dei database XML che, pur essendo più giovane rispetto alle metodologie di basi di dati relazionali, sta dando dei segni di maturità significativi (lo standard XQuery[11] e la presenza di software come Exist, Berkeley DB XML, Xindice-Apache), che riescono a superare i limiti di espressività dei database tradizionali.

In questo panorama ricco di fermenti e di sperimentazioni ci è sembrato utile proporre un momento di riflessione sugli usi di XML nel settore dei beni culturali proponendo un confronto tra esperienze significative di diversi settori dall'archeologia alla catalogazione e

[8] Cfr. Grosso, Walsh 2000; cfr. inoltre <http://xml.coverpages.org/xsl.html>.

[9] Cfr. <http://www.tei-c.org/>.

[10] Cfr. <http://www.tei-c.org/Guidelines2/index.html>; <http://crilet.scu.uniroma1.it/ricerca/SGML-XML/teiu5-it/teiu5-it.html>; Burnard, Sperberg-McQueen 1995.

[11] Cfr. Braga *et al.* 2003.

gestione museale, dagli archivi alle biblioteche, secondo interessi e percorsi sperimentati da tempo nei progetti della Scuola Normale.

Da queste aspettative è nata l'idea della giornata di studio *XML per i beni culturali. Esperienze e prospettive per il trattamento di dati strutturati e semistrutturati*, tenutasi a Pisa il 25 marzo 2004, promossa dal Centro di ricerche informatiche per i beni culturali della Scuola Normale[12] diretto da Michele Ciliberto. Gli atti della giornata di studio pubblicati in questo volume rappresentano un risultato interessante per ricchezza di approcci e ampiezza di metodi da cui ci auguriamo possano scaturire indicazioni, sollecitazioni e prospettive nuove per nuovi sperimentatori di XML nel settore dei beni culturali.

[12] Il Centro, rinnovato nelle sue finalità, metodologie e procedure, dal 2005 si è trasformato in Signum. Centro di ricerche informatiche per le discipline umanistiche, <http://www.signum.sns.it>.

Esperienze di codifica elettronica di documenti archeologici

1. *Esperienze e prospettive*

Il Progetto Caere[1], promosso e sviluppato nell'ambito delle attività del Progetto finalizzato beni culturali del CNR, ha consentito di adottare e mettere a punto una metodologia di indagine informatizzata, mirata al recupero e alla sistematizzazione dei dati contenuti all'interno di documenti archeologici non strutturati relativi a indagini condotte sul campo.

Se il punto di partenza archeologico del progetto è stata la volontà di recuperare, georeferenziare e rappresentare in formato elettronico la documentazione prodotta durante le sette campagne di scavo condotte sul pianoro urbano di Cerveteri, sotto la direzione di Mauro Cristofani, dall'Istituto per l'archeologia etrusco-italica – oggi Istituto di studi sulle civiltà italiche e del Mediterraneo antico (ISCIMA) – l'impulso metodologico è senz'altro venuto dal confronto pluriennale con le esperienze sviluppate nel settore dell'Informatica umanistica dal gruppo di ricerca diretto da Tito Orlandi e indirizzate in particolare verso le problematiche di codifica, trattamento automatico e trasmissione di dati testuali[2].

Come si è avuto più volte occasione di sottolineare[3], l'adozione di linguaggi di marcatura – SGML prima e XML poi – non solo si è dimostrata una scelta assai appropriata per la formalizzazione, l'elaborazione automatica e il recupero delle informazioni contenute nei testi dei diari di scavo relativi all'area della Vigna Parrocchiale a Cerveteri, ma soprattutto ha aperto la via a nuove sperimentazioni su tipologie

[1] <http://www.progettocaere.rm.cnr.it>, con indicazioni bibliografiche complete nella pagina relativa.

[2] Cfr., tra i primi contributi, Gigliozzi 1987 e, in particolare, Adamo 1987.

[3] Cfr. da ultimo Moscati 2003 e c.s.

di documenti diversi (fig. 1). Si è così passati, anche attraverso l'adozione della TEILite come modello standard di marcatura testuale, alla codifica di rapporti di scavo editi degli inizi del Novecento, relativi a interventi condotti da Raniero Mengarelli sempre a Cerveteri[4], e all'elaborazione elettronica di documenti di archivio dell'Ottocento, contenenti itinerari archeologici nel territorio circostante Roma.

In quest'ultimo caso, l'applicazione di XML si è per il momento concentrata sulla codifica del manoscritto di Ercole Nardi, ultimato nel 1885 e conservato presso la Biblioteca di Archeologia e Storia dell'Arte di Palazzo Venezia in Roma, dal titolo *Ruderi delle Ville Romano-Sabine nei dintorni di Poggio Mirteto*[5]. Nel manoscritto è presa in esame sotto forma di itinerario archeologico, secondo una tradizione di ricerca in voga a quel tempo, una parte del territorio della Sabina tiberina, che rientra nell'area più ampia dove è attualmente in corso un progetto di ricognizione di superficie denominato Progetto Galantina[6]. Per la codifica elettronica del manoscritto del Nardi si è scelto di effettuare un ulteriore passo in avanti per descrivere la risorsa a disposizione, inserendo i metadati che costituiscono l'insieme Dublin Core, espressi in RDF, all'interno del documento marcato TEILite, in cui ovviamente sono stati anche aggiunti gli elementi della DTD già da noi in precedenza definiti per la marcatura di documenti di carattere archeologico.

Se gli aspetti più specificamente tecnici saranno illustrati nella seconda parte di questo contributo, giova qui ricordare che la complessità della procedura adottata è strettamente connessa alla volontà di integrare i dati testuali all'interno di una piattaforma GIS, in vista di inserire e interpretare le testimonianze archeologiche nel loro contesto geografico, e di diffondere le informazioni in rete facilitandone la consultazione e la diffusione[7]. La necessità di rendere il testo narrativo fruibile da applicazioni diverse (software cartografici e database) è d'altronde legata sia al recupero di dati pregressi, già sottoposti ad informatizzazione nel corso delle campagne di scavo degli anni Ottanta[8], sia all'integrazione con Sistemi Informativi Territoriali elaborati per l'area urbana di Cerveteri e per la Sabina tiberina.

[4] Mariotti 2001.

[5] Barchesi *et al.* 2003; Scarpati c.s.

[6] Guidi, Santoro 2003; Santoro *et al.* c.s.

[7] Barchesi 2001; Ceccarelli 2001.

[8] Moscati 1991.

La naturale evoluzione degli studi compiuti e del modello di trattamento informatizzato dei dati archeologici messo a punto si rivolge oggi verso l'integrazione tra l'analisi topografica di specifiche aree territoriali e di centri antichi a continuità di vita e la documentazione prodotta per la Carta Archeologica d'Italia nel corso delle indagini condotte nell'Etruria tiberina e nell'Agro falisco[9]. In particolare, in vista dell'informatizzazione e della georeferenziazione del materiale documentario che Lucos Cozza sta donando alla Biblioteca Massimo Pallottino, che ha sede presso l'ISCIMA, si sta procedendo alla digitalizzazione della cartografia dell'antico centro di *Falerii Veteres*, situato sul pianoro tufaceo oggi occupato da Civita Castellana (VT) e già oggetto negli anni Ottanta di una intensiva campagna di ricognizioni topografiche[10].

Tale cartografia è finalizzata da un lato alla visualizzazione, anche attraverso la realizzazione di modelli tridimensionali del terreno, dell'antico assetto territoriale, così influenzato dalla particolare morfologia del suolo, caratterizzata dall'alternarsi di pianori tufacei e di profonde gole scavate dai corsi d'acqua. Dall'altro lato, essa costituirà la base per georeferenziare e integrare, all'interno di una piattaforma GIS, dati di natura diversa provenienti da indagini effettuate in tempi successivi nel territorio e concernenti anche le necropoli e i santuari relativi all'antica capitale del popolo falisco: strutture archeologiche, testimonianze grafiche e fotografiche, informazioni relative ai reperti mobili, ma soprattutto – sulla scia dell'esperienza ceretana – dati testuali provenienti da documenti d'archivio.

In vista di questa nuova iniziativa, ci si trova oggi di fronte al problema della definizione di una ontologia formale che, seguendo gli ultimi sviluppi del dibattito sul *Semantic Web*[11] e sulla scia di iniziative quali quelle promosse nell'ambito del CIDOC Conceptual Reference Model (CRM)[12], consenta da un lato di rappresentare all'interno di una struttura logica la documentazione d'archivio relativa al centro di *Falerii Veteres*, e dall'altro di costruire un modello standard di riferimento per la formalizzazione e il recupero di dati testuali non strutturati, esito di ricerche archeologiche effettuate in epoche diverse

[9] Gamurrini *et al.* 1972; Cozza, Pasqui 1981; Castagnoli 1978.

[10] Cfr. in sintesi Moscati 1990.

[11] <http://www.semanticweb.org/>. Cfr. anche Berners-Lee 1999.

[12] <http://cidoc.ics.forth.gr/>.

nel territorio circostante Roma. E questo problema non risiede tanto nella scelta di un linguaggio – o metalinguaggio – di modellazione (XML, RDF, OWL, ecc.), quanto piuttosto nella definizione di una metodologia di indagine per estrarre e strutturare logicamente la conoscenza che deve essere rappresentata in formato digitale[13].

Nelle esperienze da noi fino ad oggi realizzate, sia che si trattasse di diari di scavo, di rapporti archeologici editi o di itinerari archeologici, l'uso di standard di marcatura e di metadati si è orientato soprattutto verso la descrizione formale delle risorse a disposizione, ma la dichiarazione della struttura del testo (DTD), per quanto attiene specificamente al contenuto archeologico, è stata il frutto dell'analisi dei singoli testi, manoscritti o a stampa, e non quello di un formalismo definito *a priori*. Infatti, il sistema di codifica messo a punto per i diari di scavo, pur rimanendo – data la sua funzionalità – come punto di riferimento essenziale per le esperienze successive, è stato di volta in volta modificato e integrato in base alla struttura e ai contenuti archeologici dei testi presi in esame. D'altronde, come sostiene Tito Orlandi, manifestando tra l'altro serie perplessità verso l'applicazione dello stesso standard TEI, il sistema della codifica è il risultato dello studio dei singoli testi e non può essere ad esso preliminare[14].

La strategia che ha fin qui guidato le nostre scelte ha portato con sé una ricchezza di problemi e di riflessioni teoriche, proprio grazie all'introduzione nei testi analizzati di una sovrastruttura insieme descrittiva e interpretativa, non limitata a un livello solo formale ma tesa a un livello concettuale, che prevede una continua sovrapposizione tra la fase descrittiva delle risorse e quella interpretativa. Ci si è infatti rivolti a una lettura attiva dei testi che, dopo l'inserimento di 'metacategorie', ha permesso, soprattutto nel caso dei diari di scavo, di far emergere gli esiti delle diverse fasi di lavoro degli archeologi e di giungere a una fase ulteriore di riflessione sulla realtà storica che si andava analizzando.

Procedendo nella nostra esperienza di codifica elettronica di testi archeologici e aumentando di conseguenza la documentazione da analizzare e la possibilità di gestire grandi quantità di materiali, il sistema deve sicuramente divenire più complesso, ma non può e non deve – proprio dopo gli esiti raggiunti – tornare verso un livello di

[13] Ross 2003.

[14] Orlandi 1999, 100.

astrazione descrittiva delle risorse, che richiederebbe tra l'altro una definizione *a priori*. Questo ci porterebbe a dover nuovamente affrontare quel genere di remore e di problemi, soprattutto di carattere terminologico, che ci avevano indotto alla metà degli anni Novanta a non scegliere esclusivamente la soluzione di un database per l'informatizzazione della documentazione di scavo.

La soluzione potrebbe dunque essere quella di prevedere una scelta duplice, che per un verso continui a indirizzarsi all'uso di standard internazionali per la descrizione delle risorse e si adegui alle nuove prospettive del *Semantic Web* per potenziare la trasmissione in rete delle informazioni. Tenendo anche conto della stesura della nuova scheda Sito proposta dall'ICCD nell'ambito del Sistema Informativo Generale del Catalogo – SIGEC[15], la definizione di un livello ontologico di descrizione dei dati, che si collochi ad un gradino superiore di quello degli stessi metadati[16], dovrebbe consentire di formalizzare le gerarchie e le relazioni logiche fra le informazioni archeologiche provenienti dalle ricerche sul terreno.

La rivalutazione dei contenuti dovrà però proseguire anche nel segno di una marcatura che, salvaguardando l'integrità dei testi, consenta di ampliare la valenza interpretativa dei dati. E ciò sarà possibile solo superando un'acquisizione e una trasmissione elettroniche passive, per offrire una lettura dinamica e non astratta dei documenti, in quanto necessariamente inserita nel contesto storico-geografico di riferimento e modellata in base alle modalità di ricerca adottate dai singoli archeologi che hanno operato sul campo in epoche diverse.

Paola Moscati

2. *Tecnologie per il trattamento dell'informazione archeologica testuale*

Nell'ambito delle attività di ricerca del settore informatico dell'ISCIMA, coordinato da Paola Moscati, la sperimentazione dei linguaggi di marcatura vanta ormai una tradizione quasi decennale. L'attività è iniziata infatti nella seconda metà degli anni Novanta,

[15] Cfr. da ultimo Mancinelli 2004.

[16] Signore 2002c.

con l'applicazione di SGML[17] per la marcatura dei testi dei diari di scavo della Vigna Parrocchiale di Cerveteri. Con l'avvento di XML[18], la versione elettronica è stata adattata alle regole di questo standard; il passaggio è avvenuto senza problemi[19], e con facilità è stato possibile realizzare un modello integrato XML-XSL per la pubblicazione in Internet dei testi codificati[20]. Lo sviluppo di un'applicazione di *information retrieval*, sviluppata in ASP (Active Server Pages), ha reso poi possibile interrogare i diari in modo evoluto direttamente sul web, offrendo la possibilità di sperimentare nuovi modelli di fruizione dell'informazione archeologica.

Per la marcatura dei diari di scavo della Vigna Parrocchiale è stata realizzata una DTD adeguata, che eredita la struttura logica direttamente dai testi originali. I diari sono per loro stessa natura organizzati cronologicamente, tuttavia recano all'interno anche importanti notazioni topografiche. L'elemento più significativo nella struttura gerarchica della DTD utilizzata per i diari di scavo è identificato dalla coppia di tag <GIORNATA><\GIORNATA>, che delimita l'inizio e la fine di un blocco d'informazioni redatte nel corso dello stesso giorno[21]. Ogni giornata di scavo, nella sua specificità, è legata a una struttura formale che è del tutto identica per tutte le giornate registrate.

Sulla base di questa impostazione sistematica è stato possibile mettere a punto un programma d'interrogazione capace di recuperare sia semplici occorrenze testuali sia specifici elementi archeologici (strutture, strati, reperti, ecc.) presenti in un determinato contesto topografico, considerato come discriminante in fase di ricerca. Per realizzare un sistema d'interrogazione dinamico dei diari si sono adottate tecnologie ASP e VBscript (Visual Basic Scripting Edition), mediante le quali l'utente può accedere alla pagina di consultazione con qualsiasi browser Internet. L'URL della pagina d'interrogazione dei diari di scavo ceretani è www.progettocaere.rm.cnr.it/ApplicazioneInternet.htm, da cui è possibile accedere a un sottoinsieme di dati dimostrativo dell'intero *corpus* documentale, limitato ai diari dell'anno 1983 e all'area definita «Cisterna».

---

[17] <http://www.w3.org/MarkUp/SGML/>.

[18] <http://www.w3.org/XML/>.

[19] Boncontro 2001.

[20] A titolo esemplificativo cfr. <http://www.progettocaere.rm.cnr.it/XML/1983fxml.xml>.

[21] Moscati, Mariotti, Limata 1999, 170-173.

Per effettuare le ricerche all'interno del testo è stata utilizzata una procedura VBscript basata sulle *regular expressions*, che permettono di creare *pattern* di ricerca testuale complessi. Questa procedura acquisisce i criteri di selezione impostati dall'utente, estrae i risultati dalla collezione di documenti e invia al browser il sommario di ogni singolo risultato corredato da un link ipertestuale alla relativa pagina del diario.

La facile integrabilità delle tecnologie Microsoft ha permesso di risolvere anche alcuni problemi relativi alla semantica archeologica. La terminologia impiegata dai redattori dei diari, durante i sette anni di campagne di scavo, non è infatti omogenea. Gli archeologi hanno chiamato alcune aree e strutture con nomi diversi, cambiandone la definizione a mano a mano che l'interpretazione dello scavo prendeva forma. Per perfezionare il programma di ricerca testuale si è pensato quindi di realizzare un semplice database relazionale da associare ai documenti marcati, che contenesse in apposite tabelle la coppia significato-significante di alcuni elementi topografici per i quali era stata adottata nel testo una terminologia variabile.

Consultando questo database, l'applicazione sviluppata – dopo che l'utente ha operato alcune preselezioni di contesto – popola le liste di opzioni per la ricerca solo con valori pertinenti ai dati topografici e cronologici selezionati, e considera per essi una serie di sinonimi testuali, che vengono aggiunti al *pattern* della *regular expression*. Il database è costituito da più tabelle separate, legate da opportune relazioni, ed è realizzato con Access e collegato ad ASP tramite una stringa di connessione ed il supporto ODBC e ADO. Per fare un esempio, cercando nei diari di scavo della Vigna Parrocchiale i riferimenti all'area definita «Cisterna», si trovano anche quelli relativi all'area via via definita, nel corso delle operazioni di scavo, «scasso rettangolare», «cava» e «vasca», che si riferiscono alla medesima imponente cavità scavata nel banco tufaceo.

I risultati incoraggianti del Progetto Caere hanno prodotto stimolo e supporto per lo sviluppo di ulteriori sperimentazioni: in particolare, per l'estensione del modello realizzato a classi eterogenee di documenti archeologici testuali non strutturati. Per marcare documenti archeologici relativi a rapporti di scavo già editi ci si è indirizzati verso l'adozione di un modello di marcatura testuale standard, la TEILite[22], che è stato poi adeguatamente esteso. In un articolo di Simona

[22] Burnard, Sperberg-McQueen 1995.

Mariotti[23] si proponeva un primo modello di marcatura per testi archeologici di carattere storico; il modello affrontava l'integrazione della TEILite SGML con ulteriori elementi, adeguati a caratterizzare semanticamente, nel contesto della disciplina archeologica, elementi del testo il cui valore era ritenuto significativo.

Questa medesima procedura, rivista e integrata, è stata poi applicata al manoscritto di Ercole Nardi *Ruderi delle Ville Romano-Sabine nei dintorni di Poggio Mirteto*, redatto sul finire del XIX secolo. Testi come quelli del Nardi, come è d'altronde caratteristico dell'epoca, non seguono una struttura espositiva che permetta la creazione di una gerarchia di elementi di marcatura definita. La descrizione del Nardi, infatti, segue l'itinerario compiuto dall'autore, che prende in esame prima le cose 'grandi' (contenitore) e poi le singole parti costituenti (contenuto) che incontra nel percorso.

Come è noto, la DTD TEI è molto complessa – la TEI integrale consta di 1300 pagine di specifiche – e pone notevoli difficoltà all'utente che miri a integrare al suo interno nuovi elementi. Esiste un protocollo ufficiale per la customizzazione del modello, ma è molto complesso[24]. Da parte nostra, la customizzazione della versione P4 (XML), allo scopo di inserire un'adeguata semantica archeologica, è stata affrontata con una tecnica più semplice, ma che si è rivelata comunque efficace. La DTD TEILite è stata modificata utilizzando il meccanismo delle 'entità parametriche'. Sono state definite due entità parametriche: %ARCHEO_NEW e %TEI_OLD. In %ARCHEO_NEW sono state inserite le dichiarazioni degli elementi TEI modificati, quelle degli elementi aggiunti della semantica archeologica e le integrazioni degli attributi relativi.

In %TEI_OLD si sono spostati gli elementi originali TEI, non modificati. Utilizzando le *keywords* INCLUDE e IGNORE nella DTD interna di un documento si può decidere quale entità parametrica usare. Questo, oltre a conservare la DTD TEILite perfettamente funzionante per altri documenti standard, permette di racchiudere le modifiche apportate all'interno di un contenitore (l'entità parametrica) facilmente identificabile: ben diverso sarebbe stato infatti appor-

[23] Mariotti 2001.

[24] Text Encoding Initiative, *The XML Version of the TEI Guidelines*, Chap.29 «Modifying and Customizing the TEI DTD» (<http://www.tei-c.org/P4X/MD.html>). Per chi volesse cimentarsi, un buon supporto è Phillips 2000.

tare le modifiche necessarie nella lunghissima e complessa DTD della TEI 'a macchia di leopardo', rendendole di fatto difficilmente risarcibili. Nel subset DTD interno dei nostri documenti, %TEI_OLD viene ignorata e %ARCHEO_NEW inclusa. Grazie a questa 'inversione' dinamica il *parser* di XMLspy (l'editor XML da noi usato) valida il documento marcato con gli elementi della nuova semantica e assicura che esso sia *well-formed* secondo lo standard XML. Questo semplice meccanismo di *swapping*, realizzato attraverso le entità parametriche, ci ha dunque permesso di modificare la TEILite senza usare il complicato protocollo standard[25].

Le integrazioni apportate al modello TEILite riguardano gli elementi <p> (paragraph), <item> (elemento lista) e <note>. I nuovi marcatori introdotti sono: Toponimo, Idronimo, Posizione, Percorso, Panorama, Evidenza, Quota, Vicinanza, Riftavola, Area, Struttura, Substruttura, Unità, Azione, Reperto. Ogni elemento dispone di attributi per la georeferenziazione e di ID numerici per la correlazione con gli altri elementi archeologici.

Nello schema di marcatura da noi ideato, l'archeologo che cura l'edizione dei testi può procedere assegnando alle strutture descritte coppie attributo-valore che trasformano in senso relazionale il testo narrato, legando le parti descritte in insiemi coerenti per fase e funzione all'interno di una gerarchia semplificata ma efficace del tipo: area-struttura-substruttura. Gli elementi che sono in relazione possono essere legati da ID numerici registrati negli attributi. Nell'esempio di fig. 2 si mostra l'utilizzo dell'attributo SID (Struttura IDentificativo) per porre in relazione elementi del testo del Nardi.

Giunti a questa fase del progetto si è cercato di muovere i primi passi nella direzione del *Semantic Web*, che iniziava a far parlare di sé con sempre maggiore insistenza[26]. Il *Semantic Web*, almeno nelle idee di Tim Berners-Lee[27] inventore del *World Wide Web* e attuale direttore del W3C, è l'obiettivo futuro del www: esso si reggerà sui metadati e sulle ontologie. In tale prospettiva, RDF (Resource Description

[25] Chase 2002.

[26] Un significativo e recente esempio del crescente interesse proviene dal I Workshop *Rappresentazione della conoscenza nel Semantic Web culturale*, organizzato dal W3C Italia e dal Progetto Minerva, tenutosi a Roma presso Palazzo Massimo il 6 luglio 2004.

[27] Berners-Lee 1998.

Framework)[28], un linguaggio in sintassi XML per definire e esprimere ontologie e metadati, è considerato un pilastro per lo sviluppo del *Semantic Web*[29].

La struttura TEI (il cui primo modello ha ormai più di quindici anni) dispone di una sua peculiare struttura per i metadati, che vanno posti all'interno dell'elemento <teiHeader>. L'interesse generale della rete, tuttavia, nella prospettiva che si è detta, va verso modelli descrittivi dei metadati di tipo standard, conformi al modello strutturale che è stato scelto per il futuro del web.

La DTD della TEILite è stata perciò modificata per ospitare nell'*header* anche i metadati Dublin Core[30] espressi in RDF[31]. In questa prima fase abbiamo sperimentato l'uso dei metadati per la caratterizzazione e l'indicizzazione della risorsa. L'insieme di queste esperienze tende comunque verso la progettazione di una ontologia per l'annotazione semantica di testi archeologici relativi a ricognizioni e scavi già editi nel passato, ovvero verso la creazione di un modello di codifica semantica che, con il supporto dei metadati, possa rappresentare sia uno strumento per la realizzazione di nuove risorse sia il mezzo per il recupero e la fruizione informatica della conoscenza già acquisita nelle discipline archeologiche e già pubblicata in forma cartacea.

Per unire elementi di due diverse DTD in XML è richiesto l'uso di un particolare costrutto, pensato per assicurare l'univocità della marcatura e prevenire collisioni degli elementi: il *namespace*[32]. Occorre ricordare che, quando si utilizzano i *namespaces*, ad esempio in un

---

[28] Su RDF cfr. Miller 1998. Riferimento ufficiale da parte del W3C è *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation, 22 February 1999 <http://www.w3.org/TR/REC-rdf-syntax>.

[29] Cfr. in particolare Signore 2001, 2002a, 2002b.

[30] Dublin Core è un set di metadati costituito da 15 elementi <http://dublincore.org/documents/dces/> creato per la catalogazione delle risorse elettroniche; è particolarmente diffuso nell'ambito dell'editoria, delle biblioteche e delle istituzioni, poiché ben si presta alla descrizione delle caratteristiche di un libro o di un documento editi in forma digitale. La Dublin Core Initiative risiede all'indirizzo Internet <http://dublincore.org>.

[31] Cfr. *Expressing Simple Dublin Core in RDF/XML*, nel sito della Dublin Core Metadata Iniziative <http://dublincore.org/documents/2002/07/31/dcmes-xml/>.

[32] Cfr. la specifica tecnica sul sito W3C: *Namespace in XML* <http://www.w3.org/TR/REC-xml-names/> e il bel testo chiarificatore di T. Bray (Bray 1999).

ambiente di lavoro quale XMLspy, il testo risulterà all'analisi del *parser well-formed*, ovvero conforme allo standard XML, ma assolutamente non valido (*unvalid*) riguardo alla DTD dichiarata nello *statement* !DOCTYPE dell'*header*. I *namespaces* stabiliscono esclusivamente una regola onomastica per elementi appartenenti a DTD diverse, ma non si interessano di come queste DTD siano poi realmente interconnesse[33]. Ora, se il *parser* di un editor XML incontra nell'analisi del documento un errore, termina immediatamente il processo di *parsing*. Ciò costituisce un problema, perché impedisce di verificare la validità complessiva del documento marcato. Per essere certi che il resto del documento sia valido, è necessario dichiarare nella DTD principale anche gli elementi che si useranno con il prefisso del *namespace*.

Poiché si è scelto di inserire i metadati DC all'interno dell'elemento <note> (contenuto in <fileDesc> – <noteStmt>) del <teiHeader>, tale elemento è stato modificato nella DTD TEILite per poter accettare al suo interno anche l'elemento <Description> del *namespace* RDF (fig. 3). Le altre dichiarazioni necessarie a completare l'integrazione degli elementi DC devono essere inserite nel subset DTD interno del documento.

Come è visibile nella fig. 4, dopo aver dichiarato la versione XML utilizzata e il tipo di codifica si include l'entità parametrica ARCHEO_NEW che contiene tutte le integrazioni semantiche di carattere archeologico inserite nella TEILite. Si dichiara poi il modello di contenuto di <rdf:Description>, che fino a questo momento era ancora vuoto; si passa successivamente a dichiarare il contenuto di ogni singolo elemento che vi verrà inserito. Il contenuto di <rdf: Description> è costituito dagli elementi del *namespace* DC, che è assegnato al set di elementi Dublin Core. Tutti gli elementi appartenenti a DC devono essere dichiarati con contenuto di tipo #PCDATA. In ultimo, prima di usare gli elementi contenuti nei *namespaces* DC e RDF è necessario far riferimento all'origine URI dei modelli, come si vede nella fig. 5, in cui si presenta il <teiHeader> integrato dei metadati DC-RDF.

Il modello presentato permette, attraverso un adeguato foglio di stile, la ristrutturazione del documento e la sua impaginazione secondo criteri gerarchici. Permette inoltre di utilizzare i dati, attraverso gli

[33] Un articolo chiave per comprendere come risolvere il problema è quello di J.E. Simpson (Simpson 2001).

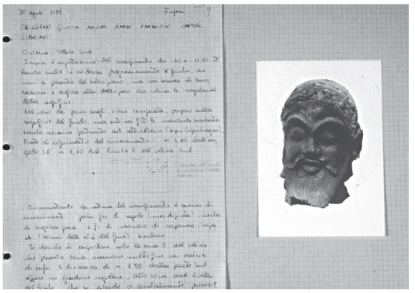
ID e le coordinate contenute in alcuni attributi, all'interno di sistemi GIS; è altresì adatto alla realizzazione di sistemi di *information retrieval* basati su discriminanti spaziali e contenutistiche e non impedisce la fruizione del documento nel suo più semplice aspetto di testo letterario. La valenza plurima del documento è esaltata e la fruizione e la rappresentazione delle informazioni sono delegate esclusivamente allo strumento informatico, che consente l'accesso al documento.

Il testo del Nardi, dopo la marcatura, è un documento del tutto conforme alla DTD TEILite, e può quindi usare il foglio di stile standard pubblicato dalla TEI. Il sito web del TEI Project mette a disposizione per i documenti TEILite dei fogli di stile modulari, in grado di trasformare il documento XML nel formato HTML o PDF. Utilizzando i fogli di stile standard pubblicati dalla TEI, il documento può essere facilmente composto nella veste grafica di un testo tradizionale, adatto ad un lettore generico (fig. 6). Il foglio di stile TEI deve essere personalizzato impostando il contenuto di alcuni parametri (il processo è detto 'parametrizzazione'), allo scopo di inserire intestazioni, logo, collegamenti ipertestuali e altre particolarità grafiche locali. Il compito può essere affrontato con l'ausilio di un'applicazione Internet quasi del tutto automatizzata creata dalla TEI.

È possibile tuttavia approntare un diverso foglio di stile XSLT[34], che trasformi il documento in una sorgente di informazioni strutturate, fruibili nella prospettiva di uno studio sistematico delle strutture archeologiche descritte nel testo (fig. 7).

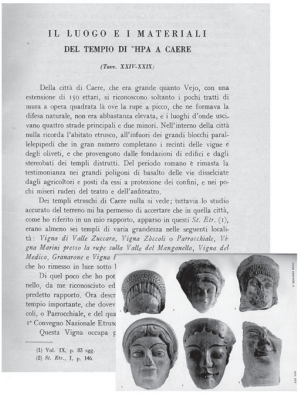
Claudio Barchesi

[34] Grosso, Walsh 2000.



a

c



b

d

Fig. 1. Diverse tipologie di documenti sottoposti a codifica elettronica: a) diari di scavo; b) rapporti di scavo editi; c) itinerario archeologico di E. Nardi; d) documentazione relativa al territorio falisco.

```
<STRUTTURA SID=”1” NOME=”Mura di cinta”>
Le fortissime mura di cinta A.A’.A”.A”’…lasciavano sempre aperto il lato che guarda la collina.
<SUB_STRUTTURA NOME=” Spezzature mura di cinta” SID=”1” SSID=”1”>
Nelle sezioni secondo la <RIFTAVOLA NOME=”Spezzature mura reticolate” SID=”1” SSID=”1” TID=”1”> spezzatura a.b.-c.d secondo d.e ed e.f Pian. Tav. I</RIFTAVOLA>…
…
</ SUB_STRUTTURA>
…
</STRUTTURA>
```

Fig. 2. Attributi di elementi usati come chiavi relazionali.

```
<!ELEMENT note (#PCDATA | …… | rdf:Description)*>
```

Fig. 3. Modifica all’elemento <note> della DTD TEILite.

```
<?xml version=”1.0” encoding=”ISO-8859-1”?>
<!DOCTYPE TEI.2 SYSTEM “teixliteARCHEO.dtd” [
        <!ENTITY % TEI_OLD “IGNORE”>
        <!ENTITY % ARCHEO_NEW “INCLUDE”>
<!ELEMENT rdf:Description (dc:Title, dc:Creator, dc:Subject, dc:Description, dc:Publisher, dc:Contributor, dc:Date, dc:Type, dc:Format, dc:Identifier, dc:Source, dc:Language, dc:Relation, dc:Rights, dc:Coverage)>
        <!ATTLIST rdf:Description about CDATA #REQUIRED>
        <!ELEMENT dc:Title (#PCDATA)>
        <!ELEMENT dc:Creator (#PCDATA)>
        <!ELEMENT dc:Subject (#PCDATA)>
        <!ELEMENT dc:Description (#PCDATA)>
        <!ELEMENT dc:Publisher (#PCDATA)>
        <!ELEMENT dc:Contributor (#PCDATA)>
        <!ELEMENT dc:Date (#PCDATA)>
        <!ELEMENT dc:Type (#PCDATA)>
        <!ELEMENT dc:Format (#PCDATA)>
        <!ELEMENT dc:Identifier (#PCDATA)>
        <!ELEMENT dc:Source (#PCDATA)>
        <!ELEMENT dc:Language (#PCDATA)>
        <!ELEMENT dc:Relation (#PCDATA)>
        <!ELEMENT dc:Rights (#PCDATA)>
        <!ELEMENT dc:Coverage (#PCDATA)>
]>
```

Fig. 4. Intestazione di un documento XML con TEILite integrata e metadati DC.

```
<TEI.2>
<teiHeader>
<fileDesc>
<titleStmt>
<title>Ruderi delle Ville Romano-Sabine nei dintorni di Poggio Mirteto</title>
<author>Ercole Nardi</author>
</titleStmt>
<publicationStmt>
<p>Pubblicato nel <date>1885</date> Edizione XML di Claudio Barchesi, giugno 2003 </p>
</publicationStmt>
<notesStmt>
<note id="rdf" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://www.purl.org/dc/">
<rdf:Description about="bagnidilucilla.xml">
<dc:Title>Ruderi delle Ville Romano-Sabine nei dintorni di Poggio Mirteto</dc:Title>
<dc:Creator>Ercole Nardi</dc:Creator>
<dc:Subject>Archeologia, Sabina, Ricognizioni archeologiche in Sabina, E. Nardi, Poggio Mirteto</dc:Subject>
<dc:Description>"Edizione XML del manoscritto di Ercole Nardi: Ruderi delle Ville Romano-Sabine nei dintorni di Poggio Mirteto "</dc:Description>
<dc:Publisher>Testo digitalizzato e commentato dal Museo Civico Ercole Nardi di Poggio Mirteto nel 2002, marcatura XML a cura di Claudio Barchesi ISCIMA-CNR 2003</dc:Publisher>
<dc:Contributor>Museo Civico Ercole Nardi di Poggio Mirteto </dc:Contributor>
<dc:Date>1885</dc:Date>
<dc:Type>ASCII text</dc:Type>
<dc:Format>text/xml</dc:Format>
<dc:Identifier>MSS96</dc:Identifier>
<dc:Source>Biblioteca di Archeologia e Storia dell'Arte di Palazzo Venezia in Roma</dc:Source>
<dc:Language>IT</dc:Language>
<dc:Relation>http://soi.cnr.it/iscima</dc:Relation>
<dc:Rights>Nessun copyright</dc:Rights>
<dc:Coverage>Sabina, Poggio Mirteto</dc:Coverage>
</rdf:Description>
</note>
</notesStmt>
<sourceDesc>
<p>bagnidilucilla.xml</p>
</sourceDesc>
</fileDesc>
</teiHeader>
…
</TEI.2>
```

Fig. 5. Sezione *header* di un documento TEILite corredato di metadati Dublin Core.

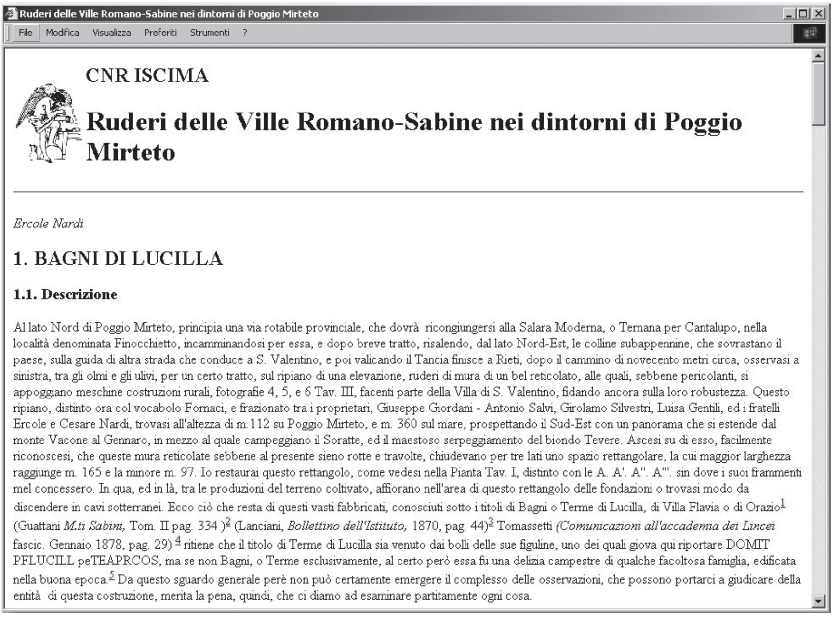


Fig. 6. Applicazione del foglio di stile TEILite.

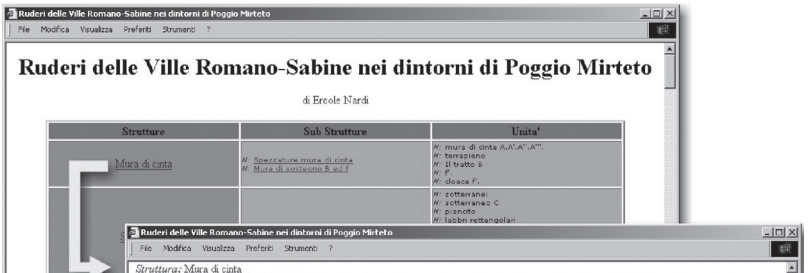


Fig. 7. Resa in Internet Explorer di un foglio di stile adeguato alla visualizzazione della marcatura archeologica.

The CollectioOrg System. An Open Hyperdocument System for Museum Collections

1. *Introduction*

Over the last few years, museums find themselves increasingly under pressure:

- economic pressure for more efficient and rational ways of operation
- public demand for access to the common cultural heritage over the Internet
- call for scientific and cultural European collaboration

To these ends, Content Management Systems and Digital Asset Management Systems systems have been proposed. Whereas the former would concentrate on the creation and management of digital resources, the latter instead would concentrate on the aspects of publishing (or selling) the digital assets over various channels to the public, to institutions, commercial content providers etc.[1].

Making all or selected parts of the collection accessible over the Internet, allowing collaborative efforts, rendering the management more effective and cost-efficient, opening new ways of marketing etcetera would bring in fact numerous advantages, as has been proved in other areas like news, legal, financial, medical.

But the following problem areas are still blocking the wide-spread use of such systems:

1. Museums are confined to one vendor's proprietary application or application technology means that the data's future is linked to that particular vendor's, with all problems regarding long-time storage and even possible data loss.

[1] Cfr. DC-TI2.

2. Costly knowledge domain islands are created, where information is at one's disposal inside the restricted information system, but without the possibility of collaboration with other departments or museums, thereby giving away possible synergism effects.

3. The specific information structure of a museum collection is manifold and complex, and traditional data formats are almost always lossy, i.e. the digitalized data – due to the constraints of the traditional table/row RDBMS – do not contain 100% of all previously available information.

4. Finally, contextual text documentation which is related but not regarded direct part of a specific object is mostly left out in the digitalizing process.

To overcome these fundamental shortcomings, the CollectioOrg system introduces the Extensible Markup Language (XML), which has been deployed in other business areas with great success, for use in an open hyperdocument system.

An «Open Hyperdocument system for the Interconnection of Knowledge Domains» was first proposed by Douglas C. Engelbart in 1990[2] to overcome the problems plaguing the airplane industry. At the time, because of incompatible document types and the lack of common protocol for interchange, important documents could not be exchanged neither horizontally: between departments of the same contractor, nor vertically: e.g., between departments and external suppliers.

To achieve interoperability, he argued for the introduction of an open document type and a transparent mechanism for linkage to other documents or media objects. Every document would be referenced by a unique 'catalog number' and could be searched, retrieved, transformed and re-used by other knowledge domains. The main characteristics of his system were: 'mixed-object documents'; 'explicitly structured documents'; 'view control of objects form, sequence and content'; 'library system' allowing referencing by catalog number; 'personal signature encryption'; 'access control'; 'every object being addressable'; finally 'inter-linkage between hyperdocuments and other system data'.

[2] Engelbart 1990.

The basic hypermedia document type of the Internet – the (X)HTML document type – and its related linking mechanisms adress these issues only in part[3]: in fact, structuring, view control, adressability and library system have been developed only in part or are lacking alltogether.

Whereas at the time of Engelbart's writing, the isolated knowledge domains he refers to were the airplane industry, today it's the museums' whose infrastructure is characterized by complex, proprietary and mostly incompatible systems. It is not uncommon to find different incompatible databases even inside the same museum structure.

In a way very similar to Engelbart's proposal, the CollectioOrg system, to achieve the widest possible interoperability, specifies:

1. A XML hyperdocument scheme for museum records used for internal production and for web applications, which combines the advantages of a document structure with a database-enabled format,

2. A mechanism of hyper-linking using URNs, combined with a basic definition of a web service for inter connectivity with other departments, other museums and for external providers.

The CollectioOrg system is therefore not a program, or an application, but the definition of a framework: an Open Framework, because all specifications are openly published, every interested museum is invited to participate, all specifications can be used without restrictions, and, finally, all technologies used are those specified by the W3C or OASIS and therefore non-proprietary.

2. *XML for the Data*

XML, the eXtensible Markup Language, is in many ways the perfect means of transport for museum records:

XML is a W3C specification like, for example, HTML. Therefore there is no risk of being trapped in a proprietary technology with all

[3] Connolly 2002.

its negative consequences: not only the data created with this technology, but all the applications built upon it can be used – and shared, for example with other departments or smaller museums – without encountering licensing problems.

XML is application-independent. That is, you don't need the original application who wrote the data to read and interpret it correctly later on another system. XML documents instead can be created, read and modified on any OS with any editor, either using simple (schema-aware) text editor, or more complex (and mostly commercial) editors with customized graphical user interfaces.

The XML specifications are surrounded with a rich application environment: XSL-T and XSL-FO for transformations into HTML, RTF, PDF, or Text; XPath and XQuery for accessing and searching documents, XML Schema for controlling the document structure and values, etc. All these specifications are W3C specifications, too.

Since XML documents are basically text files marked up in XML, the format is particularly suitable for long-time storage, because it is, like a Rolodex card, human readable and usable even when the applications which created the data don't exist anymore. The data format will outlive the applications which created the data.

Since XML isn't binary (machine-) code but basically a text file with added mark-up, it is portable across all existing OS without the need for conversions. In short time XML has thus become an international standard for data exchange, especially between different DB (even RDBMS).

XML and all its surrounding technologies are fully Unicode enabled, with the standard encoding being UTF-8. This allows not only the use of foreign language scripts including, e.g., Old Italic, but facilitates also the transcription of epigraphical texts using special characters for interpunction and combining glyphs denoting missing or illegible characters etc. The Unicode specification even allows to define Roman numerals as such.

Unlike data stored in RDBMS, XML is a self-describing format: that is, element and attribute names in XML are mostly self-explaining. With the advent of XML 1.1, even for element names the full Unicode range is available. This makes XML data understandable even in those cases, when all accompanying documentation has been lost and only the data stream is preserved.

The XML Schema specification, Part II, has made XML Data-Type

aware. Not only are a wide range of data types available out-of-the-box for specifying element and attribute values (date, time, integer, etc.), but more types can be customized: e.g., a custom scheme for museum inventory code like sc-[0-9]{4}.

XML is equally used for the storage of structured text, for example large documents divided into various sections, chapters, and paragraphs, thereby preserving all the hierarchical structure. This allows one to re-deploy smaller fragments of the original text document, which can now be reassembled in new contexts: for example, search result pages, but also printed museum catalogs built upon chosen XML data. Data structures (measurements etc.) and document structures (textual descriptions of restorations etc.) can be mixed freely side-by-side, within the same XML document.

Information in XML is grouped not in rows and tables like in 'flat' conventional databases (one field for every value) but in a hierarchical or tree-like structures, where the values of the single nodes have their meaning according to their position inside the tree i.e. its context. This lets you create very flexible patterns which reflect the way humans organize information.

View capability was one of the Engelbarts original specifications. It can be achieved using the mechanisms provided by CSS and XSLT so that the same data can be presented in different views, for example an Outline View, which would give the basic structure only, with foldable sections and paragraphs.

Finally, using XMLSignature for digital signatures and XMLEncryption for encryption of parts of documents or whole documents, there are instruments at hand to protect XML documents from eavesdropping and secure them from tampering. Ownership and authorship can be ascertained during exchange of documents, e.g. between museums, and certain informations can be hidden, e.g. insurance details, addresses of private collections etc.

3. *RelaxNG for the Logic*

XML and its related technologies allow the separation of logic and data. That is, the data and document parts which make up the XML files (called instances) are physically separate from an abstract definition of its correct syntax and content (called schema).

The schema defines firstly the sequence of elements and element groups; secondly, their possible attributes and children; thirdly, the possible data types (text, number, date, boolean, integer etc.). Therefore, schema file contains all the system logic: i.e., it prescribes exactly how information has to be stored in order to give a syntactically correct document. It offers word lists and controls the data-input by allowing only valid elements and values.

Like XML itself, the schema definitions are written in non-proprietary formats, which makes them usable without any licensing questions, and guarantees their long-term value. Mostly used is the XML Schema Definition – the format developed by the W3C – but RelaxNG, an OASIS specification and at the same time an ISO Standard, developed by James Clark, has gained significant ground lately: it is widely credited with being easier to write and to maintain, especially for larger projects.

A physically separated schema enables data input outside traditional museum systems: data can now be written – with its logic controlled – all over the museum departments, without the need for being connected to a central server. Personnel can work on private lap-tops which can be taken home at night – since there is no proprietary software, there is no risk of theft or breach of licensing. Just having the schema file/s on a floppy disk is enough for controlling all the data input, word lists included. This is in stark contrast to the data-input interfaces of traditional systems which have all the data input logic hard-wired, i.e. the built-in definitions are proprietary code which cannot be separated from the system.

3.1. *CollectioML for records*

The CollectioML Schema in RelaxNG has been optimized to allow for storing the whole history of an object, from its finding or acquisition until its latest exhibition and collocation, including various restorations and changes of ownership.

It is especially suitable for historic collections because all the information is grouped in epoch/time blocks which allow a synoptic overview of the object's history. This allows one to have different horizontal and vertical views of the history of a collection in special time frames, e.g. the collocation of the statues in the main entrance hall in the years 1880-90.

The records have unlimited extensibility, that is groups and sub-

groups can be added as the subject needs it, for example restorations of parts of an object only, various collocations and identifications, etc. This leads to asymmetric records, something unknown in conventional RDBMS: records which have extremely different structures depending on the subject they deal with (but are always in accordance to the referenced schema).

A RelaxNG schema can be built in single modules, that is a central file contains the principal schema structure, and other external files contain specification details. This makes it not only easier to gain a more structural view of the – rather complex – schema, but it also means that updates and corrections can be restricted to single components, thus gaining a higher degree of reliability for the whole system. This is especially useful when dealing with word lists, which in part are obtained from outside sources and recompiled for use in an XML schema: in this case, the update rhythm is not foreseeable. Finally it is possible to shift the necessary customizations – number schemes for museum inventories, word lists etc. – into one of the smaller files which can be edited more securely by the museum personnel.

The flexibility of XML and its schema languages allows for the incorporation of legacy or temporary data, too: i.e., data which has only a very narrow function – because it is extremely short lived or because its value is doubtful – can be inserted into the XML document using a special container which accepts any kind of elements or attributes. This is extremely useful during the passage from older archival systems – be that digital or card based – where certain running numbers or meta information (even when it is not fully understood anymore) can preserved until the new XML document system has been finished.

Finally, the definition of a public Namespace permits side by side existence of different vocabularies inside the same document. This eases the creation of composite documents combining, e.g., DocBook and CollectioML elements, and avoids name collision.

3.2. *DocBook/TEI for documents*

Whereas for museum needs an XML Schema had to be built from scratch, for pure text documents such schemata have been developed long ago. The most important are DocBook and TEI.

DocBook, developed by Norman Walsh and hosted by OASIS, has been over the years optimized for technical documentation, but can be used for almost any common text, including bibliographical references.

Instead TEI, the format developed by the Text Encoding Initiative, has been specifically developed for the mark-up of literate texts.

With the change of development of DocBook from a DTD based schema to one based on RelaxNG, an amalgam of TEI and DocBook mark-up has become possible thus creating a super-format (nicknamed «TEIBook») suitable for any needs.

Finally, the Open Office XML Format the has been presented as specification to OASIS. If accepted, this could become a very successful XML document format.

4. *Identification and Reference of Resources*

4.1. *Introducing URNs*

The Schema definitions are used to mark-up the following information resources:

1. Records:
   These are single museum records describing museum objects. The CollectioML schema is used to digitalize them.

2. Inventories:
   These are mostly entry books, acquisition lists, collection inventories, etcetera in table form. Each have to be digitalized with its own particular schema.

3. Internal documents:
   These are text documents describing acquisitions, restoration reports, opinions, contracts etcetera including letters. They are digitalized using the DocBook or TEI schema.

4. Other contextual documents:
   These are archival documents, scientific articles, etcetera which come from external sources but relate in one way or the other to the museum patrimony and its history. Most of these can be digitalized using the DocBook/TEI schemata, too.

Thus it is possible not only to describe the museum patrimony itself but also to have all the contextual documentation at hand: the

finding and collocation of museum objects, their history, restoration etcetera including archival documentation and scientific articles.

All of these resources have to be uniquely identified. To achieve this, Uniform Resource Names are deployed, which provide a consistent way to identify resources. Unlike the more common URLs, URNs do not point to physical locations of resources, – which might change. Instead they provide persistent and unique names – like an ID – which can be used for unlimited referencing without knowing the physical location of the resource at any given time. References between museum records, text documents (articles), inventories, image descriptions, from document to document or from document to external data etcetera can thus easily be accomplished by just naming the resource.

URNs can be written using UTF-8 encoding, and thus internationalized, using the escape mechanism proposed for the Internationalized Resource Identifier (IRI).

The full syntax of an URN is «urn:namespaceIdentifier:namespaceSpecificString».

For CollectioOrg URN, «collectio» has been chosen as Namespace Identifier, making the schema like this «urn:collectio[member Code]:[objectCode]».

The member code is from 0001 to FFFF in hexadecimal system. This leaves room for 65.536 members. The object code instead is defined freely by the museum. An example would be «urn:collectio0001:sc1234». Once an URN has been assigned, it cannot be changed anymore: the URN is effectively frozen.

Thus we gain the following advantages:

1 Resources which do not yet have a physical equivalent (for example, because the digitalizing work is still in progress) can nevertheless be referenced – and fully used as soon as it is physically available,

2 Resources can reference other resources. For example, a description of a series of restoration interventions can reference the objects cited by their URN – without caring about their physical location,

3 References can be made even to external resources, whose physical location (URL) isn't controlled,

4 Resources which have lost original their location (through closure of a museum, repatriation of a work of art etc.) can be transferred and stored on any other domain without invalidating the URN.

4.2. *De-Referencing URNs*

Since URNs are unique identifiers for resources, but do not locate them physically, it is necessary to de-reference the URNs to URLs.

In most cases, URNs can be de-referenced by the host system itself, because the URN points to a resource residing inside the same database. A simple query like «http://collectio.foo.com/search?urn=urn:collectio001:sc1234» gives back the requested resource – the Query effectively then becoming the URI.

In those cases where the URN points to a resource with a foreign member code – and thus outside the home domain («urn:collectio.org:005E:123456») –, the URN has to be de-referenced using a look-up table, available on the Internet server of CollectioOrg and in local copies, which lists the member codes and assigns the relevant domain names and their corresponding servers.

5. *Modelling Relationships between Documents*

Graph-based data models for modelling relationships between hyperdocuments have been analyzed by Eugene Eric Kim and Ken Holman in 2002[4]. They are used in two ways: the CollectioML Infoset provides a graph model for inter-relationships between documents, and an external graph modelling language (Topic Maps) provides for relationships between different classification systems.

5.1. *Representing Relationships between Documents*

Expressing relationships between items are basic requirements for any serious museum documentation. Not only is it necessary to express qualifying relationships between items inside the same museum, but also to items in other museums.

The possibility to declare relationships like part-of, related to, similar-to, equal-to, model-for, copy-of, and reproduction-of is part of the Infoset of each CollectioML document.

[4] Kim, Holman 2002.

Since the CollectioOrg systems is using a database-architecture, a Back-Link capability is possible: i.e., getting a list of all the documents (records, texts, image descriptions etcetera) which link back to a given document. This allows to view the context of a document in the many ways it has been cited in other documents or referenced in other records and is enormously helpfull, for example, when dealing with stylistic comparisons.

5.2. *Topic Maps for Managing Different Classification Schemes*

The modular structure of the CollectioML schema makes it possible to have different national and institutional word lists, classifications systems and customization layers live side-by-side. And since Unicode UTF-8 encoding is used across the system, full-text search facilities in different languages are provided from the start.

But searching across the different classification systems cannot be provided with the same ease, because a one-to-one mapping between abstract or very specific classification schemes is quite impossible.

In this case, Topic Maps are a solution. Topic Maps – for which there is a XML syntax – define subjects (topics), resources (occurrences) which are stand-ins for the subjects, and relationships between them (associations).

The mechanism can thus be used to build lists of topics, associations between these topics or different classification schemes and to occurrences in existing information resources. This enables building a system where detailed category searches can extend even over different classification schemes.

6. *Reference Platform*

A Reference Platform has been built which is used as showcase and for experimental features alike. Nearly all components are built using Open Source software – the only exception being Java which is used to achieve cross-platform capability.

The system is developed in the classical 3-tier architecture.

The persistence layer is a native XML database. A native XML database used as storage container (a so-called «XML Repository») is the only efficient way for storing and retrieving XML documents, especially structured ones, because it uses the same range of W3C technologies as the rest of the system and is therefore easy to integrate.

XQuery, another non-proprietary W3C technology, is the tool for interrogating large collections of XML documents. XQuery not only allows for search and retrieval of XML nodes – including full-text search – but can be used to build new XML documents out of the retrieved XML nodes. This facilitates, for example, its use in search interfaces, where lists of search hits can easily be presented.

XUpdate is used for administration work across whole groups of documents, like changing values of elements or attributes or updating the XML document structure.

The logic layer of the system catches the incoming GET requests and interprets them according to his in-built rules: sending an XQuery to the database, de-referencing the URN, eventually connecting to outside resources, executing a series of transformations and finally representing the composite result. By hiding the inner operations we are able to have a consistent interface, regardless of what resources are tinged or how the results are composed.

The presentation layer has two basic modes implemented: List View and Document View. Like any search portal, a request first returns a List View – document records, image records, related documents, inventories – and then the chosen document.

With regard to visualization and export, the W3C technologies XSL-T and XSL-FO are perfect solutions. They have become widely adopted because of their power, transparency and ease of customization. In fact, a lot of templates – e.g. for DocBook – are already available. XSL-T is used for easy export to XHTML, also for export to plain text, such as the report format requested by the ICCD. Instead XSL-FO is the first step for transformations to RTF and PDF. The interconnectivity between domains is offered along the principles of a REST architecture[5]: a state-less transfer of representations of resources where each server is connected via HTTP and where the data stream is pure XML (XML-over-HTTP). This allows interconnection not only inside a museum domain, but also between different Internet servers.

[5] Fielding 2000.

7. *Summary*

Implementing the CollectioOrg system as a service that can be reached via LAN/WAN or between selected museums or even as public Web Service that can be accessed over normal Internet protocol has far-reaching conclusions:

1 The resources of formerly isolated knowledge domains (museums' information systems) are accessible and interchangeable:

This allows not only a more efficient collaboration between different departments and museums but also between museums and universities or institutes, facilitating scientific research and allowing pan-European research projects. At the same time, the resources can be published to the general public, allowing wider participation and raising the public awareness of the museum.

2 The resources are able to create larger entities by transclusion:

By aggregation of information resources from different knowledge domains it is possible to have a common generalist view over all resources. Here we are using the transclusion principle, i.e. not copying but referencing of external resources[6]. One possible view collects all the pieces of a collection even when the collection itself is dispersed over various museums and countries. Other views comprehend collocations (e.g. all statues or objects inside a hall), or periods (e.g., all Roman busts from the Antonine period), or classifications (e.g., all Doric columns), or historic settings. The aggregation of resources is bigger than their sum.

3 The resources are uniquely addressable and have all the necessary characteristics of hyperdocuments:

The use of URNs (instead of URLs) gives the museums' objects the characteristics of real objects which exist /per se/, and not only inside HTML pages describing them. Furthermore, this

[6] Nelson 1993.

allows incoming and outgoing references not only to other documents but to any other kind of data – even data in foreign formats and applications. This is a precondition for building applications on top of the CollectioOrg system, e.g. Content Managing Systems for managing the objects, or Digital Asset Management Systems for publishing them over various channels.

4 The resources are in structured, open and internationally recognized formats, and therefore they are reusable for other applications:

Standard data formats and standard access interfaces are the preconditions for building third-party applications, which can access data over the Internet as web services or via an internal API. For example, the development of Content Managing Systems is immensely facilitated by standard formats, and the same is true for Digital Asset Management Systems and the necessary implementation of a Digital Rights Management for granting differentiated access levels for partner institutions, commercial partners etc. (e.g. access to high resolution images).

8. *Usage Cases*

Uses of the CollectioOrg System go beyond a mere Museum Information System. The following examples – only few of the many possible – comprise external organizations and service providers, the scientific community and the general public:

1 Web Portals: The most common of all usages is the opening of a web portal for showcasing the museum inventory to a broader public.

2 Mobile Electronic Museum Guides: A mobile museum guide systems can be realized in the same manner, with the only difference being that the output is customized for the use on small hand-helds and only part of the available information is displayed.

3 Virtual Exhibitions: The combined net of federated servers allows exhibitions of material stored across different museums and different countries to be displayed in virtual showrooms. This can be used to gather art objects which are otherwise inaccessible, to reunite collections which are now dispersed, etc.

4 Catalog Publishing: The database can also be used to produce printed catalogues: either the collection's information is assembled as a list, to be handed over to the typesetter in native XML format (modern typesetting software can handle XML) or the database output is rendered directly in PDF or Postscript.

5 Museum Information Interchange: Museums can access another museum's collections on-line, and check their own items against theirs. This is especially useful for items which come from common historic collections and are dispersed across multiple museums.

6 Virtual Reality Projects: Using the HTTP connectivity, the museum material can be made available to Virtual Reality projects, even remote ones, to provide background information about the objects displayed.

7 External providers: External museum service providers can build their own applications on top of the museum's web service for the whole range of services they offer. This can be used, for example, by restoration and storage companies, or companies providing image scanning services.

8 Museum exhibition organizers: Museum exhibition organizers can keep track of the latestcollocation of the items, and use their physical descriptions to organize transport and display more efficiently.

9 Providers of Digital Asset Management Systems: External DAMS providers can integrate their system into the museum's web service, eliminating the costs for a second, parallel web presence.

10 Insurance Companies and Police Departments:Insurance com-

panies can be given special access control to the museum repository (with secure authorization) to check the insurance coverage and status of the museum objects. Police departments can use the system to look for items recovered or reported stolen.

11 Scientific Research: Queries – full-text queries or queries by category – across multiple servers are essential to research projects. Universities and research institutions can even include the URNs of the museum's objects inside their texts, allowing the reader to automatically look them up on the museum's website.

12 Cultural Heritage Management: The CollectioOrg system allows a common view over nationally and internationally dispersed museum collections and provides thus the necessary tools for the evaluation and management of the common cultural heritage.

Klaus E. Werner

XML per gli archivi storici. Iniziative e progetti

1. *L'Encoded Archival Description*

Qualunque riflessione sull'applicazione dei metalinguaggi di marcatura alla descrizione di archivi storici deve prendere le mosse dalle sperimentazioni di codifica di strumenti di ricerca archivistici intraprese, nei primi anni Novanta, nella biblioteca della Berkeley University in California con il ricorso allo Standard Generalised Mark-up Language (SGML). Avviato nel 1993, il Berkeley Finding Aids Project (BFAP) si proponeva di «sviluppare uno standard non proprietario per la codifica in formato digitale di strumenti di ricerca quali inventari, elenchi sommari, indici»[1]. L'adozione di SGML come linguaggio di codifica portò all'elaborazione di una prima DTD sperimentale, che, dopo il coinvolgimento nel progetto di altre istituzioni di conservazione e di ricerca, compresa la Library of Congress di Washington, e l'interesse manifestato dalla comunità archivistica statunitense e dalla Society of American Archivists, fu ulteriormente perfezionata ed assunse, nell'estate del 1995, il nome di Encoded Archival Description[2]. Nei tre anni successivi le versioni alfa e beta della DTD di EAD, elaborate dall'apposito gruppo di lavoro, furono

Maurizio Savoja ha letto e discusso con l'autore una prima stesura di questo contributo, offrendo un generoso contributo di critiche e suggerimenti, di cui ho tenuto ampiamente conto. Di ciò gli sono molto grato. Ovviamente la responsabilità di errori ed omissioni è soltanto mia.

[1] *Development of the Encoded Archival Description DTD*, revisione dicembre 2002, <http://www.loc.gov/ead/eaddev.html>.

[2] Una ricostruzione delle vicende che hanno portato all'elaborazione di EAD, oltreché nel documento del sito 'ufficiale' del progetto citato nella nota precedente, si può leggere in Pitti 1997. Il Berkeley Project fu presentato assai precocemente in Italia dallo stesso autore (Pitti 1994).

ampiamente testate e dibattute nella lista di discussione istituita dalla Library of Congress. La versione 1.0 di EAD è stata rilasciata nel 1998[3], accompagnata dalla pubblicazione di due numeri monografici della rivista della Society of American Archivists[4] e dall'edizione della *Tag library* e delle *Application guidelines*[5]. Nel contempo la *Tag library* fu anche resa disponibile sul sito 'ufficiale' del progetto, ospitato all'interno di quello della Library of Congress, che del mantenimento di EAD era, nel frattempo, diventata l'agenzia responsabile.

Nel 2002, dopo una fase di revisione durata molti mesi e che ha visto il coinvolgimento di archivisti di altri paesi (Canada, Australia, Gran Bretagna e Francia) è stata rilasciata una nuova versione della DTD (EAD 2002), che non ha introdotto radicali modifiche rispetto alle versione 1.0, ma ha provveduto soprattutto ad adeguare taluni suoi elementi alla nuova edizione del *General International Standard Archival Description*, ISAD (G), pubblicata nel 2000[6]. Frattanto è stata realizzata la conversione della DTD in XML, linguaggio di marcatura con il quale essa era già sostanzialmente compatibile fin dalla versione 1.0 del 1998.

Questa rapida illustrazione delle origini di EAD suggerisce quanto esso sia il frutto di una iniziativa fortemente sostenuta dalla comu-

[3] La struttura di EAD – nel disegnare la quale è stato tenuto conto dell'esperienza della *Text Encoding Iniziative*, riproponendone gli elementi tutte le volte che era possibile – si articola in due parti principali, l'<eadheader> e l'<archdesc>. La prima contiene i metadati relativi allo strumento di ricerca codificato (autore, editore, date di redazione, revisione, ecc.). Essa è affiancata da un elemento denominato <frontmatter>, che provvede lo strumento di ricerca digitale di un frontespizio preparato *ad hoc*. Il tag <archdesc>, invece, racchiude tutti gli elementi che sono utilizzati per la descrizione archivistica vera e propria (identificazione, provenienza, contenuto, ecc.) e che servono in prima istanza per la descrizione del fondo nel suo insieme. All'interno dell'elemento <archdesc>, raggruppate per mezzo di appositi tag, possono essere annidate, secondo una struttura ricorsiva, le descrizioni dei livelli inferiori al fondo, per le quali è possibile utilizzare tutti i tag definiti all'interno di <archdesc>. Una esposizione molto chiara delle logiche che hanno ispirato l'elaborazione di EAD è in Ruth 1997, in parte riedito nella EAD *Tag library* citata nella nota 5.

[4] EAD 1997, Part 1; EAD 1997, Part 2.

[5] EAD 1998; EAD 1999.

[6] Cfr. EAD 2002.

nità archivistica americana che vi ha riconosciuto, nella sostanza, lo strumento fondamentale, per non dire esclusivo, per la produzione di strumenti di ricerca archivistici in formato digitale e per la loro pubblicazione sul web. Lo specifico contesto statunitense, da un lato, ed alcuni caratteri peculiari di SGML ed XML spiegano il notevole successo che EAD ha conosciuto, successo che, come avremo modo di ricordare, si è in seguito esteso anche ad alcuni paesi europei.

EAD è stato, in primo luogo, la risposta archivistica all'egemonia dei formati bibliografici, utilizzati anche dagli archivisti – in particolare da quelli operanti nelle università e in altre istituzioni di ricerca – per l'elaborazione di record descrittivi destinati a confluire nei cataloghi on-line dei network bibliografici quali OCLC e RLG. Nel corso degli anni Ottanta, grazie alla messa a punto di un formato MARC per la descrizione di archivi e di collezioni di manoscritti, il MARC-AMC (Archives and Manuscript Collections), affiancato da un apposito manuale catalografico, *Archives Personal Papers and Manuscripts* (APPM)[7], un numero cospicuo di 'schede' bibliografiche di fondi archivistici e di raccolte di materiale manoscritto erano entrate a far parte dei cataloghi on-line dei principali network bibliografici americani: agli inizi degli anni Novanta il database del Research Libraries Group (RLIN) ne contava circa 450.000[8]. Tuttavia, la capacità dei formati bibliografici, e del MARC in particolare, di rappresentare in maniera efficace gli archivi era sostanzialmente limitata. Come ha notato Daniel Pitti[9] – cui va in parte non piccola il merito dell'elaborazione di EAD –, le ragioni principali di ciò sono sostanzialmente due. La prima è che i record in formato MARC hanno una lunghezza massima di centomila caratteri, cioè di circa trenta pagine di testo libero, mentre, non raramente, gli strumenti di ricerca archivistici ne contano molte di più. La seconda, e ben più importante, è che gli strumenti di ricerca archivistica sono articolati in genere secondo complesse relazioni gerarchiche che, procedendo dal generale al particolare, traducono in sede di rappresentazione descrittiva la classica articolazione degli archivi in partizioni, serie, sottoserie, ecc. che comprendono registri, buste, fascicoli, a loro volta formati di sottofascicoli, inserti oppure singoli documenti (vedi fig. 1).

[7] Hensen 1989. Ne esiste anche una traduzione italiana, cfr. Hensen 1996.

[8] Per un ulteriore approfondimento di queste tematiche cfr. Vitali 1994.

[9] Pitti 1997.

I tracciati descrittivi bibliografici, compreso il formato MARC, hanno una scarsa capacità di rappresentare relazioni gerarchiche di questo genere.

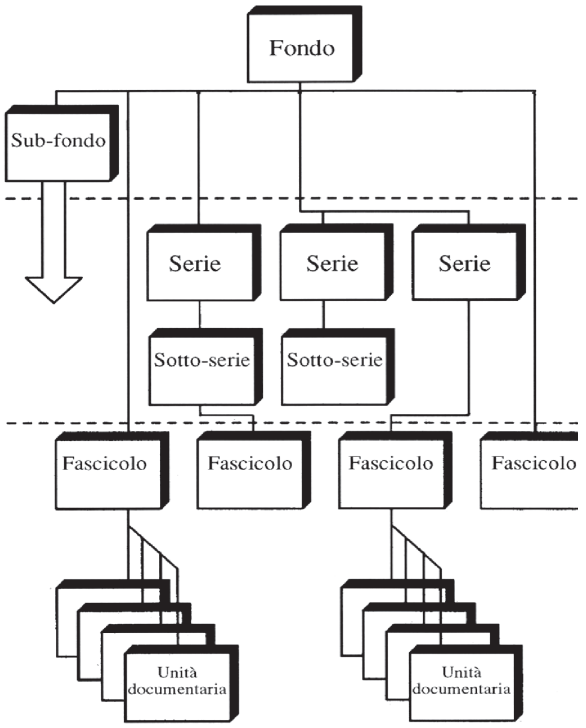


Fig. 1. La struttura gerarchica dei fondi archivistici così come è rappresentata in ISAD (G).

Come gli archivisti sono sconsolatamente consapevoli – ha scritto Daniel Pitti – il MARC è stato concepito per rappresentare in primo luogo dati che descrivono entità bibliografiche irrelate; raccolte complesse che richiedono livelli decrescenti di analisi sovraccaricano rapidamente la struttura del MARC. Al massimo, può essere aggiunto un secondo livello, ma il genere di informazioni fornito è limitato[10].

I metalinguaggi di marcatura, al contrario, per le loro specifiche caratteristiche, hanno dimostrato di essere in grado di rappresentare efficacemente in ambiente digitale quella particolare tipologia di

[10] Ivi, p. 275.

'testi', che è costituita dagli strumenti di ricerca negli archivi e, in particolare, dal più tipico e diffuso di tali strumenti, l'inventario, che descrive, in genere, un singolo fondo archivistico. Nonostante la complessità della descrizione archivistica e le molteplici possibili peculiarità che ciascun inventario può avere, è possibile individuare una struttura logica ed una semantica delle informazioni che, all'ingrosso, tutti gli inventari condividono e che, almeno in parte, hanno in comune con altri strumenti di ricerca archivistici di tipo diverso (ad esempio le guide). I promotori di EAD si sono così proposti di elaborare una DTD in grado di codificare quella struttura valendosi proprio di alcune delle specifiche proprietà di SGML e di XML che permettono di riprodurre, anche in ambiente digitale, le tipiche caratteristiche della descrizione archivistica tradizionale. L'organizzazione gerarchica dei documenti SGML ed XML consente infatti di riprodurne l'articolazione in livelli, di collocare in maniera chiara ciascun livello all'interno della catena gerarchica cui appartiene e quindi di rappresentare in maniera efficace l'ereditarietà delle informazioni da un livello all'altro, di navigare, infine, all'interno della struttura, passando da un livello all'altro, senza smarrire l'organica visione d'insieme dell'inventario (e quindi del fondo). Inoltre una codifica SGML o XML appare particolarmente idonea a gestire quella combinazione di testo non strutturato e di dati strutturati o semistrutturati, che caratterizza gli strumenti di ricerca archivistici. Essa supporta, al contempo, il recupero e l'indicizzazione di componenti del testo appositamente codificate (ad esempio titoli, date, ecc.) e di elementi informativi quali nomi di persona, toponimi, ecc., che nei tradizionali inventari cartacei sono soliti confluire negli indici.

Insomma, grazie ai caratteri di SGML e ancor più di XML, EAD si è dimostrato capace di realizzare una trasposizione in ambiente digitale delle modalità di descrizione degli archivi tradizionalmente praticate dagli archivisti. Ciò è stato certamente una delle ragioni della sua popolarità, accelerata dalla diffusione di Internet. A ciò si è aggiunta la possibilità, per gli istituti di conservazione che lo hanno adottato, di ricorrere a strumenti software per la codificazione e la pubblicazione sul web degli strumenti di ricerca, già messi a punto da altre istituzioni, realizzando così significative economie di scala. A parte rare eccezioni (rilevante quella dei National Archives di Washington) si può affermare, senza allontanarsi troppo dal vero, che EAD costituisce il formato di gran lunga più utilizzato negli Stati Uniti per la pubblicazione sul web di strumenti di ricerca archivisti-

ci[11]. Ma EAD ha conosciuto anche un discreto successo in Europa. In Gran Bretagna innanzitutto, dove è stato impiegato nello sviluppo di alcuni importanti progetti nazionali[12], e in Francia dove è stato adottato largamente sia a livello di Amministrazione archivistica nazionale, che a livello periferico. «La forza di EAD», ha scritto Catherine Dhérent della Direction des Archives de France «risiede nella sua capacità di diffusione di strumenti di ricerca archivistici sulla rete e di costituzione di ampi depositi di risorse»[13].

Ad EAD si ricorre quindi largamente nei progetti di conversione in formato digitale di strumenti di ricerca esistenti su supporto cartaceo (a stampa o meno). Ma non solo. Infatti EAD, per i propri caratteri specifici e per le prerogative dei metalinguaggi di marcatura con cui è realizzato, presenta una feconda ambiguità. Esso è certamente, in primo luogo, uno strumento per rappresentare una peculiare classe di 'testi', cioè gli inventari e gli altri strumenti di ricerca archivistici. Tuttavia, dato che questi testi, indipendentemente dal loro formato e supporto, rappresentano a loro volta degli 'oggetti', delle entità

[11] Un'ampia lista di progetti di pubblicazione in rete di strumenti di ricerca archivistici codificati secondo la DTD EAD può vedersi nelle *EAD Help Pages* ospitate sul sito della Virginia University; *EAD Sites Annotated*, <http://www.archivist.org/saagroups/ead/sitesann.html>. Per la maggior parte i progetti segnalati sono, ovviamente, americani. Fra i più cospicui, l'*Online Archive of California*, <http://www.oac.cdlib.org/> e *OASIS, Online Archival Search Information System* dell'Harvard University, <http://findingaids.harvard.edu/dfap.html>.

[12] Cfr., ad esempio, *Access to Archives (A2A)*, un *repository* on-line, che raccoglie circa 77.500 strumenti di ricerca dalle caratteristiche e dai contenuti descrittivi alquanto difformi, migrati dal supporto cartaceo al formato digitale. Gli istituti di conservazione coinvolti sono circa 300: *A2A: Access to Archives*, <http://www.a2a.org.uk>.

[13] *Une DTD pour la description des fonds d'archives et collections spécialisées, l'EAD*, di C. Dhérent, <http://www.archivesdefrance.culture.gouv.fr/fr/archivistique/presentationEAD.html>. Sulla utilizzazione di EAD negli archivi francesi molte informazioni nel «Bullettin d'information francophòne sur l'EAD» <http://www.archivesdefrance.culture.gouv.fr/fr/publications/DAFbuldtd.htm> e nella rassegna sulle *Journées européennes sur le DTD EAD et EAC*, 7-8 octobre 2004, <http://www.archivesdefrance.culture.gouv.fr/fr/archivistiques/journeeseuropeennes.pdf>. Per gli strumenti software utilizzati nei progetti di applicazione di EAD agli archivi francesi cfr. il sito web del progetto *Pleade*, <http://www.pleade.org>.

del mondo reale, cioè i fondi archivistici e le loro componenti, di EAD può essere proposta una 'lettura' che sottolinea principalmente questo aspetto e che ne promuove l'utilizzazione quale strumento per descrivere *ex novo*, direttamente in formato digitale, gli archivi oppure che lo adotta come formato di scambio e migrazione dei dati fra applicazioni diverse e per realizzare progetti di interoperabilità che prevedano la confluenza di strumenti di ricerca prodotti da molteplici istituzioni archivistiche, all'interno di *repository* comuni. Insomma EAD non raramente viene considerato ed utilizzato come standard di formato dei dati. In effetti ha assunto molti caratteri dello standard *de facto*, anche se resta, in primo luogo, un prodotto americano, sia per come è stato sviluppato e come viene gestito ed aggiornato, sia per il fatto che riflette non raramente certi caratteri specifici della pratica archivistica statunitense o, comunque, un peculiare approccio a nodi teorici cruciali della descrizione archivistica che si riverbera nella natura di alcuni elementi e nella loro ambigua denominazione e che, dire il vero, lascia un po' perplessi[14] .

[14] Spicca per la sua notevole ambiguità concettuale il tag <origination>. La *Tag library*, se da un lato ne stabilisce l'equivalenza con l'elemento 3.3.1 di ISAD (G), cioè con la denominazione del soggetto produttore, dall'altro precisa che per *origination* può intendersi «such agents as correspondents, records creators, collectors, and dealers»: un po' di tutto, insomma, con una disinvoltura teorica che è solo in minima parte attenuata dalla possibilità di utilizzare l'attributo *label*, per precisare «the role of the originator, e.g., 'creator', 'collector', or 'photographer'». L'attributo, infatti, non ha la funzione di qualificare effettivamente la relazione fra il soggetto indicato nel tag <origination> e la documentazione descritta, ma solo di aggiungere, in sede di presentazione, un'informazione che «may help identify for a finding aid reader the role of the originator». Con ciò non solo si evita di confrontarsi con un notevole problema teorico (quello di gestire correttamente il rapporto fra archivi e loro soggetti produttori) ma si rischia anche una certa confusione dal punto di vista pratico, accomunando sotto un unico tag informazioni che concettualmente dovrebbero rimanere ben distinte. I possibili effetti negativi sono rimarchevoli soprattutto laddove EAD sia assunto come strumento per realizzare l'interoperabilità fra progetti diversi. Per le citazioni cfr. EAD 2002: *EAD elements*. <*origination*> *Origination*, <http://www.loc.gov/ead/tglib/elements/origination.html>.

2. *La (s)fortuna di EAD in Italia*

EAD non avuto in Italia una diffusione paragonabile a quella registratasi in altri paesi. Se è pur vero che nel nostro paese spicca l'assenza di iniziative di studio e divulgazione di EAD (quali ad esempio una traduzione in italiano della *Tag library*) paragonabili a quelle intraprese in altri paesi come la Francia, assenza che può averne negativamente condizionato la popolarità, è pur vero che, probabilmente, le ragioni della scarsa applicazione di EAD sono inscritte nei percorsi che l'applicazione dell'informatica agli archivi storici ha seguito in Italia a partire dalla fine degli anni Ottanta e nelle conseguenti scelte di modello descrittivo e di strumenti tecnologici utilizzati. Infatti, quando EAD si è affermato a livello internazionale, in Italia, contrariamente a quanto era avvenuto in altre situazioni anche a noi vicine (come ad esempio, ancora una volta, in Francia), si era già sviluppata una discreta attività di messa a punto e di utilizzo di programmi per l'inventariazione basati sull'utilizzo di *information retrieval systems* (come CDS-ISIS promosso dall'UNESCO) oppure di *database management systems* di tipo relazionale (quali ad esempio Sesamo, promosso dalla Regione Lombardia, oppure, più recentemente Arianna, sviluppato dal Cribecu e dalla cooperativa Hyberborea, e Guarini della Regione Piemonte ed altri ancora)[15]. Tali programmi, inizialmente progettati per produrre inventari cartacei, oppure per essere consultati *on site*, per lo più con la mediazione degli archivisti responsabili, avevano dato buona prova di sé e ad essi in genere sono ricorse le istituzioni archivistiche e i singoli archivisti che si proponevano di utilizzare l'informatica nel proprio lavoro. A questi programmi di inventariazione, si sono poi affiancati, negli anni Novanta, progetti di più ampio respiro che miravano a realizzare censimenti sia dei fondi conservati negli Archivi di Stato che di quelli, posseduti o detenuti a vario titolo da privati ed istituzioni pubbliche, sui quali si esercita la vigilanza delle Sovrintendenze archivistiche. Nonostante tali progetti non abbiano raggiunto completamente il proprio scopo, essi hanno sedimentato banche dati locali, talvolta assai cospicue, attorno alle quali si è svolta a partire dalla fine degli anni Novanta

[15] Per una rassegna dei software per l'inventariazione archivistica sviluppati in Italia negli ultimi anni, cfr. Vitali 2003a.

un'intensa attività di reingegnerizzazione e di recupero dei dati in altri ambienti software[16].

Inoltre, vi è stata, nel nostro paese, una precoce ricezione degli standard promossi dal Consiglio internazionale degli archivi e del modello di articolazione della descrizione proposto dall'*International Standard Archival Authority Record for Corporate bodies, Persons and Familes*, ISAAR (CPF)[17]. Al contrario delle pratiche descrittive tradizionali che prevedono l'integrazione delle informazioni sul contesto di produzione degli archivi – cioè sulla storia e la struttura degli enti, persone o famiglie, la cui attività pratica è stata all'origine della loro sedimentazione – nel cuore stesso della descrizione dei singoli fondi (ad esempio nell'introduzione di un inventario), ISAAR (CPF) suggerisce una gestione separata delle descrizioni dei soggetti produttori per mezzo di file d'autorità posti in relazione con le pertinenti descrizioni archivistiche, indipendentemente dal livello che occupano nella gerarchia descrittiva. Questa soluzione permette, fra l'altro, di rappresentare brillantemente situazioni assai complesse, come quelle che, in un panorama archivistico storicamente assai stratificato come quello italiano, si sono venute determinando fra i fondi archivistici e i loro soggetti produttori. Permette, in sostanza, di gestire una relazione molti a molti (un complesso archivistico può aver avuto più soggetti produttori e viceversa un soggetto produttore può aver sedimentato più complessi archivistici), evitando di schiacciarla in una relazione uno a uno come tradizionalmente è avvenuto con gli inventari prodotti su supporto cartaceo.

Inoltre, nei sistemi descrittivi, per dir così, di 'seconda generazione', quelli progettati per il recupero delle banche dati di *Anagrafe* e concepiti fin dall'origine per essere pubblicati sul web, il modello della

[16] Sul progetto *Anagrafe degli archivi italiani* e sui successivi progetti di reingegnerizzazione, cfr. *Anagrafe* 2000.

[17] Le traduzioni italiane della seconda edizione di ISAD (G) e di ISAAR (CPF) sono consultabili sul sito dell'Associazione nazionale archivistica italiana, <http://www.anai.org/politica/standard_descri.htm>. Cfr. anche il numero monografico sulle *Norme per la descrizione archivistica* della «Rassegna degli Archivi di Stato», LXIII, 2003, che contiene il testo inglese e la traduzione italiana degli standard, oltre ad articoli di presentazione.

cosiddetta descrizione separata, ma interrelata, di soggetti produttori e complessi archivistici, è stato reso ulteriormente più complesso, attraverso una concezione per così dire 'allargata' del contesto di produzione che ha portato all'integrazione, nel modello descrittivo, di altre 'entità', quali i contesti politico-istituzionali in cui le istituzioni hanno operato, gli ambiti territoriali di loro pertinenza, i soggetti (istituzioni, persone) che conservano ora o hanno conservato nel passato i complessi archivisti descritti, ecc. Tipici esempi di questi sistemi sono la guida on-line dell'Archivio di Stato di Firenze, il Sistema unificato delle Soprintendenze archivistiche, e il Progetto lombardo archivi in Internet (PLAIN), la cui realizzazione informatica ad opera del Cribecu si è affidata, visti gli specifici caratteri del loro tracciato (la prevalenza di dati strutturati, di molti vocabolari controllati, ecc.) e la necessità di gestire un insieme complesso di relazioni fra numerose entità (spesso anche in relazione riflessiva) ad un DBMS relazionale come Oracle[18].

Come si vede, l'applicazione dell'informatica alla descrizione archivistica nel nostro paese non ha teso semplicemente a riproporre metodi e pratiche consolidati, ma ha piuttosto cercato di innovarli, anche radicalmente. Ciò è derivato dalla convinzione che elaborare descrizioni archivistiche in ambiente digitale e comunicarle attraverso Internet non è la medesima cosa che farlo con i tradizionali supporti cartacei. Anche in questo, come in altri casi, il mezzo condiziona fortemente la struttura e i contenuti delle informazioni e spinge a riconsiderare le forme di organizzazione delle conoscenze e le modalità di venirne in possesso.

EAD, per parte sua, se è perfettamente in grado, come abbiamo visto, di riprodurre in ambiente digitale i tradizionali stili di descrizione archivistica non è apparso in grado di rappresentare con la stessa efficacia quelle architetture descrittive previste dagli standard internazionali o suggerite dalle potenzialità dagli strumenti digitali,

[18] Sulla guida on-line ai fondi dell'Archivio di Stato di Firenze, <http://www.archiviodistato.firenze.it/siasfi/>, cfr. Bondielli, Vitali 2000 e Taglioli, Valgimogli 2002, entrambi anche on-line, <http://www.archiviodistato.firenze.it/materiali/materiali.htm>. Sul Sistema unificato delle Soprintendenze archivistiche cfr. Bondielli 2001; Pastura *et al.* 2004. Su PLAIN, <http://plain.unipv.it/plain/index.php> cfr. Savoja, Weston 2003, anche on-line, <http://eprints.rclis.org/archive/00000326/01/savoja_ita.pdf>; Bondielli 2002; per l'avvio del progetto Savoja 2001; Almini 2004.

che sono state sperimentate negli ultimi anni nel nostro paese. Solo da poco tempo è stata varata un'iniziativa che mira a fornire di adeguati strumenti coloro che vogliano applicare, usando EAD ed XML, il modello della descrizione separata e del controllo d'autorità dei soggetti produttori d'archivio. In seguito ad un incontro, promosso a Toronto nel 2001 da alcuni componenti del gruppo che aveva a suo tempo avviato il processo di sviluppo di EAD è stata infatti messa a punto una DTD XML, denominata Encoded Archival Context, per la descrizione di enti, persone, famiglie che si trovino ad essere soggetti produttori di archivi[19]. La DTD elaborata, che è attualmente alla sua versione beta, sembra in effetti molto solida e ben congegnata, sia dal punto di vista della struttura logica, che della idoneità a rappresentare in modo efficace le informazioni relative al contesto di produzione degli archivi. Essa inoltre si propone di gestire complesse reti di relazioni fra gli stessi soggetti produttori e di questi con il materiale (non solo archivistico) che essi hanno prodotto o di cui sono stati autori (si pensi a materiale bibliografico oppure ad opere d'arte) o con cui essi intrattengono un qualche rapporto significativo. Il valore e l'importanza dell'iniziativa è testimoniata anche dal fatto che la totale compatibilità fra EAC e il corrispondente standard internazionale

[19] Su EAC, oltre al sito web 'ufficiale', <http://www.iath.virginia.edu/eac/> cfr. Pitti 2003, anche on-line <http://eprints.rclis.org/archive/00000316/02/pitti_ita.pdf>. La DTD di EAC, che ha un struttura simile a quella di EAD, è suddivisa in due parti principali, l'<eacheader>, e il <condesc>. Il primo contiene i metadati relativi alla redazione del file XML (chi lo ha compilato, quando, seguendo quali norme, ecc.). Il secondo che comprende gli elementi necessari per la descrizione di un ente di una persona o di una famiglia, è ulteriormente suddiviso nei seguenti cinque elementi, che includono a loro volta altri elementi: <identity>, che è l'unico obbligatorio e contiene la denominazione, o le denominazioni, e le altre informazioni che identificano l'ente, la persona o la famiglia; <desc> contiene le informazioni relative alla storia, struttura, ecc. dell'ente e della famiglia, oppure la biografia della persona; <eacrels> permette di codificare le relazioni del soggetto descritto con altri enti, persone o famiglie; <resourcerels>, consente invece di porre in relazione il soggetto descritto con risorse informative (archivi, documenti bibliografici, ecc.) di cui esso è produttore, autore, ecc.; <funactrels> permette di porre la descrizione di un ente in relazione con una definizione standardizzata delle funzioni e delle attività, da esso svolte. Ciascun file EAC descrive, singolarmente, un ente, una persona o una famiglia, che può, tuttavia avere più denominazioni o intestazioni.

è stata assunta come impegno programmatico da parte del gruppo che ha sovrinteso alla sua realizzazione, cosicché la DTD elaborata presenta una notevole aderenza ad ISAAR (CPF), il cui processo di revisione, conclusosi pochi mesi fa, ha del resto tenuto conto della contestuale messa a punto di EAC[20]. Per questo stretto rapporto con ISAAR (CPF) e per la presenza di vari archivisti europei ed australiani nel gruppo che lo ha promosso e sviluppato, EAC presenta una maggiore apertura internazionale di EAD. È, fra l'altro, probabile che anche l'agenzia per il suo mantenimento possa essere una istituzione od un consorzio di istituzioni europee.

Va tuttavia rilevato che, almeno fino ad oggi, EAC non sembra conoscere un sostegno ed una popolarità equivalenti a quelli che avevano contraddistinto le origini di EAD. I progetti relativi ad sua reale utilizzazione sono per adesso limitati e nel più rilevante di essi, che ha costituito una sorta di test della DTD stessa, il progetto LEAF[21], EAC, piuttosto che come strumento per la codificazione di record d'autorità di soggetti produttori, è stato adottato come formato di scambio fra sistemi. Resta in sostanza ancora da capire se, dopo il suo varo definitivo, EAC sarà fatto proprio dalla comunità degli utilizzatori di EAD per integrare i loro strumenti di ricerca pubblicati sul web con strumenti per il controllo d'autorità e la descrizione dei soggetti produttori e se, viceversa, l'esistenza di EAC gioverà ad EAD, accentuandone la popolarità.

3. *XML e gli archivi storici italiani*

Pur non essendo EAD così popolare come in altri paesi, non sono tuttavia mancati negli anni scorsi, in Italia, progetti che, nell'ambito degli archivi storici, hanno fatto propri i metalinguaggi di marcatura e che, più recentemente, hanno avviato una qualche forma di impiego di EAD come formato di rappresentazione delle descrizioni

[20] Cfr. Vitali 2003b, anche on-line, <http://eprints.rclis.org/archive/00000334/02/vitali_ita.pdf>.

[21] Il progetto LEAF (*Linking and Exploring Authority Files*) si propone di collegare *authority records* di uno stessa persona presenti in sistemi bibliotecari e archivistici, attraverso un meccanismo di *harvesting* e di analisi dei dati: oltre al sito web 'ufficiale', <http://www.crxnet.com/leaf/>, cfr. Weber 2003.

archivistiche. Di essi si cercherà qui di seguito di tracciare una rapida rassegna, senza avere la pretesa di illustrare tutti quelli esistenti e, soprattutto, senza illudersi di riuscire a farlo in maniera esauriente e con sufficienti dettagli. La bibliografia esistente e le informazioni reperibili in rete potranno integrare quanto non saremo in grado di documentare a sufficienza.

Dei metalinguaggi di marcatura sono state, in primo luogo, messe a frutto le capacità di rappresentare l'articolazione interna e l'organizzazione logica dei testi, nonché la specifica semantica delle informazioni contenute, per realizzare una trasposizione di strumenti di ricerca originariamente cartacei in un formato digitale che si proponga di salvaguardarne la struttura e i significati. Di tali prerogative si è in particolare giovato fin dal 1998 il progetto 'pionieristico' – come è giustamente definito dai suoi responsabili, che fanno riferimento al Centro metodologie e applicazioni per archivi storici del Consorzio Roma ricerche – di informatizzazione della *Guida generale degli Archivi di Stato*, l'opera in quattro volumi che fra il 1981 il 1994 è venuta pubblicando la descrizione sommaria dell'insieme dei fondi conservati in ciascuno degli Archivi di Stato italiani[22]. Data la struttura sostanzialmente omogenea del testo a stampa, nel quale le diverse tipologie di informazioni possono essere identificate con una certa precisione grazie alla loro disposizione sulla pagina, alle dimensioni dei corpi e alle spaziature è stato possibile realizzare un riconoscimento semiautomatico delle diverse componenti del testo e produrre, tramite OCR, file di testo marcati secondo una DTD SGML appositamente definita. Così 'destrutturate' le singole informazioni sono state riversate nel programma di *information retrieval* Highway® I. R., sviluppato dalla 3D Informatica S.r.l. di Bologna, che è dotato di funzionalità che lo assimilano ad un database. La banca dati è stata successivamente pubblicata in rete e può essere sia esplorata seguendo l'articolazione dell'opera a stampa, che interrogata sulla base della struttura dei campi del database. Inoltre è stata prevista una procedura di aggiornamento dei dati, che consente di integrare nella banca dati le eventuali modifiche alle singole voci, cioè alle descrizioni dell'insieme dei fondi conservati nei singoli Archivi di Stato, fatte pervenire su file da questi ultimi.

[22] *Guida generale degli Archivi di Stato*, 1981-1994. Molte informazioni sul progetto sul suo sito, <http://www.maas.ccr.it/cgi-win/h3.exe/aguida/findex>; una sua recente illustrazione in Rendina 2003.

Più recentemente, il progetto ha conosciuto un'ulteriore fase di sviluppo che, nel quadro di un aggiornamento degli strumenti tecnologici e dell'adozione di XML, ha compiuto un percorso in qualche modo inverso a quello seguito inizialmente. Si è cioè provveduto ad esportare «dal DB Highway [l'] intera *Guida generale* in formato XML, secondo un modello dei dati (DTD) direttamente derivato dalla struttura logica del DB»[23]. Ciò rende in sostanza possibile conservare le informazioni direttamente in file XML, intervenire su di essi per aggiornare o modificare le singole voci, tramite un form di *data entry* cui è possibile accedere via rete, produrre un insieme differenziato di *output*, grazie all'applicazione di vari fogli di stile, consultare, infine, i file XML in forma nativa grazie all'apposito motore di ricerca ad oggetti XML nativi eXtraWay®. Questa nuova fase del progetto ha in qualche modo segnato anche una sorta di riorientamento nella logica che sottende il modello dei dati, che si è, almeno in parte, svincolato dalla puntuale rappresentazione dell'originario testo a stampa. Ciò, oltre a consentire una più efficiente descrizione della struttura dei complessi archivistici, ha posto le condizioni per introdurre, in prospettiva, elementi di distinzione fra le descrizioni dei fondi archivistici e quelle dei soggetti produttori e di sperimentare, a questo fine, una codificazione di questi ultimi basata sull'Encoded Archival Context. Con una modalità del genere dovrebbero infatti essere gestiti i cosiddetti *Repertori delle magistrature uniformi*, cioè i profili istituzionali degli uffici periferici degli stati postnapoleonici, la cui la pubblicazione a stampa, originariamente prevista, non si è potuta fino adesso realizzare.

Se questa ulteriore fase è in corso di ultimazione ed i suoi risultati definitivi non sono ancora pubblicamente disponibili in rete, se ne possono tuttavia intuire i caratteri consultando altri progetti del Centro MAAS, per la realizzazione dei quali è stata adottata una metodologia e strumenti tecnologici simili (Guida agli archivi storici delle Camere di commercio italiane[24], Guida agli archivi della Fondazione Istituto Gramsci di Roma[25], Progetto RinASCo, Recupero degli inventari

[23] Ivi, p. 91.

[24] Alla guida si accede a partire dall'URL <http://www.camerecultura.it/GuidaArchiviStorici2/index.htm>.

[25] Vedila all'URL <http://www.maas.ccr.it/GuidaGramsci/default.html>

degli archivi storici comunali della provincia di Latina[26], ed altri ancora[27]). Anche in questi casi si è proceduto al recupero di strumenti di ricerca esistenti su supporto cartaceo e alla loro codificazione in XML secondo DTD elaborate più o meno *ad hoc* che sono state poi mappate su quella di EAD. Dai file XML vengono generati *output* statici di diverso tipo, quali file HTML e PDF, consultabili in linea. Inoltre è possibile accedere direttamente ai file XML grazie al motore di ricerca eXtraWay® che consente sia la navigazione nella struttura gerarchica degli archivi che la ricerca per parola, libera o guidata.

Per l'insieme di questi progetti, EAD è venuto ad assumere il ruolo di tracciato di scambio che favorisce l'interoperabilità e, in prospettiva, l'integrazione e l'interrogazione comune. Si tratta di una modalità di utilizzo di EAD cui non è estraneo un altro progetto, DAMS (Digital Archives and Memory Storage) Solutions della società Regesta.exe che si basa, almeno in parte, su soluzioni tecnologiche non dissimili da quelle adottate nelle realizzazioni del Centro MAAS, ma che tuttavia, presenta vari caratteri specifici. In primo luogo, piuttosto che al recupero degli inventari cartacei, esso è finalizzato alla produzione di nuovi strumenti di ricerca in formato digitale. DAMS.Solutions è basato interamente su tecnologie web e su un'architettura *three-tier*, che consente all'utente di generare file XML, attraverso la compilazione di appositi form che gli propongono un subset di elementi e attributi di EAD, secondo la sequenza e la denominazione in lingua inglese che questi hanno in quella DTD. I file generati vengono salvati direttamente sul file server in formato XML nativo e come tali vengono visualizzati, in sede di navigazione ed interrogazione, mediante il motore di ricerca eXtraWay®. Alcune applicazioni di DAMS, come ad esempio quella realizzata presso l'Archivio di Stato di Napoli, consentono anche, sulla base di un tracciato appositamente definito, la gestione separata di *authority file* dei soggetti produttori, che si propone

[26] Vedine i risultati all'URL <http://www.maas.ccr.it/ProgettoRinasco/default.html>, cfr. anche Auricchio *et al.* 2002 e Mazzini, Veninata 2004.

[27] Per una dettagliata illustrazione dei progetti del Centro MAAS, molti dei quali fanno uso delle soluzioni cui si accenna nel testo, cfr. il sito del Centro <http://www.maas.ccr.it/>, alla voce «Progetti».

di tener conto del modello previsto dagli standard di descrizione internazionali[28].

Resta invece ancorato prevalentemente ad uso di XML quale strumento di trasposizione in formato digitale di strumenti di ricerca esistenti su supporto cartaceo, il progetto per la pubblicazione in rete degli inventari degli archivi comunali toscani pubblicati a stampa nel corso degli ultimi decenni. Il progetto è promosso dalla Regione Toscana e portato avanti dal Centro Signum della Scuola Normale Superiore di Pisa con la collaborazione anche di chi scrive. Dati i criteri non sempre omogenei di redazione di tali inventari, i quali a fronte di una struttura sostanzialmente simile, presentano un livello di dettaglio descrittivo assai differenziato, XML è apparso particolarmente appropriato, proprio perché permette di rispettare i diversi livelli di granularità dei dati presenti nei vari inventari, senza dover ricondurre questi ultimi ad un modello eccessivamente astratto, che rischia di far perdere, nella codificazione, preziose informazioni[29].

Un aspetto che differenzia questo progetto da altri segnalati sopra è che, in questo caso, si è cercato costruire un modello di codifica che distinguesse fra la rappresentazione del testo originario e quello delle 'entità' descritte al suo interno. Oltre ad una DTD che codifica l'inventario come documento bibliografico e che quindi ne rispetta la struttura editoriale (con le diverse premesse, presentazioni, i diversi apparati, ecc.), si sono messe a punto altre tre distinte DTD, che descrivono rispettivamente i complessi archivistici, le istituzioni di conservazione, e i soggetti produttori. Ciò ha permesso di liberarsi dall'appiattimento necessariamente indotto negli strumenti originari dai caratteri del supporto cartaceo, di restituire alla descrizione degli

[28] Per ulteriori dettagli su DAMS.solutions cfr. Barbanti, Bruno 2003 e Grossi 2003; inoltre si veda la documentazione presente sul sito web del programma ten-Telecom, all'interno del quale la realizzazione dell'applicativo è stata finanziata, <http://www.damssolutions.org/dams/>.

[29] Così, ad esempio, laddove è possibile, sono distintamente codificati i vari elementi della consistenza (cioè della quantità o estensione del materiale compreso in una determinata unità archivistica) quali il numero e la tipologia del materiale. Laddove ciò non sia consentito dal modo in cui la consistenza è espressa nel testo originario, ci si limita a codificare l'informazione all'interno del tag più generale, senza scendere ulteriormente nel dettaglio.

archivi la dinamicità propria dell'ambiente digitale e di renderla maggiormente conforme agli standard internazionali, in particolare a ISAAR (CPF). Ha consentito al contempo di mettere a punto DTD che, oltre a promuovere la conversione dalla carta al digitale per strumenti di ricerca già esistenti, possono essere utilizzate nel quadro di progetti di descrizione *ex novo* di archivi e soggetti produttori.

Anche in questo caso i file XML vengono visualizzati ed interrogati mediante un motore di ricerca, TReSy[30], che, sviluppato presso il Cribecu (ora Signum) per gestire e recuperare in modo efficiente l'ingente massa di informazioni presenti all'interno di testi letterari antichi codificati per finalità di ricerca e di analisi testuale, si sta dimostrando perfettamente in grado di restituire anche le articolate strutture dei fondi archivistici, consentendo la navigazione nella gerarchia dei complessi archivistici e nei link che li connettono alle descrizioni dei soggetti produttori e delle istituzioni di conservazione.

La mappatura fra le DTD elaborate per il progetto, ed EAD e EAC hanno dimostrato una quasi completa sovrapponibilità. È stato anche possibile stabilire un buon livello di corrispondenza fra elementi e attributi della DTD ed alcuni tracciati di banche dati particolarmente autorevoli, quali ad esempio quello del Sistema informativo unificato delle soprintendenze archivistiche. Ciò dovrebbe permettere, in prospettiva, l'esportazione dei dati codificati (o, più probabilmente, della parte di essi che si riferisce alle descrizioni di fondi e serie) verso il sistema archivistico cui la Regione Toscana e la Sovrintendenza archivistica di Firenze progettano di dar vita, ricorrendo proprio a SIUSA od a una sua versione adattata alle necessità locali.

3. *Conclusioni*

Al termine di questa, per quanto rapida, rassegna di iniziative e progetti portati avanti nel nostro ed in altri paesi, l'interrogativo se XML e i linguaggi di marcatura in generale siano applicabili anche agli archivi storici, ed in particolare alla loro descrizione, ha un sapore decisamente retorico. L'interrogativo che occorre porsi non è

[30] Maggiori dettagli su TReSy (*Text Retrieval System*), nelle pagine dedicategli sul sito del Centro Signum <http://www.signum.sns.it/index.php?id=492>.

se essi siano strumenti utili, ma semmai in quali contesti e per quali finalità essi si dimostrino particolarmente efficienti nel rappresentare, in formato digitale, le descrizioni degli archivi e delle entità che ad essi sono collegati. Non è evidentemente un interrogativo che possa essere risolto con una risposta definitiva, valida una volta per tutte, ma cui si deve cercare risposta nel concreto dipanarsi di progetti di applicazione dell'informatica alla descrizione degli archivi. Certo, il nostro veloce *excursus* ha messo in evidenza come vi siano delle finalità per le quali XML 'funziona' particolarmente bene: ad esempio la trasposizione digitale di strumenti di ricerca esistenti in formato cartaceo oppure la pubblicazione on-line di inventari e di altri strumenti archivistici che abbiamo un taglio, per così dire, tradizionale.

Ma ci sono anche altri ambiti nei quali XML si dimostra oggi quasi una scelta obbligata o comunque fortemente consigliata. Ad esempio nella messa a punto di formati di scambio dei dati fra sistemi diversi che utilizzino anche tecnologie fondate sui DBMS. Vista la strutturale – genetica verrebbe voglia di dire – alta portabililità di XML, esso appare particolarmente indicato per sostenere progetti di interoperabilità, di importazione-esportazione dei dati da un sistema all'altro oppure di elaborazione di protocolli comuni di interrogazione di banche dati distribuite. D'altronde alcuni dei sistemi complessi basati su architetture *three-tier*, citati in precedenza, come SIUSA o la Guida on-line dell'Archivio di Stato di Firenze, già oggi utilizzano XML come strumento di comunicazione dei dati fra i diversi 'strati' del sistema, proprio per la possibilità che XML offre di mettere a punto applicazioni standard che favoriscono l'interoperabilità fra sistemi.

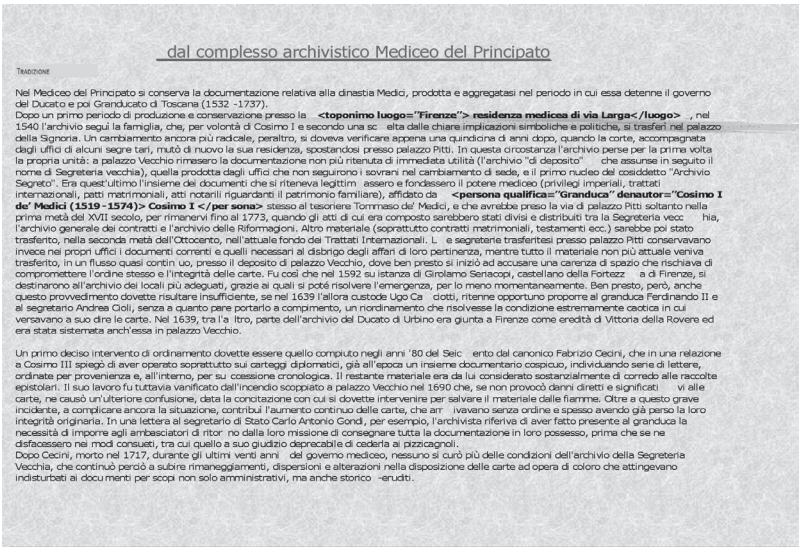
Per le proprie caratteristiche, insomma, XML si candida oggi ad essere il naturale formato per la realizzazione di un 'tracciato' di scambio standard per le descrizione archivistiche, per quella sorta di formato MARC per gli archivi di cui spesso e da più parti, a livello nazionale come a quello internazionale, si indica la necessità. Può già considerarsi EAD un tale tracciato? Mi sia consentito esprimere qualche perplessità, non solo per alcuni limiti per così dire 'teorici' che EAD dimostra e che ho segnalato sopra, ma anche per altre ragioni che hanno a che fare con le stesse origini di EAD. Esso è nato infatti come formato di rappresentazione di una determinata categoria di testi, gli strumenti di ricerca archivistici di tipo tradizionale, e per la loro pubblicazione sul web, piuttosto che come strumento di codificazione di descrizioni di archivi in quanto tali, cioè di rappresentazione formalizzata di informazioni relative ad 'oggetti' archivistici e alle entità ad

essi collegate. L'*imprinting* originario non può non condizionare l'uso di EAD quale formato di scambio, tant'è vero che quando esso viene adottato per una finalità del genere (come nelle applicazioni britanniche segnalate sopra) o all'interno di sistemi che mirano ad una descrizione archivistica 'pura' svincolata dalla sua rappresentazione all'interno degli strumenti di ricerca (come nell'esperienza italiana di DAMS.solutions), si ricorre in genere ad un subset piuttosto limitato degli elementi e degli attributi della DTD di EAD. Un subset che, in particolare, elimina gran parte, se non tutti, i marcatori orientati alla formattazione testuale, necessari per rappresentare in formato digitale quella categoria di 'documenti' o 'testi' che sono gli strumenti di ricerca e gli inventari in particolare. Insomma, EAD può indicare una strada per realizzare un formato di scambio, può forse essere una base di partenza o uno strumento che parzialmente va in quella direzione, ma la sensazione di chi scrive è che il lavoro da fare per raggiungere lo scopo sia ancora molto. A cominciare da una maggiore chiarezza concettuale su cosa debba davvero intendersi per 'formato di scambio delle descrizioni archivistiche'.

Infine, dei metalinguaggi di marcatura va segnalata un'ulteriore apprezzabile peculiarità. Le loro caratteristiche specifiche hanno la capacità di mettere in evidenza, per contrasto, taluni limiti o aspetti problematici dei DBMS che, in contesti di utilizzo come la descrizione di archivi storici, appaiono con particolare evidenza. Ne vorrei indicare almeno due. In primo luogo una certa rigidità, che si manifesta soprattutto nella difficoltà di gestire, nello stesso modello dei dati, rappresentazioni a diverso grado di granularità e di strutturazione delle stesse tipologie di informazioni. XML, invece, è come è noto, molto più flessibile e dà la possibilità di prevedere marcatori che possono contenere direttamente dati oppure altri marcatori più specifici. In secondo luogo – ed è il limite che mi sembra più rilevante – l'impossibilità di gestire indici controllati di termini nel contesto dei campi a testo libero, come invece consente di fare XML grazie alla possibilità di marcare ogni elemento di un testo e di qualificarlo con specifici attributi.

Per un archivista che, come chi scrive, da lungo tempo lavora con i DBMS e ne apprezza le molte qualità, ma ne riconosce anche gli aspetti problematici, costituirebbe la realizzazione di una aspirazione a lungo vagheggiata poter un giorno avere a disposizione uno strumento tecnologico che avesse l'efficienza dei DB nel gestire relazioni molteplici e complesse ed indici dei campi strutturati, ma che permettesse

anche di fare operazioni simili a quella indicata nella seguente schermata tratta dalla banca dati del sistema fiorentino ed opportunamente modificata, nella quale possano intravedersi, all'interno di un campo a testo libero, marcatori che dovrebbero consentire di estrarre indici normalizzati di nomi di persona, toponimi, ecc.



dal complesso archivistico Mediceo del Principato

TRADIZIONE

Nel Mediceo del Principato si conserva la documentazione relativa alla dinastia Medici, prodotta e aggregatasi nel periodo in cui essa detenne il governo del Ducato e poi Granducato di Toscana (1532 -1737).
Dopo un primo periodo di produzione e conservazione presso la **<toponimo luogo="Firenze"> residenza medicea di via Larga</luogo>** , nel 1540 l'archivio seguì la famiglia, che, per volontà di Cosimo I e secondo una sc elta dalle chiare implicazioni simboliche e politiche, si trasferì nel palazzo della Signoria. Un cambiamento ancora più radicale, peraltro, si doveva verificare appena una quindicina di anni dopo, quando la corte, accompagnata dagli uffici di alcuni segre tari, mutò di nuovo la sua residenza, spostandosi presso palazzo Pitti. In questa circostanza l'archivio perse per la prima volta la propria unità: a palazzo Vecchio rimasero la documentazione non più ritenuta di immediata utilità (l'archivio "di deposito" che assunse in seguito il nome di Segreteria vecchia), quella prodotta dagli uffici che non seguirono i sovrani nel cambiamento di sede, e il primo nucleo del cosiddetto "Archivio Segreto". Era quest'ultimo l'insieme dei documenti che si riteneva legittim assero e fondassero il potere mediceo (privilegi imperiali, trattati internazionali, patti matrimoniali, atti notarili riguardanti il patrimonio familiare), affidato da **<persona qualifica="Granduca" denautor="Cosimo I de' Medici (1519-1574)> Cosimo I </per sona>** stesso al tesoriere Tommaso de' Medici, e che avrebbe preso la via di palazzo Pitti soltanto nella prima metà del XVII secolo, per rimanervi fino al 1773, quando gli atti di cui era composto sarebbero stati divisi e distribuiti tra la Segreteria vecc hia, l'archivio generale dei contratti e l'archivio delle Riformagioni. Altro materiale (soprattutto contratti matrimoniali, testamenti ecc.) sarebbe poi stato trasferito, nella seconda metà dell'Ottocento, nell'attuale fondo dei Trattati Internazionali. L e segreterie trasfentesi presso palazzo Pitti conservavano invece nei propri uffici i documenti correnti e quelli necessari al disbrigo degli affari di loro pertinenza, mentre tutto il materiale non più attuale veniva trasferito, in un flusso quasi contin uo, presso il deposito di palazzo Vecchio, dove ben presto si iniziò ad accusare una carenza di spazio che rischiava di compromettere l'ordine stesso e l'integrità delle carte. Fu così che nel 1592 su istanza di Girolamo Seriacopi, castellano della Fortezz a di Firenze, si destinarono all'archivio dei locali più adeguati, grazie ai quali si poté risolvere l'emergenza, per lo meno momentaneamente. Ben presto, però, anche questo provvedimento dovette risultare insufficiente, se nel 1639 l'allora custode Ugo Ca ciotti, ritenne opportuno proporre al granduca Ferdinando II e al segretario Andrea Cioli, senza a quanto pare portarlo a compimento, un riordinamento che risolvesse la condizione estremamente caotica in cui versavano a suo dire le carte. Nel 1639, tra l'a ltro, parte dell'archivio del Ducato di Urbino era giunta a Firenze come eredità di Vittoria della Rovere ed era stata sistemata anch'essa in palazzo Vecchio.

Un primo deciso intervento di ordinamento dovette essere quello compiuto negli anni '80 del Seic ento dal canonico Fabrizio Cecini, che in una relazione a Cosimo III spiegò di aver operato soprattutto sui carteggi diplomatici, già all'epoca un insieme documentario cospicuo, individuando serie di lettere, ordinate per provenienza e, all'interno, per su ccessione cronologica. Il restante materiale era da lui considerato sostanzialmente di corredo alle raccolte epistolari. Il suo lavoro fu tuttavia vanificato dall'incendio scoppiato a palazzo Vecchio nel 1690 che, se non provocò danni diretti e significati vi alle carte, ne causò un'ulteriore confusione, data la concitazione con cui si dovette intervenire per salvare il materiale dalle fiamme. Oltre a questo grave incidente, a complicare ancora la situazione, contribuì l'aumento continuo delle carte, che arr ivavano senza ordine e spesso avendo già perso la loro integrità originaria. In una lettera al segretario di Stato Carlo Antonio Gondi, per esempio, l'archivista riferiva di aver fatto presente al granduca la necessità di imporre agli ambasciatori di ritor no dalla loro missione di consegnare tutta la documentazione in loro possesso, prima che se ne disfacessero nei modi consueti, tra cui quello a suo giudizio deprecabile di cederla ai pizzicagnoli.
Dopo Cecini, morto nel 1717, durante gli ultimi venti anni del governo mediceo, nessuno si curò più delle condizioni dell'archivio della Segreteria Vecchia, che continuò perciò a subire rimaneggiamenti, dispersioni e alterazioni nella disposizione delle carte ad opera di coloro che attingevano indisturbati ai documenti per scopi non solo amministrativi, ma anche storico -eruditi.

La messa a punto di una tecnologia del genere potrebbe costituire un ulteriore passo verso quello che dovrebbe essere un obiettivo o, piuttosto, una opzione metodologica da perseguire con decisione: l'indipendenza dei dati dai formati e dagli applicativi utilizzati per la loro memorizzazione e quindi l'integrazione o, se vogliamo, l'interscambiabilità fra questi ultimi.

Stefano Vitali

Interrogare collezioni di documenti XML: una interfaccia utente

Introduzione

L'interrogazione di collezioni di documenti XML è un argomento investigato da vari autori[1]. È necessario che gli utenti possano comprendere la semantica dei tag e la struttura dei documenti[2]. In ogni caso, va tenuto presente che la formulazione della query è una operazione complessa, e gli strumenti previsti (XPath e XQuery) non sono particolarmente adatti ad un utente finale.

Questo lavoro[3] descrive un'interfaccia utente in grado di adeguarsi alle caratteristiche dell'utente, utilizzando alcune informazioni semantiche aggiunte allo schema XML che descrive il documento. L'interfaccia assiste l'utente nella fase di preparazione della query, ed è indipendente dallo specifico motore di ricerca utilizzato per interrogare la base documentaria. Nel resto di questo lavoro discuteremo prima alcuni aspetti generali relativi ai requisiti utente. Successivamente, presenteremo le caratteristiche essenziali considerate nella fase di progetttazione dell'interfaccia e di composizione della query, e poi, in maniera più articolata, l'architettura. Infine, vengono forniti alcuni dettagli relativi all'implementazione e al caso di studio. Nelle conclusioni sono anche delineati alcuni possibili scenari evolutivi.

Un doveroso ringraziamento va al prof. Paolo Ferragina del Dipartimento di Informatica dell'Università di Pisa, e ai suoi collaboratori Alessandro Perucci e Rosanna Rossi.

[1] Braga *et al.* 2003; Ceri *et al.* 2000; Deutsch *et al.* 1999; Guha *et al.* 2003; Halevy *et al.* 2003; Hayashi *et al.* 2003; Kotsakis 2003; Miller, Sheth 2000.

[2] Navarro, Baeza-Yates 1997.

[3] Questo lavoro è stato finanziato dal progetto QUESTION-HOW (Quality Engineering Solutions via Tools, Information and Outreach for the New Highly-enriched Offerings from W3C: Evolving the Web in Europe), contratto IST-2000-28767.

Le esigenze degli utenti

Il numero di collezioni di documenti XML va crescendo, e la possibilità di interrogarle in modo efficace diventa un obiettivo importante. Considerato che un documento è costituito essenzialmente da tre componenti: **contenuto, struttura** e **presentazione**, e che XML ha favorito la separazione degli aspetti di presentazione dagli altri, va tenuto presente che l'utente, pur attribuendo una significativa importanza alla presentazione del documento, è comunque particolarmente interessato a ritrovare i documenti pertinenti ai suoi interessi, e quindi a formulare query precise e significative.

```
<books>
    <book number="1">
        <metadata>
            <author>Oreste Signore</author>
            <title>The evolving Web</title>
        </metadata>
        <content>
        <introduction>
                <author>Oreste Signore</author>
            </introduction>
            <part number="1">
                <chapter>
                    <author>Silvia Martelli</author>
                    <author>Oreste Signore</author>
                    <author>Cristian Lucchesi</author>
                    <title>About the future of the Web</title>
                </chapter>
            </part>
            <part number="2">
                <chapter>
                    <author>Oreste Signore</author>
                    <author>Silvia Martelli</author>
                    <author>Marco Andreini</author>
                    <author>Cristian Lucchesi</author>
                    <title>The future of technologies</title>
                </chapter>
            </part>
        </content>
    </book>
</books>
```

Fig. 1. Un esempio di documento XML.

Tutti i sistemi di *information retrieval*, così come molti motori di ricerca, affrontano il problema della ricerca per contenuto, e risolvono la query ricercando la presenza nel documento di specifiche parole. Tuttavia, sono noti alcuni problemi intrinseci in questo approccio. La principale controindicazione è dovuta al fatto che ricercare una parola non significa necessariamente ricercare uno specifico concetto. Infatti, le parole sono semanticamente povere, e sono molti i casi di sinonimie e omografie, che hanno un impatto negativo su **precisione** e **richiamo**. Inoltre, spesso alcune parole hanno un significato diverso a seconda del contesto o parte del documento in cui appaiono. A puro titolo di esempio, il nome di una persona ha un significato diverso se appare nella sezione autore o nella sezione bibliografia o ringraziamenti. Da qui l'importanza di poter formulare le query indicando le specifiche sezioni del documento in cui ricercare i termini.

Spesso è disponibile una descrizione della struttura del documento, ma quasi sempre la semantica del campo è affidata alla comprensione del suo nome, e questo è inaccettabile nel contesto del *World Wide Web*, in cui sono presenti lingue e culture diverse. Inoltre, in presenza di un documento con strutturazione complessa, parte della semantica viene anche dalla struttura, per cui conoscerla e comprenderla diventa essenziale. A titolo di esempio, si consideri una collezione di documenti che descrivono documenti scientifici. In una struttura di documento come quella riportata in fig. 1, il tag <author> contiene l'informazione che *Oreste Signore*, *Silvia Martelli*, *Cristian Lucchesi* e *Marco Andreini* sono autori di qualche capitolo del libro. Tuttavia, il loro ordine può essere il vettore di un'informazione addizionale: *Silvia Martelli* è il primo autore del capitolo *About the future of the Web*. Un utente potrebbe essere interessato a ritrovare i libri che contengono capitoli scritti da *Oreste Signore AND Silvia Martelli*, dove *Silvia Martelli* è il primo autore, e il numero di autori è inferiore a tre.

È forse inutile sottolineare che il tag <author> (ma potrebbe anche essere il tag <a>, o <au> o <x>) non ha, di per sé, nessun particolare significato, mentre si assume implicitamente che esso contenga il nome dell'autore dell'articolo, e che l'utente ne comprenda il significato qualunque sia la sua lingua madre. Per una ricerca realmente efficace, l'utente dovrebbe comprendere appieno la semantica del tag, per esempio sapendo che essa è la stessa del tag <dc:creator> secondo la definizione della Dublin Core initiative[4]. In definitiva, l'utente ha

[4] The Dublin Core Home Page. URL: <http://dublincore.org/>.

bisogno di una *descrizione semantica di ogni tag* e di comprendere chiaramente la *struttura dei documenti* della collezione da ricercare.

Va poi tenuto presente che, per poter formulare query efficaci, individuando con esattezza i concetti da ricercare, è assolutamente necessario che utente e indicizzatore possano *condividere la stessa base di conoscenza*. Questo obiettivo viene normalmente conseguito accedendo a dizionari o thesauri, in modo che l'utente possa verificare l'esattezza dei termini utilizzati e rendersi conto delle relazioni semantiche tra di essi.

Nel contesto del *World Wide Web*, accade di frequente che si effettui una ricerca su più database o collezioni di documenti. È ovvio che collezioni diverse di documenti, anche se dello stesso tipo o molto simili, avranno strutture e nomi dei tag diversi. L'identificazione delle equivalenze semantiche in collezioni eterogenee, e quindi la possibilità di ricercare e collegare documenti presenti in queste collezioni, è un obiettivo rilevante nel panorama del *Semantic Web*[5].

Progettazione dell'interfaccia di interrogazione

Considerazioni di base

L'obiettivo essenziale perseguito dall'interfaccia è supportare l'utente nella formulazione di query in cui vengono espresse condizioni sui valori assunti dai singoli elementi tenendo conto anche della *struttura* del documento. L'interfaccia deve consentire di formulare query *semanticamente corrette*, quindi valide per quanto concerne sia i valori specificati che le condizioni imposte sulla struttura, evitando le difficoltà derivanti dall'utilizzo di linguaggi complessi[6]. È forse pleonastico sottolineare che una query semanticamente corretta può anche restituire un insieme vuoto di documenti, ma solo perché non esiste un sottoinsieme della base documentaria che soddisfi la richiesta, il che è molto diverso dal caso in cui il risultato della ricerca è nullo perché sono state specificate condizioni non accettabili (per esempio, «PY=02» invece di «PY=2002», dove PY è l'anno di pubblicazione, codificato come numero a quattro cifre nella particolare collezione).

[5] <http://www.semanticweb.org/>.

[6] Bonifati, Ceri 2000.

Per raggiungere questo scopo è necessario che l'utente sia posto nella condizione di:

- comprendere la *semantica degli elementi;*
- immettere i *dati corretti* per formulare le condizioni;
- identificare i *concetti* da ricercare.

Per quanto riguarda il primo aspetto, va ricordato che la semantica del singolo elemento non può essere affidata unicamente al nome del tag, spesso criptico o poco significativo. Nella specifica dello XML schema è possibile utilizzare un campo descrittivo, che però non sempre viene riempito dal progettista.

Per immettere i dati corretti nella specifica delle condizioni elementari è necessario avere conoscenza delle liste di *valori ammissibili* e degli eventuali *vincoli* esistenti. Liste di valori ammissibili sono specificabili a livello di XML Schema, anche se appare noioso, ridondante e poco flessibile a fronte di possibili variazioni della lista di valori ammessi. Inoltre, non è possibile specificare vincoli nel caso in cui i valori ammissibili per un certo elemento dipendono dal valore assunto da un altro elemento. Per esempio, in una collezione di documenti eterogenei descritti da un unico schema, come nel caso di libri e atti di conferenze, l'elemento <conferenceLocation>, contenente il *luogo della conferenza* è obbligatorio, o deve assumere valore nullo, a seconda che il tipo di documento (elemento <documentType>) sia *libro* o *atti di conferenza*.

Infine, per alcuni elementi semanticamente pregnanti, come le parole chiave, si rende necessario poter identificare il concetto da ricercare. Sotto questo aspetto, il problema fondamentale è la condivisione della stessa base di conoscenza tra l'indicizzatore e l'utente, e quindi la necessità di poter accedere al thesaurus, o più in generale all'ontologia alla quale fa riferimento lo specifico elemento. Questa esigenza è presente anche nel caso in cui si voglia ricercare in un elemento a testo libero, per il quale potrebbe essere comunque utile indicare uno o più thesauri di riferimento.

Vi sono anche altri aspetti, più strettamente legati alle *modalità di interazione*, in cui l'utente deve essere supportato nella fase di formulazione della query.

Primo fra tutti è l'aspetto del *multilinguismo*, per cui tutti gli elementi descrittivi devono poter essere forniti nella lingua preferita dall'utente. Un aspetto non trascurabile, nel contesto aperto implicitamente sottinteso dal web, è la possibilità di adattarsi alle *diverse culture* dei potenziali utenti. A titolo di esempio, esistono tradizioni diverse per

esprimere le date, e non sempre riesce naturale utilizzare il formato adottato nella specifica collezione.

Un altro elemento distintivo è il livello di competenza dell'utente. Un utente che accede regolarmente ad una collezione conosce bene l'organizzazione e il significato degli elementi, per cui potrebbe preferire una interfaccia meno verbosa, soprattutto nel caso in cui acceda al servizio tramite una connessione lenta o un dispositivo con schermo di limitate dimensioni.

Sarebbe possibile interrogare la collezione utilizzando delle opportune *espressioni XPath*, ma è evidente che non si tratta di una sintassi particolarmente adatta ad un utente finale.

Una esigenza molto sentita è quella di elaborare i documenti restituiti, sia per la tradizionale enfatizzazione dei termini ricercati o per la formattazione del documento, che per arricchire il testo con hyperlink, annotazioni, elementi aggiuntivi derivanti da una analisi del documento.

Composizione della query

Comporre una query utilizzando un'interfaccia grafica è spesso un'esperienza frustrante. Anche se l'utente ha una conoscenza approfondita della struttura dei documenti e della sua semantica, deve passare attraverso una lunga serie di passi, selezionando campi, aprendo finestre a scorrimento, selezionando gli operatori booleani, le parentesi, ecc. E la sensazione, alla fine, è quella di aver impiegato molto tempo per preparare una query elementare, mentre riesce a volte impossibile formulare query realmente complesse. Questo tipo di problemi può assumere anche maggiore consistenza nel caso in cui ci si trovi di fronte a strutture complesse come possono essere quelle dei documenti XML. Considerazioni simili sono espresse anche dal lavoro di D. Braga *et alii*[7], che illustra XQBE, un dialetto visuale di XQuery che implementa un linguaggio visuale di interrogazione basato su alberi, coerentemente con il modello di dati XML.

Noi abbiamo cercato di trovare un modo di comporre la query che fosse del tutto generale, e dipendesse solo dalle funzionalità supportate dal particolare motore di ricerca. Concordiamo pienamente con

[7] Braga *et al.* 2003.

questi ricercatori[8] nel ritenere che un'interfacccia troppo complessa riuscirebbe di difficile utilizzo per l'utente medio, senza peraltro avere un potere espressivo maggiore una query testuale XQuery, e quindi sarebbe sostanzialmente inutile.

Nel nostro approccio la query viene composta interattivamente navigando sulla struttura del documento e interagendo con una *queryForm* predisposta sulla base delle informazioni deducibili dallo schema e dalle informazioni aggiuntive specificate nella sua annotazione semantica.

Nel formulare condizioni su una struttura dati gerarchica esiste il noto problema di normalizzazione della query, ovvero di individuazione del padre comune agli elementi sui quali si impongono delle condizioni. In altri termini, quando si specificano condizioni su rami diversi dell'albero, bisogna anche specificare qual è la radice del sottoalbero rispetto alla quale devono essere verificate le condizioni. Per esempio, interrogando una collezione di documenti contenente record del tipo riportato in fig. 1, una query che imponesse le condizioni

author ="Oreste Signore" AND title="The future of technologies"

deve specificare la radice (<chapter> o <content>) del sottoalbero nel quale le due condizioni devono essere valide.

Per l'utente riesce naturale legare l'individuazione della radice del sottoalbero ad una navigazione nella struttura, durante la quale vengono specificate le condizioni sui singoli elementi.

In base a queste considerazioni, e avendo presente che la composizione di una query non banale, con vari livelli di parentesi, risulta sempre complessa e farraginosa, mentre un utente esperto preferisce formulare le condizioni elementari una volta per tutte, combinandole poi in vario modo con operatori booleani e livelli di parentesi, abbiamo ritenuto opportuno definire la query come la combinazione di vari elementi.

Una query (identificata come q1, ..., q99) è una composizione booleana (con parentesi) di uno o più *queryFragment*, che sono, a loro volta, una composizione booleana, con parentesi, di *queryToken*. Un *queryFragment* viene costruito automaticamente durante una singola navigazione sull'albero del documento, mentre l'utente specifica le condizioni sui singoli elementi. Ogni *queryFragment* è identificato univocamente come: qF1, ..., qF99.

[8] *Ibid*.

Il *queryToken* (identificato come qT0, qT1, ... all'interno del singolo qF) è il componente elementare della query, e viene creato mediante un'interazione con il *queryForm* (fig. 2). Nel caso in cui si vogliano specificare condizioni su più istanze dell'elemento, è necessario iniziare una nuova interazione sullo stesso elemento. Nell'esempio di fig. 1, per ricercare i libri scritti da due autori occorrerebbe specificare due *queryToken*: <author>="X" AND <author>="Y", mentre, se l'elemento <author> contenesse tutti gli autori del libro, occorrerebbe specificare: <author>= ("X" AND "Y"). Quale sia la modalità da utilizzare dipende quindi dalla struttura del documento, che può essere resa nota all'utente sfruttando la metainformazione codificata nell'opportuno elemento della annotazione RDF (*longDescription*).

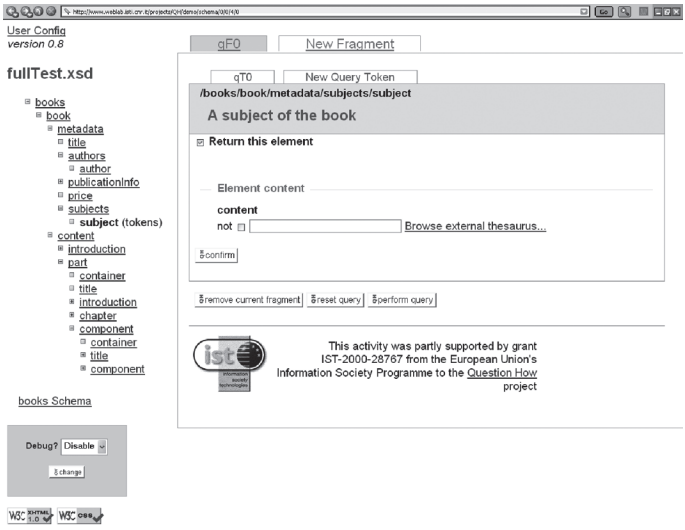


Fig. 2. Compilazione di un query*Token* in un Query *Form* (con selezione della lingua inglese nello *User Profile*).

Negli esempi precedenti abbiamo mostrato solo query elementari, che utilizzavano esclusivamente gli operatori «=» e «AND», mentre l'interfaccia può supportare un'ampia varietà di operatori e operandi. Più esattamente, gli operatori di confronto includono tutti quelli tipici dei sistemi di *information retrieval* (adiacenza, frase, troncamento, mascheramento, ecc.) e possono essere specificati a livello di ogni *queryToken* (fig. 3). Successivamente, come verrà specificato in

dettaglio, l'utente può modificare la query, introducendo un numero arbitrario di operatori booleani e di livelli di parentesi, combinando in vario modo i *queryToken* e i *queryFragment*, per ottenere query anche molto complesse.

La motivazione di base di questa scelta è stata supportare l'utente nella formulazione dei componenti elementari della query, fornendo adeguato supporto mediante informazioni semantiche e descrizione della struttura dell'albero, lasciandogli poi la libertà di combinarli per creare query sofisticate. Si presume, infatti, che un utente che intenda formulare query complesse non abbia difficoltà nell'utilizzare i connettori booleani e le parentesi.

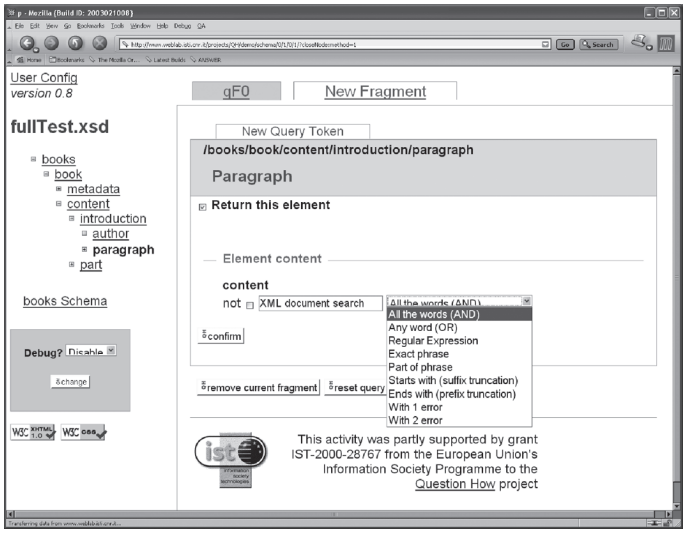


Fig. 3. Scelta di operatori su testo libero nella compilazione di un query*Token*.

L'architettura

Descrizione generale

Dal punto di vista meramente architetturale, due condizioni sono state ritenute particolarmente stringenti e significative:

- **non modificare** la collezione di documenti, lasciando aperta la possibilità di interagire con una qualunque collezione di documenti XML, senza doverne modificare lo schema, che viene semplicemente corredato di una annotazione esterna;

- definire un'**architettura aperta e distribuita**, coerente con i principi di estensibilità, interoperabilità e decentralizzazione che sono alla base di ogni applicazione web. In alcuni casi, può risultare necessario invocare dei *web services*, identificati da URI.

La fig. 4 descrive sommariamente l'architettura. In questa figura il motore di ricerca e il Document Server appaiono risiedere sulla stessa macchina, come spesso accade. Tuttavia, questo non è un vincolo imposto dall'interfaccia, che si limita ad avere conoscenza della localizzazione del motore di ricerca.

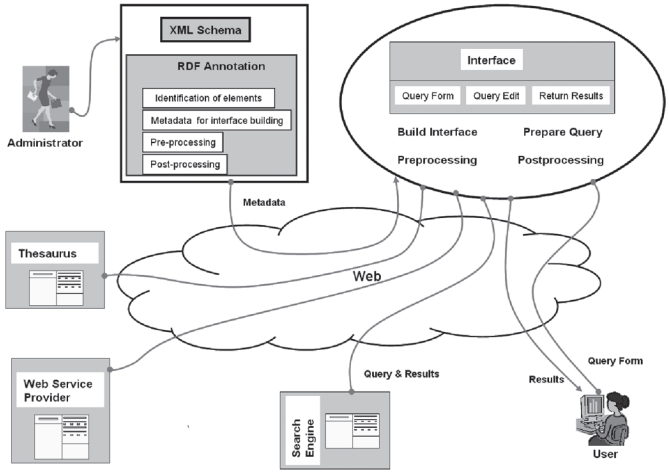


Fig. 4. Una visione generale dell'architettura concettuale.

È opportuno sottolineare che è appunto l'architettura, completamente distribuita, che differenzia questa interfaccia da tante altre applicazioni simili, basate sul riempimento di campi predefiniti. La caratteristica chiave è che i singoli componenti possono risiedere su qualunque macchina raggiungibile in Internet, e le risorse utilizzate dall'interfaccia sono identificate univocamente dai loro URI.

Funzionalmente, possiamo distinguere tre livelli:
- *amministratore* della collezione,
- *utente*,
- motore di ricerca e *document server*.

Nel seguito descriveremo brevemente questi tre livelli[9].

Amministrazione della collezione

Per definire un'interfaccia utente efficace, è necessario descrivere le caratteristiche dell'utente e quelle dei dati. Le caratteristiche dell'utente, necessarie per consentire all'interfaccia di adeguarsi alle sue preferenze, possono essere descritte definendo un opportuno *User Profile*, quelle degli elementi possono invece essere descritte dagli *Element Metadata*.

L'amministratore della collezione di documenti, che si suppone abbia una conoscenza approfondita sia della collezione che del dominio di conoscenza relativo, ha il compito di *gestire* la collezione di documenti, utilizzando il motore di indicizzazione che preferisce; *definire*, ove questo non esistesse, lo XMLSchema della collezione; *annotare esternamente lo schema*, utilizzando RDF. I metadati descritti nell'annotazione RDF possono essere importati nel sistema, e memorizzati nei suoi oggetti interni. Una componente del sistema permette di aggiungere i metadati agli oggetti del sistema, che li può quindi esportare sotto forma di annotazione RDF. In questo modo si può creare l'annotazione RDF senza dover utilizzare un editor *ad hoc*. L'annotazione RDF permette, tra l'altro, di definire le modalità tipiche di interazione per i vari *tipi di utente* e *formalizzare la metainformazione*.

Per quanto concerne il primo aspetto, è possibile definire alcune caratteristiche dell'utente, che possono influire sulle modalità di interazione: il *livello* (neofita o esperto), la *lingua* preferita per l'interazione, il *tipo di informazione* restituita (record completo, parte di esso, ecc.).

Si noti, riguardo all'effetto delle caratteristiche specificate nel profilo utente, come l'informazione in figg. 2 e 7 sia in due lingue

[9] Per una descrizione più dettagliata cfr. <http://www.weblab.isti.cnr.it/projects/QH/docs/deliverable.html>.

diverse. Il profilo utente è modificabile durante la sessione, con effetto immediato.

Per quanto riguarda la presentazione dei risultati, ogni elemento del documento è assegnato ad un formato record (*record format*). Il *recordFormat full* include il *recordFormat long*, che, a sua volta, include il *recordFormat short*, a cui deve essere assegnato almeno un elemento.

Nell'implementazione attuale, lo *User Profile* è stato mantenuto particolarmente semplice, lasciando comunque aperta la possibilità di ulteriori arricchimenti.

L'aspetto più significativo è tuttavia quello dei metadati relativi ai singoli elementi. Alcune informazioni possono essere desunte direttamente dallo schema (es. il tipo, la molteplicità, la lista o l'intervallo di valori ammessi, ecc.). Tuttavia, non sempre lo schema contiene tutte queste informazioni, e ve ne sono alcune che non possono essere espresse (riferimento ad ontologie, dipendenze dei valori di elementi). Di conseguenza, sono state identificate alcune proprietà, che possono essere suddivise, per comodità, in vari gruppi:

- **Identificazione** degli elementi: per poter identificare univocamente gli elementi o attributi nel contesto del documento
- **Proprietà** degli elementi:
  - descrizione (per esprimere più chiaramente il loro significato e supportare lingue diverse)
  - ricercabilità (per individuare gli elementi che hanno una valenza stilistica, ma non semantica)
  - ruolo (nodi puramente strutturali o con contenuto)
- **Vincoli**
  - thesaurus, dizionari, liste esterne, liste locali
  - *web service* da invocare
  - dipendenze tra i valori assunti
- **Opzioni di *pre-processing*** (per esempio per supportare formati data più adeguati alla cultura dell'utente)
- **Opzioni di *post-processing*** per manipolare e arricchire i documenti restituiti.

L'informazione contenuta nei metadata viene quindi utilizzata sia per personalizzare l'interfaccia (proprietà e descrizioni degli elementi, pre e post elaborazione) che per supportare l'interazione (vincoli, ele-

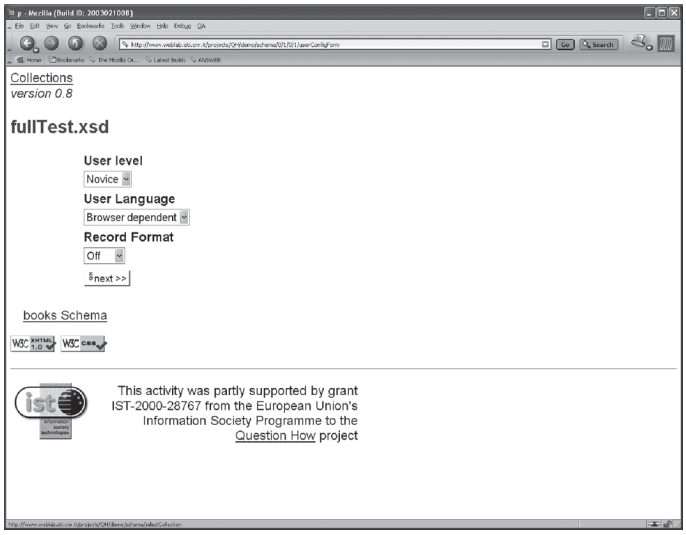


Fig. 5. Impostazione del profilo utente.

menti contenenti testo controllato). Al fine di evitare una eccessiva complessità nella formulazione dei vincoli in RDF, si è ritenuto opportuno limitarsi a descriverli in linguaggio naturale. Nelle prossime versioni verrà considerata la possibilità di utilizzare espressioni formali per definire in modo dichiarativo vincoli abbastanza semplici e di tipo frequente (per esempio, vincoli di esistenza, elementi i cui valori devono essere in ben precise relazioni di equivalenza, ecc.). Per gli elementi il cui contenuto è disciplinato da liste di valori, dizionari o thesauri, è già possibile avere l'invocazione, automatica o a seguito di esplicita richiesta utente, di *web services* o applicazioni CGI, a seconda di quanto specificato nella annotazione RDF (fig. 2).

Nella fase di definizione delle caratteristiche dell'interfacccia, è stata anche considerata la possibilità di espansione automatica dei termini della query, ma è stata ritenuta in contrasto con la filosofia di base. Infatti, specialmente in alcuni contesti come i beni culturali, l'espansione automatica può comportare l'inclusione di un gran numero di termini indesiderati, determinando una perdita di precisione, con il reperimento di molti 'falsi positivi'. Abbiamo ritenuto di gran lunga preferibile offrire all'utente la possibilità di navigare nell'albero dei concetti, percorrendo le relazioni opportune e individuando i termini più adeguati per formulare la query.

Il livello utente: l'interfaccia

L'interfaccia, creata automaticamente sulla base delle informazioni gestite dall'amministratore della collezione, supporta l'utente nella

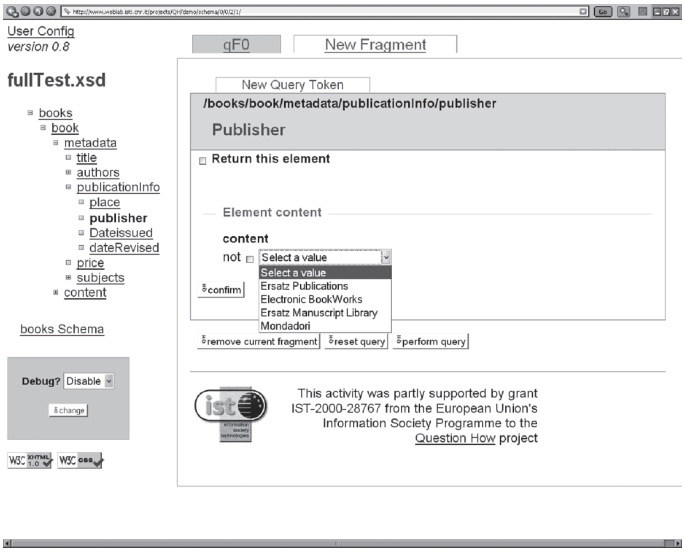


Fig. 6. Selezione di un valore da una lista ottenuta mediante invocazione di un *web service*.

formulazione della query, assicurandone la correttezza formale e consentendo di utilizzare tutte le caratteristiche del motore di ricerca sottostante. Il sistema, infatti, conosce perfettamente la semantica dei vari elementi e i vincoli esistenti, ed è in grado di esplicitarli all'utente e di verificarli.

L'utente può *navigare sulla struttura* del documento e formulare la query, disponendo di tutte le informazioni relative alla *semantica* dell'elemento e ai *vincoli* esistenti.

La query viene preparata in *un formato generico*, che può successivamente essere trasformato nel formato previsto dal *search engine* sottostante.

Il sistema utilizza i metadati per generare una *query form*, che è corredata di tutte le informazioni necessarie allo *user agent* (il browser in questa prima implementazione) per la generazione dell'interfac-

cia utente. Quest'ultima è l'elemento che implementa l'interazione uomo-macchina, il che implica, tra l'altro:

- presentare la *query form*;
- mostrare la struttura del documento;
- gestire l'interazione con altre fonti di informazione (dizionari, thesauri, ontologie, ecc.);
- comporre la query in un formato adeguato per la successiva elaborazione da parte del motore di ricerca sottostante. In questa implementazione supportiamo un sottoinsieme del linguaggio XQuery, con alcune caratteristiche disponibili nei sistemi di *information retrieval* (prossimità, troncamento, espressioni regolari, ecc.);
- modificare e inviare la query;
- mostrare i documenti restituiti a fronte della ricerca.

Fig. 7. Fase di *editing* della query (con selezione della lingua italiana nello *User Profile*).

Il primo passo nell'utilizzo del sistema è la selezione della collezione di documenti e dello *User Profile*. Successivamente (fig. 5) l'utente può consultare, sulla sinistra dello schermo, l'albero del documento, i cui nodi possono essere compressi o espansi a seconda delle necessità. Per ogni nodo, un click del mouse attiva, nella parte destra dello schemo, la *queryForm*, in cui l'utente può immettere i valori e gli operandi

desiderati, costruendo così un *queryToken*. Nella composisione del *queryToken* l'utente può selezionare o deselezionare l'elemento come elemento da restituire nella risposta, e può invocare il supporto di un dizionario o di un thesaurus (fig. 2). In alcuni casi, le liste esterne di valori ammessi vengono ottenute automaticamente mediante un *web service* (fig. 6).

Tutti i *queryToken* compilati durante una singola navigazione sull'albero vengono posti automaticamente in AND booleano e costitui-

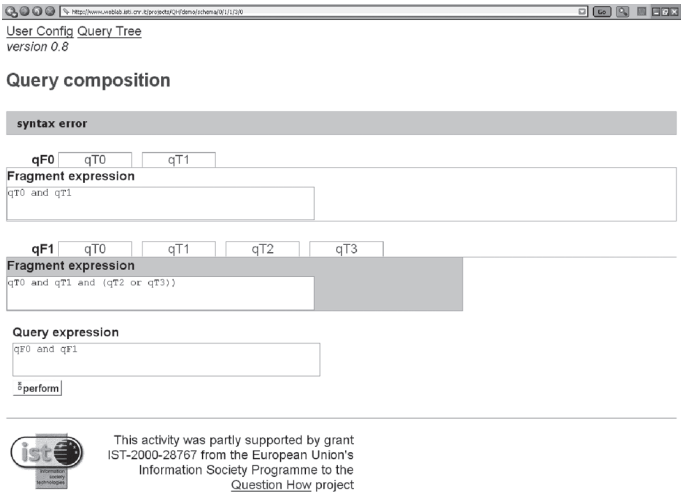


Fig. 8. Verifica della correttezza sintattica della query.

scono un *queryFragment*, che, a sua volta, può essere combinato con altri *queryFragment* per creare una query complessa. Un *queryFragment* viene creato quando l'utente seleziona l'opzione *newFragment* (dando così inizio a una nuova navigazione sull'albero). I vari *queryFragment* vengono posti automaticamente in AND booleano nel momento in cui, a seguito della selezione dell'opzione *buildQuery*, viene generata una query. A questo punto, l'utente entra in un ambiente di *editing* della query, nel quale può modificare i *queryFragment* e la query, aggiungendo parentesi o modificando gli operatori booleani, per combinare diversamente i vari *queryToken* (fig. 7). Sulla query modi-

ficata viene eseguito un controllo sintattico (fig. 8). Per modificare un *queryToken*, è necessario ritornare su di esso e interagire mediante il *queryForm*.

Quando l'utente ritiene di aver formulato la query rispondente alle sue esigenze, può richiederne l'esecuzione. Il sistema provvede poi a presentare il documento XML restituito dal motore di ricerca. In questa fase (*post-processing*), sono previste elaborazioni del risultato, non ancora implementate nella presente versione.

Il livello del Document Server

Come già detto, l'interfaccia è del tutto indipendente dal *search engine* e dall'architettura del *document server*. L'interfaccia deve solo sapere quali sono gli operandi e gli operatori supportati dal motore di ricerca, per poter presentare all'utente le varie opzioni accettabili.

Le funzioni principali da svolgere a livello del *document server* sono la conversione della query preparata dall'interfaccia nel formato richiesto dal *search engine*, l'esecuzione della query, e la restituzione dei risultati.

Dettagli implementativi

Il search engine: XCDE

Nell'implementazione corrente, è stato adottato come *search engine* la libreria XCDE, sviluppata presso il Dipartimento di Informatica dell'Università di Pisa (prof. Paolo Ferragina, gruppo algoritmi). Nell'ambito di questo progetto, la libreria è stata estesa allo scopo di supportare un *query language* sofisticato che combina alcune caratteristiche di XQuery[10] con altre tipiche dei sistemi di *information retrieval*. Questa direzione di ricerca è stata suggerita recentemente anche in ambito W3C[11].

Si tratta di uno dei primi tentativi di implementare questa proposta utilizzando indici compressi su documenti XML. Il *query language*

[10] XQuery 1.0: An XML Query Language – W3C Working Draft 22 August 2003, <http://www.w3.org/TR/xquery/>.

[11] XQuery and XPath Full-Text Requirements – W3C Working Draft 02 May 2003, <http://www.w3.org/TR/xquery-full-text-requirements/>.

definito consente interrogazioni testuali sofisticate su documenti XML: prossimità di parole, ricerca con espressioni regolari o con corrispondenza parziale, estrazione di *snippet* per selezionare il contesto di un documento reperito, ricerca per sottostringhe, troncamento di prefissi e suffissi, ecc.

La libreria, scritta in C, è utilizzabile liberamente per scopi non a fini di lucro.

La libreria XCDE. – La libreria fornisce un insieme di algoritmi e strutture dati per indicizzare e ricercare collezioni di documenti XML. I documenti devono essere *well-formed* e possono essere eterogenei, facendo riferimento a DTD o XML Schema diversi. La libreria supporta l'archiviazione e la gestione dei file XML in forma nativa, operando direttamente a livello di *file system*.

La principale struttura dati di indicizzazione è la classica lista invertita, in cui vengono utilizzate opportune tecniche di compressione. Vengono indicizzati indipendentemente le parole, i tag, gli attributi e i loro valori, consentendo così l'esecuzione efficiente di query sofisticate. Le query che coinvolgono la struttura del documento possono essere eseguite in modo efficiente grazie ad un nuovo metodo di indicizzazione della struttura gerarchica, basato su strutture dati geometriche.

Il testo originale viene compresso in modo che l'accesso ad alcune parti di esso possa essere eseguito senza decomprimere l'intero file. Questa caratteristica si rivela particolarmente utile per presentare rapidamente il risultato di una query (detto normalmente *snippet*). Complessivamente, il documento XML compresso con tutti i suoi indici ha una dimensione all'incirca pari a quella del documento originale. Questa caratteristica rende XCDE nettamente migliore di tutti gli altri approcci conosciuti, e può costituire un elemento importantissimo per applicazioni su apparecchiature con memoria limitata.

Una delle scelte di progetto in XCDE è stata quella di operare a livello di granularità del singolo documento. Lo svantaggio è che le ricerche su collezioni costituite da molti documenti di dimensioni ridotte possono risultare lunghe. D'altra parte, vi sono alcuni vantaggi, quali la realizzazione di indici distribuiti, la facilità di aggiornamento dell'indice a fronte di inserimenti o aggiornamenti dei documenti, la possibilità di gestire agevolmente documenti XML eterogenei e di personalizzare gli indici sulla base delle caratteristiche del documento. È comunque possibile indicizzare collezioni di documenti abbastanza vaste senza pagare in prestazioni con due accorgimenti: o con un *docu-*

ment merging, combinando più documenti in un unico documento più ampio, o aggiungendo un *indice leggero* per filtrare parte della collezione di documenti prima di eseguire la query sui singoli documenti.

La libreria fornisce una API con un ricco insieme di funzioni C con cui è possibile implementare la maggior parte delle funzionalità di base di XQuery, e supportare query più complesse di tipo IR. L'obiettivo è la gestione efficace ed efficiente di documenti caratterizzati da una rilevante porzione di testo libero e/o struttura gerarchica profonda, in cui i valori degli attributi sono sequenze complicate di numeri e lettere. Questo è lo scenario tipico che si incontra nella gestione di archivi di testi letterari con tag XML-TEI, una delle applicazioni che hanno portato alle definizione della libreria. In questo contesto, la granularità a livello di singolo documento non costituisce un problema.

XCDE Query Syntax. – La sintassi supportata da XCDE per la formulazione della query è simile alla classica sintassi *SQL: SELECT-FROM-RETURN*.

L'*output* della query (lo *snippet*) può essere formattato nella clausola Return, che, a sua volta, include una porzione di XML *well-formed text*.

Un attributo particolare, denominato xml_var (*pivot*) viene aggiunto nella clausola SELECT per identificare i sottoalberi che soddisfano la query.

Ambiente software

L'ambiente applicativo in cui è stato sviluppato il progetto è *open source software* (Zope, Python, Soaplib Python, parsedXML). Postgresql è utilizzato per la memorizzazione dei dizionari e thesauri. La versione attuale dipende da parsedXML, in quanto utilizza le sue funzioni DOM per accedere a XML schema. La prossima versione utilizzerà invece le funzioni SAX incluse in Python.

Tra i componenti completamente realizzati fino a questo punto sono:

- la selezione dello *User Profile* e della *Document Collection*;
- un *parser* per XMLSchema (con alcune limitazioni);
- un modulo che trasforma un XMLSchema in un albero XHTML, consentendo la navigazione sull'albero che rappresenta il documento;
- l'interfaccia di amministrazione per memorizzare gli *Element Metadata*;
- un *form builder* basato su XMLSchema e *Element Metadata*;
- un componente modulare per assemblare la query, editare i

queryFragment e la query finale, tradurre la query nel formato richiesto da XCDE;
- il *wrapper* per chiamare le funzioni di libreria di XCDE e recuperare i risultati restituiti;
- un *web service* per verificare i termini consultando un dizionario;
- una applicazione di test per la navigazione su un thesaurus, che consente di selezionare uno o più valori da inserire nella query.

Il caso di test

Come campione è stata utilizzata la collezione di documenti a cui fa riferimento il *W3C Working Draft*[12].

Su questa collezione è stata predisposta una annotazione RDF[13] dello XMLSchema[14] in cui sono stati specificati vincoli di:
- lista locale
- lista esterna
- thesaurus
- elementi correlati.

Informazioni più dettagliate sono reperibili nel sito Internet[15], dove è anche disponibile una demo.

Conclusioni e sviluppi futuri

È stato definito un contesto consistente per l'accesso intelligente e controllato a collezioni di documenti XML.

Lo schema della collezione non viene modificato, ma annotato esternamente utilizzando RDF.

L'amministratore della collezione può fare riferimento a una qualunque fonte di conoscenza esterna, invocando i thesauri di supporto specificando l'opportuno *web service* (se esiste) o con un tradizionale accesso CGI. Le modalità di invocazione del servizio o della partico-

[12] XQuery and XPath Full-Text Requirements – W3C Working Draft 02 May 2003, <http://www.w3.org/TR/xquery-full-text-requirements/>.

[13] <http://www.weblab.isti.cnr.it/projects/QH/docs/fullText.rdf >.

[14] <http://www.weblab.isti.cnr.it/projects/QH/docs/fullText.xsd>.

[15] <http://www.weblab.isti.cnr.it/projects/QH/docs/>.

lare applicazione vengono specificate su una opportuna risorsa XML, identificata da un URI.

L'interfaccia è generale, e può essere utilizzata verso una qualunque collezione di documenti descritta da un appropriato XMLSchema, e adattata a qualunque motore di ricerca. Al momento, tuttavia, non è prevista la specifica delle funzioni e degli operatori di ricerca supportati dal motore di ricerca sottostante. Alcune funzioni non sono state ancora implementate, ma non presentano difficoltà concettuali.

Come sviluppi futuri, sono in fase di considerazione la possibilità di utilizzare questa interfaccia per aggiornare documenti XML e la migrazione verso un'architettura più orientata all'elaborazione su client, sfruttando le possibilità offerte da tecnologie come XForms.

Un'ulteriore linea di sviluppo, molto interessante, è quella di accedere collezioni di documenti eterogenei. Quest'ultima linea di ricerca prevede di poter rappresentare e comprendere la semantica della struttura dei documenti, problema non banale.

Infine, è opportuno sottolineare che l'architettura è assolutamente coerente con gli standard web e i principi del *Semantic Web*. In particolare, l'unico onere che viene imposto al detentore della collezione di documenti, che la voglia rendere disponibile e ricercabile, è definire uno Schema XML e fornire un'adeguata descrizione semantica degli elementi e degli attributi, requisito essenziale nel contesto del *Semantic Web*.

Oreste Signore, Marco Andreini
Cristian Lucchesi, Silvia Martelli

Guida SignumETexts di base: una proposta per documenti conformi allo standard XML

Una prefazione necessaria

Alla data del convegno *XML per i beni culturali. Esperienze e prospettive per il trattamento di dati strutturati e semistrutturati* (Pisa, 25 marzo 2004) era in corso presso il Centro di ricerche informatiche per i beni culturali un importante cambiamento strutturale, che solo oggi – alla data di stampa degli *Atti* – può dirsi completato. Indirizzi e prospettive di studio di allora, e che a quel convegno furono testimoniati da interventi di ricercatori del Centro, hanno sorretto idee e contributi scientifici che ora vediamo realizzati, quanto meno nelle linee essenziali.

Questo intervento costituisce perciò il ponte necessario fra la situazione fotografata al marzo 2004 e quella di oggi, fra quanto si è proposto allora e quanto possiamo scrivere attualmente sullo stesso argomento: insomma, spiega e rende conto di una storia articolata e complessa.

Gli ultimi due anni del Centro sono stati caratterizzati da un consistente mutamento strutturale, da un avvicendamento del personale, dal contributo di nuove e diverse competenze: il Centro ha ampliato e mutato le proprie prospettive di ricerca e di lavoro, orientandosi su un versante ancora più specificatamente tecnologico. Tale diversa realtà strutturale, scientifica e tecnologica ha portato con sé, come effetto, un secondo naturale cambiamento: Signum è il nome che identifica oggi il Centro di ricerche informatiche per le discipline umanistiche nato dal 'vecchio' Centro di ricerche informatiche per i beni culturali.

Scegliendo di studiare, di progettare e di sviluppare soluzioni per l'analisi, l'archiviazione, la catalogazione e la ricerca di sequenze (nell'ambito della ricerca di base e nell'ambito della sperimentazione tecnologica e dello sviluppo di sistemi applicativi), Signum si propone in sostanza come un centro avanzato di studi informatici applicati a

diverse aree scientifiche. Limitatamente all'area testuale, questo si è tradotto in un rinnovato sforzo concettuale e in una nuova attenzione allo studio delle metodologie e dei linguaggi negli standard di rappresentazione, della codifica testuale e del *text retrieval*.

Facciamo un passo indietro. All'epoca del convegno, la biblioteca virtuale on-line *BiViO* proponeva un certo numero di testi e di documenti della cultura rinascimentale ordinati per autore e secondo alcuni percorsi tematici. *Gli astri e l'uomo*, *Colori*, *Dialettica e Retorica*, *Iconologia*, *Immagini del macrocosmo*, *Immagini del microcosmo*, *Magia ed Ermetismo*, *Pietre*, *Traduzioni* e *Varia* costituivano per così dire gli scaffali di quella biblioteca. Rispetto ad allora, parallelamente ad una riorganizzazione del materiale pregresso, accanto a quegli scaffali ne sono stati aperti – se ne stanno aprendo – altri: *Lessici*, *Enciclopedismo*, *Biografia*, *Fisiognomica*, *Storia*, *Religione*, *Arte della guerra*, *Pedagogia*. Non solo, ampliamento e specializzazione delle aree di ricerca hanno portato alla definizione di nuove diverse biblioteche: *La Bibbia nel Cinquecento*, *Il Vocabolario filosofico*, *La biblioteca delle fonti storico-artistiche* sono i progetti che si stanno affiancando a quelli già esecutivi, e che hanno portato con sé – inevitabilmente – problemi ed esigenze legate a specifiche tipologie testuali (sono testi eterogenei facenti parte, sia pure con modalità diverse, di una stessa collezione virtuale: tutti ugualmente ricercabili mediante un adeguato motore di ricerca).

Decidendo di ripensare il ruolo e gli obiettivi del Centro, punti fermi e qualificanti nella determinazione dell'attività di ricerca sono stati un adeguamento ancora più rigoroso agli standard internazionali e la divulgazione dei risultati teorici conseguiti mediante conferenze e pubblicazioni su riviste specializzate. Se gli interventi dei ricercatori del Centro al convegno del 2004 già si muovevano sulla linea di questo duplice impegno, le pagine che seguono concretizzano in modo compiuto il processo che da allora ad oggi è andato maturando sul fronte della ricerca testuale e delle sue applicazioni. Redatta dal gruppo di lavoro che si occupa di codifica testuale nei suoi aspetti sia teorici che pratici (Marzia Bonfanti, Francesca Dell'Omodarme, Serena Ferretti, Matteo Gallo, Daniele Marotta), la *Guida SignumETexts di base* costituisce insomma il punto di arrivo di una linea di ricerca che ha caratterizzato il Centro di ricerche informatiche per i beni culturali fin dalla sua nascita, ma risponde anche adeguatamente al nuovo profilo scientifico che oggi contraddistingue Signum.

Sono ormai molti anni che, in parallelo allo studio delle problematiche legate al trattamento dei testi, l'attività di sperimentazione informatica intrapresa presso la Scuola Normale Superiore fornisce soluzioni per l'analisi delle fonti letterarie. In un primo tempo è stato risposto con DBT all'esigenza di disporre di strumenti informatici raffinati, in grado di condurre ricerche lessicali sui testi; i suoi limiti (una codifica statica e proprietaria, una possibilità di ricerca circoscritta) hanno poi naturalmente promosso lo studio sia di un software alternativo, che utilizzasse SGML per la codifica testuale, sia – dal punto di vista informatico – di un nuovo motore di ricerca.

Nel 1998 il Centro di ricerche informatiche per i beni culturali si è segnalato per la progettazione e la realizzazione – in collaborazione con il Dipartimento di Informatica dell'Università degli Studi di Pisa – di TReSy (acronimo per *text retrieval system*), un motore di ricerca testuale per documenti SGML/XML. Il progetto, nato con lo scopo di fornire una metodologia efficace e innovativa per il recupero di informazioni testuali con una certa connotazione strutturale, nel tempo si è evoluto: a una prima versione, che gestiva SGML e aveva un'indicizzazione basata sul carattere, ne è seguita un'altra con potenzialità superiori (estensioni per la gestione di parole); attualmente si sta lavorando alla messa a punto di un linguaggio di interrogazione ancora più efficace, che caratterizzerà la terza versione, di prossima uscita.

Utilizzato in molte applicazioni (anche per la realizzazione di siti Internet e di CD-Rom) e nella gestione di grandi volumi di testi strutturati, TReSy è insomma andato perfezionandosi su due fronti distinti, il potenziamento della struttura dati sui marcatori e la messa a punto di un *query language* che permetta di sfruttare in modo semplice e completo le potenzialità messe a disposizione dal motore.

Parallelamente alla sperimentazione informatica, lo sviluppo cui il Centro è andato incontro negli anni successivi al 1998 ha portato ad ampliamenti importanti e forse non prevedibili degli ambiti di ricerca e delle tipologie testuali trattate. Se agli inizi i materiali erano orientati in prevalenza alla storia dell'arte, il settembre 2002 registra un importante cambio di rotta. Dalla collaborazione con l'Istituto di Studi sul Rinascimento nasce il progetto *BiViO*: intento, promuovere ricerche storiche, filosofiche, storico-artistiche e filologiche per la costituzione di una biblioteca on-line di testi (spesso rari, sempre nelle edizioni e nelle traduzioni più significative) codificati in XML-TEI e resi consultabili tramite adeguati sistemi informatici.

È forse superfluo sottolineare in questa sede che per affrontare in modo realmente efficace il problema del trattamento informatico di un singolo testo o di una collezione di testi è necessario analizzare accuratamente il materiale che si intende elaborare. Poiché gli strumenti informatici tradizionalmente adottati non sono in grado di garantire da soli il recupero di tutto il complesso insieme di informazioni che configurano un documento, è necessario che si intervenga preliminarmente sul testo, per evidenziare quali parti e quali fenomeni siano rilevanti e apprezzabili ai fini della ricerca. Non esistono criteri univoci o assoluti per stabilire a priori cosa è importante o significativo: si individua nel tempo, volta per volta, sulla base delle esperienze pregresse, della tipologia del documento oggetto di studio e secondo i dati che dal testo si intende estrarre.

Per rispondere adeguatamente alle nuove esigenze legate al progetto *BiViO*, di respiro più ampio e più diversificato rispetto alle applicazioni precedenti, è stato redatto il *CRiBeCuCorpus*, come venne chiamato allora forse con un certo margine di improprietà: scopo del documento, identificare e definire lo standard di marcatura adottato dal Centro per tutte le proprie fonti letterarie, effettuando una scelta all'interno delle molte possibilità offerte dalla TEI e orientando la marcatura allo standard XML.

Nei pochi anni che ci dividono dal 2002, in un contesto nazionale e internazionale che si è caratterizzato per una crescente attenzione teorica e metodologica al tema delle biblioteche digitali, sono maturati o nati presso il Centro nuovi progetti di biblioteche tematiche. Accanto a *BiViO* stiamo ora realizzando, nella sola area testi, scaffali a tema rappresentati da *La Bibbia nel Cinquecento*, *Il Vocabolario filosofico*, *La biblioteca delle fonti storico-artistiche*.

Lo studio di strumenti informatici sempre più raffinati, capaci di gestire ricerche sul testo e sulla struttura dei documenti, si è così incontrato con nuove e articolate possibilità di applicazione nelle aree di ricerca di Signum. Sono state allora le molteplici esigenze di una biblioteca sempre più variegata e impegnata su fronti talora fra loro lontani a convincerci dell'opportunità di stendere un nuovo documento, e di pensarlo sia sulla base dei casi concreti già trattati sia sulla base dei nuovi progetti in cantiere.

In termini molto generali, possiamo dire che la *Guida SignumETexts di base*, di cui presentiamo una breve descrizione nelle pagine seguenti, si muove sulla falsariga del precedente *CRiBeCuCorpus* (come quello, rappresenta una selezione di elementi tratti dallo schema di

codifica TEI[1], ed espone le norme seguite per la trascrizione e la codifica dei testi nell'ambito dei progetti di digitalizzazione di Signum, al fine di creare documenti conformi allo standard XML). Ma di quel documento rappresenta soprattutto un ampliamento ed un approfondimento: più rigoroso nell'adeguamento agli standard internazionali della codifica dichiarativa-strutturale, e più attento nella scelta dei marcatori che, immessi nel testo, servono a descrivere la funzione logica dei diversi blocchi cui si riferiscono.

Possiamo, in termini più concreti, parlare di una sostanziale continuità a livello dei fenomeni strutturali analizzati, unita a qualche cambiamento nell'utilizzo di alcuni marcatori. Parallelamente, in alternativa ad un impiego completo e incondizionato del modello strutturale offerto dalle linee guida TEI, abbiamo operato alcune scelte che non ricalcano esattamente lo standard generale e rispondono meglio alle nostre esigenze di trattamento dei testi. Due soli esempi. Abbiamo scelto di rinunciare all'articolazione del testo nelle tre parti (*front, body, back*) tradizionalmente consigliate per la sua macrostruttura, perché questa griglia si adatta con evidenti difficoltà ad alcuni testi – come quelli del Cinque-Seicento – che alternano e ripetono al loro interno parti introduttive, prefazioni e dedicatorie. Nel trattamento di eventuali integrazioni ed aggiunte – diversamente da quanto suggeriscono le linee guida della TEI – equipariamo le figure dell'autore e dell'eventuale curatore, segnalando invece con un diverso elemento gli interventi del codificatore.

Quanto il 'vecchio' manuale non prevedeva è invece un'articolazione in due successivi (e complementari) livelli di codifica, che abbiamo introdotto sia per la necessità di ottenere database comunque uniformi, sia per l'esigenza di garantire flessibilità ai diversi progetti. I criteri che illustriamo nelle pagine seguenti sono insomma validi per tutti i progetti che partecipano a Signum e rappresentano il trattamento di base al quale ogni testo, indipendentemente dal progetto di cui fa parte, deve essere sottoposto. Nel secondo livello di codifica, che è legato ai singoli progetti, la marcatura si evolve e si differenzia: approfondisce aspetti e caratteristiche dei testi che al singolo progetto afferiscono, ed è pensata sia in relazione alle peculiarità dei testi raccolti in quel particolare 'scaffale' sia per una loro specifica fruizione.

[1] <http://www.tei-c.org/P4X/> (13/06/2005).

Ma andiamo per ordine. È necessaria una premessa: come nel documento precedentemente elaborato – di cui questo rappresenta uno sviluppo naturale e obbligato – la gestione informatica dei file .xml resta per noi finalizzata ad una fruizione tramite pagine web e ad una ricerca testuale complessa tramite TReSy. In pratica, ciò significa che l'adozione di alcuni elementi per marcare fenomeni rilevanti risulta determinata anche da tali obiettivi. Detto questo, nelle indicazioni del trattamento di base dei testi – dunque al primo livello – ci siamo concentrati sulla modellizzazione della testata elettronica, sulla codifica della struttura e su alcuni singoli aspetti semantici.

Nell'organizzazione del materiale della *Guida SignumETexts di base* la segnalazione dei principi editoriali da noi adottati precede le sezioni dedicate alla struttura del documento e alla sua codifica, e viene poi adeguatamente rappresentata nella <encodingDesc>, quella sezione del <teiHeader> che specifica i metodi e i principi che sovrintendono alla trascrizione elettronica della fonte.

In generale, nell'acquisizione dei testi eleggiamo come regola guida la fedeltà all'opera secondo l'edizione di riferimento. L'esigenza di non perdere di vista tale edizione ci porta a privilegiare determinati elementi di codifica piuttosto di altri (secondo tale edizione manteniamo per esempio i segni meno che indicano la parola spezzata per fine riga, numeriamo le pagine e le linee tipografiche, rispettiamo l'eventuale distribuzione del testo o di una sua porzione su colonne o tabelle), anche se poi, per una migliore e più agile visualizzazione online del testo, introduciamo alcuni elementi di formattazione che ne sono indipendenti (inseriamo una spaziatura dopo i titoli, come anche prima e dopo i passi citati, se centrati nella pagina) e ad altri rinunciamo (per esempio, agli stili tipografici). Un esempio rovesciato di tale regola di base: in <tagsDecl> esplicitiamo che l'elemento <head>, quando accompagnato dall'attributo *type* con valore «indexEntry» e «runningTitle», può contenere del testo non presente sulla fonte e che viene aggiunto dal codificatore per esclusive finalità informatiche (ad esempio, per costruire l'indice strutturale dell'opera). Nell'atto pratico della codifica, questo si traduce nell'adozione di parentesi quadre: usando questo segno diacritico il codificatore dichiara, distingue e circoscrive il proprio intervento.

Se la fedeltà alla struttura del testo è criterio per noi generale e cogente, quanto caratterizza la descrizione che diamo della codifica della testata – sono qui contenute le informazioni bibliografiche ed editoriali relative al documento elettronico e alla sua fonte – è una

maggiore cura e attenzione nel trattamento dei metadati, e il tentativo costante di coniugare esigenze diverse ma complementari, quali chiarezza e rigore scientifico. Un esempio: nell'elemento <title>, contenuto nella sezione <fileDesc>, riportiamo il titolo completo dell'opera secondo l'edizione di riferimento adottata, mentre un elemento <xptr/> rimanda ad un *authority file* esterno, dove il titolo viene trascritto e proposto nelle altre diverse forme legate alla gestione informatica del file.

Abbiamo voluto analogamente più agile e accurata la compilazione relativa alla voce dell'autore: l'elemento <name>, inserito in <author>, riporta il nome dell'autore del contenuto dell'opera riprodotta così come appare nel frontespizio, mentre il valore dell'attributo *key* (che si accompagna all'elemento <name>) rimanda ad una forma normalizzata del nome (e un file centralizzato si occupa ovviamente di raccogliere tutte queste informazioni). Questo garantisce, oltre che pulizia dei dati contenuti nel file .xml, uniformità nell'indicizzazione, e concorre alla costituzione di una banca dati coerente e rigorosa, anche se ampia e complessa.

L'esigenza di una analoga accuratezza e precisione nelle indicazioni bibliografiche ci ha guidato nella modellizzazione di <sourceDesc>. Mentre <title> contiene il titolo della fonte (e coincide con il titolo riportato già in <title> figlio di <titleStmt>), il suo attributo *level* prevede diversi valori (riferibili a titolo analitico o monografico, titolo di giornale, di seriale o di materiale non pubblicato), anche se di default gli viene attribuito il valore «m» proprio del titolo monografico. Non solo: nel caso in cui l'opera non sia pubblicata autonomamente, è necessario duplicare l'elemento <title>, riportando l'attributo *level* con il valore corrispondente al titolo che contiene come una sua parte l'opera digitalizzata.

Ancora all'interno di <sourceDesc>, la presenza del campo <edition> permette di indicare eventuali peculiarità dell'edizione presa in considerazione (prima edizione, revisione o altro), mentre <respStmt> consente di individuare responsabilità d'autore secondarie relative alla fonte (curatore, traduttore); <notesStmt>, infine, riporta eventuali annotazioni riguardanti la fonte, se ritenute significative e utili per una migliore intelligibilità del testo (per esempio i dati concernenti un'edizione anastatica, o le informazioni relative al testo rispetto al quale la nostra fonte ha un rapporto di dipendenza o di derivazione).

Nel <profileDesc>, sezione dedicata alla descrizione degli aspetti non bibliografici del testo, l'elemento <langUsage> ci permette di indicare

tutte le lingue presenti nel documento. Ogni lingua viene identificata dall'elemento <language> accompagnato dall'attributo *id* (che ha come valore la sigla della lingua a cui si riferisce e come contenuto il nome della lingua. Le sigle adottate sono quelle previste nello standard ISO 639, 3-letter Language Codes, o da noi definite se lì assenti).

Passiamo alla descrizione di aspetti più specificamente strutturali del documento. Come già accennato, sebbene la TEI preveda un' articolazione dell'elemento <text> – che racchiude il contenuto testuale vero e proprio dell'opera – in tre sottogruppi (<front>, <body>, <back>), per la natura particolare dei testi che trattiamo abbiamo scelto di utilizzare il solo elemento <body>, all'interno del quale confluisce l'intero contenuto testuale, compresi quindi i materiali peritestuali preliminari e conclusivi. All'interno di <body>, le effettive divisioni strutturali dell'opera vengono rappresentate in un massimo di sette livelli gerarchici mediante l'elemento <div*n*>, ognuno specificato attraverso il valore dell'attributo *type* e attraverso un secondo attributo, *id* (immesso con l'ausilio dei SignumEText Tools dopo che sono state marcate le divisioni strutturali dell'intero testo). Per quanto riguarda i paragrafi, poiché lavoriamo su testi nei quali la divisione secondo tali elementi può essere o assente o difficilmente identificabile, a livello base non prendiamo in considerazione la possibilità di una loro codifica: allo scopo di racchiudere del testo puro usiamo ora l'elemento <ab>, proprio dei generici blocchi di testo, e lasciamo ai singoli progetti l'eventuale adozione dell'elemento <p>.

Fedeltà all'edizione di riferimento significa anche recupero di un eventuale apparato di note al testo. Proponiamo in proposito una distinzione delle note secondo due diversi modelli o tipologie: il valore «marginal» dell'attributo *type* aggiunto a <note> dichiara cioè che siamo in presenza di annotazioni semplicemente riassuntive; nel caso di una cosiddetta glossa (la spiegazione di un termine, un commento più esteso, un riferimento bibliografico aggiunto) il valore di *type* è invece «gloss». Al fine di non perdere informazione, anche nella trascrizione del testo di una nota vengono codificati tutti i fenomeni da noi considerati, con l'esclusione delle spezzature delle parole a fine riga e della numerazione delle righe. Altri attributi presi in considerazione nel trattamento delle note sono *place* e *n*. Se queste sono le indicazioni fornite per il trattamento di base del testo, a livello di progetto l'esame delle note va poi incontro ad approfondimenti ulteriori: un esempio significativo in questa direzione è costituito dalle Bibbie.

Nel trattamento delle immagini – le quali, come è noto, non possono essere inserite direttamente in un file .xml e richiedono di essere gestite a parte (l'attributo *entity* di <figure> mette in relazione il file .xml e un file in cui sono definite tutte le entità corrispondenti alle immagini presenti sulla fonte; il file delle entità costituisce cioè il ponte che permette di collegare il file .xml e i file delle immagini) – motivi legati alla gestione informatica e alla fruizione del contenuto dei documenti determinano alcune scelte nella codifica.

Segnaliamo in proposito l'adozione dell'attributo *value* (attributo di <interp/>; i valori da noi previsti sono «illustr», «text», o «symbol», a seconda che si abbia a che fare con un apparato iconografico di corredo al testo, con schemi e tabelle, o con immagini di dimensioni pari o simili ad un carattere tipografico). Un elemento <p> figlio di <figure> racchiude inoltre l'eventuale didascalia presente sulla fonte (nella sua trascrizione sono tenuti in considerazione tutti i fenomeni da noi codificati, con l'esclusione della spezzatura delle parole a fine riga e della numerazione delle linee), mentre un altro elemento figlio di <figure>, <text>, racchiude il testo eventualmente presente all'interno dell'immagine, marcato secondo le regole sopra enunciate. Una simile strutturazione dell'elemento <figure> permette in sostanza di interrogare testo e didascalie delle figure, con evidenti vantaggi nella ricerca dei dati e nel recupero delle informazioni.

Al di là del trattamento del testo su un piano strutturale, abbiamo scelto di trattare anche alcuni particolari fenomeni semantici che, pur non addentrandosi nei contenuti specifici del testo, ne consentono tuttavia una migliore fruizione. Ci riferiamo alla individuazione delle lingue presenti nel testo, al testo in versi e alla maiuscola significativa.

Attraverso l'attributo *lang* (avente come valore la sigla della lingua definita in <language>) abbiamo deciso di censire le lingue presenti nel testo, tanto quella prevalente (indicata con l'aggiunta di *lang* in <text>) che semplici porzioni di testo scritte in una lingua diversa da quella. Al livello standard della codifica il censimento prende in considerazione le lingue cosiddette moderne, senza dar conto di ulteriori delimitazioni cronologiche e geografiche, e fra quelle antiche il latino, il greco, l'ebraico. In omaggio alla natura di buona parte dei testi da noi trattati, riconosciamo dignità di lingua specifica anche al volgare italiano. Mentre nel documento base la distinzione delle lingue opera una scrematura di massima, adatta e valida per l'intera biblioteca digitale, nel piano di lavoro di alcuni singoli progetti sono previsti censimenti più dettagliati e funzionali a ricerche mirate.

Un criterio analogo ci ha guidato nel decidere se e come segnalare la forma, versi o prosa, in cui si presenta un testo. Nella codifica di base tale informazione è unicamente volta a distinguere in modo generico un testo in versi da un testo in prosa: pur senza un'attenzione specifica allo studio del fenomeno (anch'esso viene demandato, se ritenuto significativo, ai singoli progetti), tale trattamento uniforme garantisce già in partenza una ricerca più raffinata e consegna risultati corretti e rigorosi. Nella pratica, ci limitiamo a distinguere due casi, quello in cui è in versi il testo di tutta l'opera o di una sua intera divisione logica, e quello in cui porzioni di testo in versi sono per così dire immerse in un testo prevalentemente in prosa. Nel caso di opere interamente in versi, curiamo inoltre di conservare la numerazione come indicata sulla fonte, al fine di potere meglio orientare lettura e fruizione del testo stesso.

Per finire, proseguendo su una linea operativa che caratterizzava già la codifica con software DBT, differenziamo quei termini del testo che sono omografi ma hanno funzione grammaticale diversa, al fine di perfezionare i risultati della ricerca. La codifica della cosiddetta maiuscola significativa (applicata a livello base alle sole categorie dei toponimi e degli antroponimi) avviene tramite l'elemento <c> e l'attributo *type* con valore «capital».

La dimensione manualistica della *Guida SignumETexts di base* diventa ancora più evidente in chiusura. Le pagine finali – e questo costituisce una indubbia novità nel panorama di analoghi contributi – sono dedicate all'ambiente e alle procedure di lavoro: in esse vengono cioè descritti i 'luoghi' in cui i file devono essere collocati e salvati e sono forniti agli addetti ai lavori quei consigli pratici che tornano utili per l'organizzazione delle cartelle. Questi suggerimenti hanno lo scopo di garantire una migliore gestione del materiale, valida (necessaria!) tanto per il singolo curatore che a livello collettivo di redazione.

Nella sezione Procedure, dove il lavoro viene suddiviso in due fasi (preparazione del testo da copiare nel modello .xml e codifica vera e propria: la seconda di queste fasi è quella che viene poi ulteriormente strutturata in un livello base e in un livello specifico), descriviamo un modello possibile di flusso di lavoro e di sequenze operative programmate, in sostanza un *iter* verificato e ottimizzato. L'idea di questa appendice, che in pratica ricostruisce passo passo, meticolosamente, il percorso che va dalla pagina su carta alla pagina in rete, è maturata

in seguito alla ormai lunga esperienza sul campo del nostro gruppo di lavoro, nella certezza che la condivisione dei documenti informativi, degli strumenti tecnici e dei percorsi operativi sia fondamentale per costruire, gestire e sviluppare una biblioteca digitale che voglia essere di qualità. Non solo: utili in particolare a chi si cimenta per la prima volta con il trattamento e la codifica di un testo secondo lo schema da noi elaborato, queste istruzioni accessorie consentono anche di determinare con buona approssimazione le diverse aree di competenza nelle operazioni condotte sul testo e sui suoi metadati (quali del codificatore, quali della redazione), e di tracciare finalmente un quadro complessivo ed eloquente dei compiti, delle responsabilità, delle competenze richiesti ai singoli collaboratori. Le 'normali' attività di acquisizione e di codifica del testo si affiancano infatti – se e dove necessario – ad interventi di correzione, di scioglimento, di integrazione; di titolazione di testo e di immagini, se mancano i riferimenti necessari; in generale, nel trattamento dei metadati, di cura della coerenza interna del testo; e come vedremo più avanti, dove parliamo del secondo livello di codifica, in testi privi di edizione moderna o critica è al codificatore che sono affidati due compiti particolarmente laboriosi, delicati e preziosissimi dal punto di vista scientifico, quali lo studio e il reperimento delle fonti e delle citazioni.

Questa consapevolezza spiega perché, all'interno della *Guida*, sono per noi costante oggetto di definizione e di precisazione i compiti, le competenze e le responsabilità dei singoli collaboratori. Nel <teiHeader> per esempio abbiamo individuato e indicato espressamente tre diversi tipi di responsabilità, due dei quali sempre presenti. All'interno del campo <respStmt> (il tipo di responsabilità viene segnalato con <resp> seguito da <name>, contenente i dati del singolo codificatore), l'operazione di digitalizzazione del testo, della correzione e della uniformazione ai principi editoriali adottati viene definita come «acquisizione»; «codifica» definisce invece l'operazione di marcatura (e qualora il responsabile della marcatura di primo livello non coincida con quello del secondo, distinguiamo fra responsabilità di codifica di base e di codifica di secondo livello; «elaborazione immagini» si riferisce alle operazioni di acquisizione, ritocco e ridimensionamento delle immagini (là dove il testo presenti un apparato iconografico).

Abbiamo detto che, al di là del trattamento di base a cui i testi di tutti i progetti che afferiscono a Signum devono essere sottoposti, prevediamo un secondo livello di codifica legato ai singoli progetti: qui, elementi specifici sono pensati in funzione di specifici contenuti,

propri o esclusivi di determinate aree di studio e di ricerca. Poiché nella pratica succede che alcuni fenomeni previsti nella codifica di secondo livello riguardano la quasi totalità dei progetti, riteniamo opportuno darne qualche indicazione, limitandoci agli aspetti a nostro parere scientificamente più rilevanti. Fenomeni su cui abbiamo concentrato la nostra attenzione riguardano i titoli delle opere, le date e le citazioni.

Ancora in modo più evidente che nel livello standard, la marcatura è stata qui pensata per un'utenza di tipo specialistico, interessata ad un pieno recupero delle informazioni a livello linguistico e filologico, spesso orientata ad un censimento delle fonti, sempre comunque attenta ad un'indicizzazione testuale che sia il più possibile esaustiva e rigorosa.

Nella ricerca costante di coniugare chiarezza e rigore scientifico, anche a questo livello abbiamo mantenuto ferma la scelta di gestire separatamente i metadati dai file .xml: così, ad esempio, tutte le informazioni relative agli autori (i nomi sono normalizzati secondo le norme RICA) e alle opere (<title> in <list> riporta il titolo normalizzato secondo una lista messa a disposizione dalla redazione), sono contenute in un apposito file esterno .xml dei riferimenti bibliografici (costituito da una testata codificata dall'elemento <teiHeader> e da un gruppo base di campi racchiusi nell'elemento <list> in cui si riportano le informazioni).

Nell'<editorialDecl> del <teiHeader> abbiamo indicato le modalità uniformemente adottate nella nostra biblioteca digitale per il trattamento delle date: sono rappresentate nel sistema ISO (e cioè un numero costituito da otto cifre indicanti in successione anno, mese e giorno, precedute da un «–» quando la data sia un termine *ante Christum natum*). A livello di progetto, approfondimento significa soprattutto possibilità di recuperare informazioni anche sugli intervalli di tempo segnalati nelle espressioni di data e sulle indicazioni di periodi corrispondenti a secoli: stringhe di testo come «fra l'11 maggio e il 13 giugno del 1963» e «nel XIII secolo» sono così codificate ricorrendo all'elemento <dateRange> accompagnato dagli attributi *from* e *to*. Situazioni più complesse, che in una sola indicazione di data includono per esempio notizie sovrapposte (tipo: «il 4 o il 5 maggio del 1450 o del 1451»), sono risolte facendo ricorso all'elemento <dateStruct> (che racchiude l'intera indicazione temporale: al suo interno vengono poi adeguatamente marcati i singoli dati).

Trovare una soluzione calzante ed appropriata per il trattamento di quanto definiamo citazioni e rimandi – riprese letterali di fonti

esterne al testo e passi non strettamente letterali ma che parafrasano un luogo identificato o identificabile – ha richiesto un notevole impegno. Limitato a *BiViO*, alle *Bibbie* e alle *Fonti storico-artistiche* – e in questi progetti diversamente declinato secondo le specifiche esigenze – anche il censimento delle citazioni e dei rimandi comporta – come già nelle immagini – un lavoro su tre file (due .xml, relativi al testo e ai riferimenti bibliografici, e uno che collega i primi due tramite la definizione di entità utilizzate nel file .xml del testo). Senza entrare nel trattamento specifico del fenomeno, nel file .xml del testo la citazione viene racchiusa all'interno di un elemento <quote> figlio di <xref>, che è accompagnato dagli attributi «type», «doc» e «from», il rimando è segnalato invece con <xptr>. La funzione degli attributi «doc» e «from» è, in entrambi i casi, quella di mettere in relazione il file .xml del testo con il file dei riferimenti bibliografici, e di collegare il fenomeno censito sul testo con la notizia bibliografica relativa.

Istruiti da una casistica significativa, nel trattare i casi di citazione annidata (una citazione all'interno di un'altra citazione) e di citazione errata (il passo oggetto di citazione viene attribuito erroneamente dall'autore del testo), abbiamo scelto soluzioni che consentono di disegnare un quadro il più possibile vivo e preciso della biblioteca ideale propria dei singoli autori (in questi casi è pertanto data sempre la precedenza alla fonte citata sul testo, mentre la voce bibliografica viene strutturata riportando in <bibl> figlio di <item> le notizie della fonte indicata dall'autore e in <bibl> figlio di <note> le notizie relative alla fonte esatta). Nel riportare le indicazioni bibliografiche facciamo analogamente particolare attenzione a quelle edizioni antiche che si ha motivo di credere essere state usate o possedute dai diversi autori: il campo <edition>, in <bibl>, ci consente di recuperare eventuali indicazioni in proposito, e collabora alla ricostruzione della biblioteca materiale dell'autore.

Nella sezione dedicata al file .xml dei riferimenti bibliografici, campi specifici consentono di dichiarare l'identità del codificatore e del ruolo da lui svolto: nel caso – tutt'altro che raro nella pratica dei testi che digitalizziamo – in cui non abbia utilizzato informazioni bibliografiche individuate da altri, una formula predefinita (che sostituisce la dichiarazione di istruzione racchiusa in un elemento <p> all'interno di <sourceDesc>) segnala che il contenuto del file è stato curato dal codificatore stesso. Il controllo delle eventuali informazioni bibliografiche fornite dal testo fonte rientra in ogni caso fra le responsabilità del codificatore.

Ricordiamo infine che, per garantire quell'uniformità imprescindibile nella costituzione di una biblioteca digitale, tutti i lavori vengono sottoposti ad *editing* complessivo dopo la prima consegna, e ad un ulteriore *editing* finale a livello di progetto.

Guida SignumETexts di base: <http://e-texts.signum.sns.it/> (in costruzione).

Marzia Bonfanti - Daniele Marotta
Francesca Dell'Omodarme - Serena Ferretti - Matteo Gallo

Bibliografia

Adamo 1987 = G. Adamo, *La codifica come rappresentazione. Trasmissione e trattamento dell'informazione nell'elaborazione automatica di dati in ambito umanistico*, in Gigliozzi 1987, pp. 39-63.

Almini 2004 = S. Almini, *Le propettive di sviluppo del PLAIN (Progetto Lombardo Archivi in Internet) e del portale Lombardia Storica nel corso del 2004*, in «Archivi&Computer», XIV, 2004, pp. 106-117.

Anagrafe 2000 = *Riprogettare "Anagrafe". Elementi per un nuovo sistema archivistico nazionale. Relazione del gruppo di lavoro per la revisione e la reingegnerizzazione del sistema informativo nazionale "Anagrafe informatizzata degli archivi italiani"*, in «Rassegna degli Archivi di Stato», LX, 2000, pp. 373-454.

Auricchio *et al.* 2002 = S. Auricchio, I. Barbanti, S. Mazzini, C. Veninata, *XML per l'informatizzazione degli archivi storici comunali*, in «Bollettino d'informazioni. Centro di ricerche informatiche per i beni culturali della Scuola Normale Superiore di Pisa», XII, 2002, pp. 95-112.

Authority Control 2003 = *Authority Control. Definizione ed esperienze internazionali*. Atti del convegno internazionale (Firenze, 10-12 febbraio 2003), a cura di M. Guerrini, B. Tillett, L. Sardo, Firenze, Firenze University Press-Associazione italiana biblioteche 2003.

Barbanti, Bruno 2003 = I. Barbanti, G. Bruno, *DAMS Solutions. Presentazione*, in «Archivi&Computer», XIII, 2003, pp. 29-35.

Barchesi 2001 = C. Barchesi, *Progetto Caere: un'applicazione Internet attiva per l'information retrieval di documenti SGML*, in «Archeologia e Calcolatori», XII, 2001, pp. 71-90.

Barchesi *et al.* 2003 = C. Barchesi, P. Moscati, P. Santoro, D. Scarpati, *Ricerche archeologiche sul campo e archivi digitali: il manoscritto di Ercole Nardi*, in «Archeologia e Calcolatori», XIV, 2003, pp. 295-325.

Berners-Lee 1998 = T. Berners-Lee, *Semantic Web Road Map*, <http://www.w3.org/DesignIssues/Semantic.html>.

Berners-Lee 1999 = T. Berners-Lee, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*, San Francisco, HarperCollins 1999.

Bondielli 2001 = *SIUSA – Sistema informativo unificato per le soprintendenze archivistiche. Genesi e sviluppo di un progetto*, a cura di D. Bondielli, in «Bollettino d'informazioni. Centro di ricerche informatiche per i beni culturali della Scuola Normale Superiore di Pisa», XI, 2001.

Bondielli 2002 = D. Bondielli, *I sistemi informativi archivistici in rapporto alle risorse telematiche: nuovi progetti a confronto*, in «Archivi & Computer», XI, 2002, pp. 48-57.

Bondielli, Vitali 2000 = D. Bondielli, S. Vitali, *Descrizioni archivistiche sul Web: la guida on-line dell'Archivio di Stato di Firenze*, in «Bollettino d'informazioni. Centro di ricerche informatiche per i beni culturali della Scuola Normale Superiore», X, 2000, pp. 7-27.

Bonifati, Ceri 2000 = A. Bonifati, S. Ceri, *Comparative Analysis of Five XML Query Languages*, in «ACM SIGMOD Record», XXIX, 1, 2000, pp. 68-79.

Bonincontro 2001 = I. Bonincontro, *Progetto Caere: prospettive di applicazione degli standard internazionali per la codifica dei dati testuali*, in «Archeologia e Calcolatori», XII, 2001, pp. 55-70.

Braga *et al.* 2003 = D. Braga, A. Campi, S. Ceri, E. Augurusa, *Xquery By Example*, in *Proceedings of WWW2003,* Poster Section, <http://www2003.org/cdrom/papers/poster/p291/p291-braga.html>.

Bray 1999 = T. Bray, *XML namespaces by example,* <http://www.xml.com/pub/a/1999/01/namespaces.html>.

Burnard, Sperberg-McQueen 1995 = L. Burnard, C.M. Sperberg-McQueen, *TEI Lite: An Introduction to Text Encoding for Interchange*, revised May 2002, <http://www.tei-c.org/lite/>.

Castagnoli 1978 = F. Castagnoli, *La Carta archeologica d'Italia*, in «La Parola del Passato», XXXIII, 1978, pp. 78-80.

Ceccarelli 2001 = L. Ceccarelli, *Progetto Caere: dallo scavo al territorio. Una soluzione per la distribuzione dei dati mediante un GIS on-line*, in «Archeologia e Calcolatori», XII, 2001, pp. 105-121.

Ceri *et al.* 2000 = S. Ceri, P. Fraternali, S. Paraboschi, *XML: Current Developments and Future Challenges for the Database Community*, in *Proceedings of the 7th International Conference on Extending database Technology (EDBT 2000)*, Konstanz (Germany) 2000, pp. 3-17.

Chase 2002 = N. Chase, *Customize a DTD with parameter entities*, <http://www-106.ibm.com/developerworks/xml/library/x-tiparam.html?dwzone=xml>.

Connolly 2002 = D. W. Connolly, *An Evaluation of the World Wide Web with Respect to Engelbart's Requirements*, <http://www.w3.org/Architecture/NOTE-ioh-arch/>.

Cozza, Pasqui 1981 = A. Cozza, A. Pasqui, *Carta archeologica d'Italia (1881-1897) – Materiali per l'Agro Falisco*, *Forma Italiae*, Serie II, Documenti 2, Firenze, Leo S. Olschki 1981.

Deutsch *et al.* 1999 = A. Deutsch, M.F. Fernandez, D. Florescu, A.Y. Levy, D. Maier, D. Suciu, *Querying XML Data,* in «IEEE Data Engineering Bulletin», XXII, 3, 1999, pp. 10-18.

DC-TI2 = G. Geser, N. Kanter, J. vaan Kasteren, M. Moon, S. Ross, M. Steemson, *Digital Asset Management Systems for the Cultural and Scientific Heritage Sector,* <http://www.digicult.info/downloads/thematic_issue_2_021204_low_resolution.pdf/>. Thematic Issues. 2. DigiCult Consortium.

EAD 1997, Part 1 = *Encoded Archival Description. Part 1 – Context and Theory*, in «The American Archivist», LX, estate 1997.

EAD 1997, Part 2 = *Encoded Archival Description. Part 2 – Case Studies*, in «The American Archivist», LX, autunno 1997.

EAD 1998 = *EAD. Encoded Archival Description Tag Library. Version 1.0*. Prepared and Maintained by the Encoded Archival Description Working Group of the Society of American Archivists and the Network Development and MARC Standards Office of the Library of Congress, Chicago, The Society of American Archivists 1998.

EAD 1999 = *EAD. Encoded Archival Description Application Guidelines. Version 1.0*. Prepared and Maintained by the Encoded Archival Description Working Group of the Society of American Archivists, Chicago, The Society of American Archivists 1999.

EAD 2002 = *EAD – Encoded Archival Description, Version 2002*, <http://www.loc.gov/ead/ead2002.html>.

Engelbart 1990 = D.C. Engelbart, *Knowledge-Domain Interoperability and an Open Hyperdocument System*, in *Proceedings of the Conference on Computer-Supported Cooperative Work*. October 7-10, 1990, Los Angeles 1990, pp. 143-156, <http://www.bootstrap.org/augdocs/augment-132082.htm>

Fielding 2000 = R. T. Fielding. *Representational State Transfer. Doctoral Thesis* 2000. *Chapter 5, Architectural Styles and the Design of Network-based Software Architectures*, <http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm>.

Gamurrini *et al.* 1972 = G.F. Gamurrini, A. Cozza A. Pasqui, R. Mengarelli, *Carta Archeologica d'Italia (1881-1897). Materiali per l'Etruria e la Sabina*, *Forma Italiae*, Serie II, Documenti 1, Firenze, Leo S. Olschki 1972.

Gigliozzi 1987 = *Studi di codifica e trattamento automatico di testi*, a cura di G. Gigliozzi, Roma, Bulzoni Editore 1987.

Grossi 2003 = M. Grossi, *Recensione* a *DAMS Solutions*, in «Archivi & Computer», XIII, 3, 2003, pp. 35-42.

Grosso, Walsh 2000 = P. Grosso, N. Walsh, *XSL Concepts and Practical Use*, <http://www.nwalsh.com/docs/tutorials/xsl/xsl/frames.html>.

Guha *et al.* 2003 = R. Guha, R. McCool, E. Miller, *Semantic Search*, in *Proceedings of WWW2003*, pp 700-709.

Guida generale degli Archivi di Stato 1981-1994 = *Guida generale degli Archivi di Stato*, I-IV, Roma, Ministero per i beni culturali e ambientali 1981-1994.

Guidi, Santoro 2003 = A. Guidi, P. Santoro, *Il Progetto Galantina*, in AA.VV., *Lazio e Sabina*. Atti del Convegno (Roma 2002), Roma, De Luca Editore 2003, pp. 109-114.

Halevy *et al.* 2003 = A.Y. Halevy, Z.G. Ives, P. Mork, I. Tatarinov, *Piazza*: *Data Management Infrastructure for Semantic Web Applications*, in *Proceedings of WWW2003*, pp. 556-567.

Hayashi *et al.* 2003 = Y. Hayashi, J. Tomita G. Kikui, *Searching Text-rich XML Documents with Relevance Ranking*, in *ACM SIGIR2000 Workshop on XML and Information Retrieval*, July 28, 2000, Athens 2003.

Hensen 1989 = S.L. Hensen, *Archives, personal papers and manuscripts*, 2nd ed., Chicago, Society of American Archivists 1989.

Hensen 1996 = S.L. Hensen, *Archivi, manoscritti e documenti*, San Miniato, Archilab 1996.

Kim, Holman 2002 = E. Eric Kim, K. Holman. *Interoperability between Collaborative Knowledge Applications*, in «Extreme Markup Languages 2002», August 6-9, 2002, Montréal 2002, <http://www.eekim.com/papers/2002/eml/EML2002Holman01-toc.html>.

Kotsakis 2003 = E. Kotsakis, *Structured information retrieval in XML documents*, in *Proceedings of the 2002 ACM symposium on Applied Computing*, Madrid 2003, pp. 663-667.

Mancinelli 2004 = M.L. Mancinelli, *Sistema Informativo Generale del Catalogo: nuovi strumenti per la gestione integrata delle conoscenze sui beni archeologici*, in *Nuove frontiere della ricerca archeologica. Linguaggi, comunicazione, informatica*, a cura di P. Moscati, in «Archeologia e Calcolatori», XV, 2004, pp. 115-128.

Mariotti 2001 = S. Mariotti, *Progetto Caere: proposta di un modello per il trattamento e la codifica di documenti archeologici editi*, in «Archeologia e Calcolatori», XII, 2001, pp. 91-104.

Mazzini, Veninata 2004 = S. Mazzini, C. Veninata, *Il progetto "RInASCo. Recupero Inventari degli Archivi Storici Comunali"*, in «Archivi & Computer», XIV, 2004, pp. 118-135.

Miller 1998 = E. Miller, *An Introduction to the Resource Description Framework*, in «D-Lib Magazine», May 1998, <http://www.dlib.org/dlib/may98/miller/05miller.html>.

Miller, Sheth 2000 = J.A. Miller, S. Sheth, *Querying XML documents. Potentials*, in «IEEE Data Engineering Bulletin», XIX, 1, 2000, pp. 24-26.

Morgan 2002 = E.L. Morgan, *Fun with XML*, <http://www.infomotions.com/xml/fun.html>.

Moscati 1990 = P. Moscati, *Nuove ricerche su Falerii Veteres*, in AA.VV., Atti del XV Convegno di Studi Etruschi e Italici, *La Civiltà dei Falisci* (Civita Castellana 1987), Firenze, Leo S. Olschki 1990, pp. 141-171.

Moscati 1991 = P. Moscati, *Ricerche nell'area urbana di Caere: la gestione automatizzata dei dati di scavo*, in «Archeologia Classica», XLIII, 1991, pp. 303-316.

Moscati 2001 = P. Moscati, *Progetto Caere: questioni di metodo e sperimentazioni*, in «Archeologia e Calcolatori», XII, 2001, pp. 47-54.

Moscati 2003 = P. Moscati, *Dal dato al modello: l'approccio informatico alla ricerca archeologica sul campo*, in *I modelli nella ricerca archeologica: il ruolo dell'informatica*. Atti del Convegno (Roma 2000), Roma, Accademia Nazionale dei Lincei 2003, pp. 55-76.

Moscati c.s. = P. Moscati, *Linguaggi di marcatura per la conservazione e la valorizzazione dell'informazione archeologica*, in Atti del Convegno *Trusted Digital Repository for Cultural Heritage* (Roma 2003), in corso di stampa.

Moscati, Mariotti, Limata 1999 = P. Moscati, S. Mariotti, B. Limata, *Il "Progetto Caere": un esempio di informatizzazione dei diari di scavo*, in «Archeologia e Calcolatori», X, 1999, pp. 165-188.

Navarro, Baeza-Yates 1997 = G. Navarro, R. Baeza-Yates, *Proximal nodes: a model to query document databases by content and structure*, in «ACM TOIS», XV, 4, Oct. 1997, pp. 400-435.

Nelson 1993 = Th.H. Nelson, *Literary Machines*, Sausalito CA, Mindful Press 1993.

Orlandi 1999 = T. Orlandi, *Ripartiamo dai diasistemi*, in *I nuovi orizzonti della filologia. Ecdotica, critica testuale, editoria scientifica e mezzi informatici*. Atti del Convegno internazionale (Roma 1998), Roma, Accademia Nazionale dei Lincei 1999, pp. 87-101.

Pastura *et al.* 2004 = M.G. Pastura, D. Iozzia, D. Spano, M. Taglioli, *Il Sistema Informativo Unificato per le Soprintendenze Archivistiche*, in «Archivi & Computer», XIV, 2004, pp. 64-77.

Phillips 2000 = L.A. Phillips, *Usare XML*, Milano, Mondadori Informatica 2000.

Pitti 1994 = D. Pitti, *Una ricchezza in comune. Verso uno standard che codifichi strumenti per la consultazione*, in «Archivi & Computer», VI, 1994, pp. 154-162.

Pitti 1997 = D. Pitti, *Encoded Archival Description: The Development of an Encoding Standard for Archival Finding Aids*, in «The American Archivist», LX, 1997, pp. 268-283.

Pitti 2003 = D. Pitti, *Descrizione del soggetto produttore. Encoded Archival Context*, in *Authority Control* 2003, pp. 133-177.

Powell 2003 = A. Powell, *Expressing Dublin Core in HTML/XHTML meta and link elements*, <http://dublincore.org/documents/dcq-html/>.

Proceedings of WWW2003 = *Proceedings of WWW2003, The Twelfth International World Wide Web Conference*, 20-24 May 2003, Budapest (Hungary).

Rendina 2003 = E. Rendina, *Strumenti di ricerca e trattamento informatico: la* Guida generale degli Archivi di Stato *in formato XML*, in «Archivi & Computer», XIII, 2003, pp. 85-96.

Ross 2003 = S. Ross, *Position paper*, in *Towards a Semantic Web for Heritage Resources*, in «Digicult», Thematic Issue 3, <http://www.digicult.info/downloads/ti3_high.pdf>.

Ruth 1997 = J. E. Ruth, *Encoded Archival Description. A Structural Overview*, in «The American Archivist», LX, 1997, pp. 310-329.

Santoro *et al.* c.s. = P. Santoro, M.L. Agneni, C. Barchesi, F. Candelato, R. Gabrielli, D. Peloso, A. Guidi, H. Patterson, *Il progetto Galantina: primi risultati*, in Atti del VI Convegno di Archeologia italiana (Groningen 2003), in corso di stampa.

Savoja 2001 = M. Savoja, *L'archivista in rete: primi cenni ad un progetto in corso*, in «Archivi per la Storia», XIV, 2001, pp. 341-354.

Savoja, Weston 2003 = M. Savoja, P.G. Weston, *Progetto Lombardo Archivi in INternet (PLAIN): identificazione, reperimento e presentazione dei soggetti produttori e dei complessi archivistici*, in *Authority Control* 2003, pp. 387-399.

Scarpati c.s. = *Ruderi delle Ville Romano Sabine nei dintorni di Poggio Mirteto, illustrati dal Prof. Ercole Nardi, 1885. Itinerario Primo*, a cura di D. Scarpati, Poggio Mirteto, in corso di stampa.

Signore 2001 = O. Signore, *Il ruolo centrale di XML nell'evoluzione del Web - Proceedings XML Day, LogOn Technology Transfer GmbH,* Milan September 21, 2001, <http://www.w3c.it/papers/signorePerXMLDays2001.pdf>.

Signore 2002a = O. Signore, *Una panoramica delle tecnologie W3C-XML Day Milano, Conference proceedings,* Milan, September 20, 2002 <http://www.w3c.it/papers/signoreForXMLDays2002.pdf>.

Signore 2002b = O. Signore, *RDF per la rappresentazione della conoscenza, - Knowledge Management 2002 (V edizione) - Forum Proceedings on the Knowledge Management in the Organizations,* Jekpot S.r.l. Milano, 27-28 marzo 2002, pp. 35-46 <http://www.w3c.it/papers/RDF.PDF>.

Signore 2002c = O. Signore, *Qualità dei siti web pubblici culturali: dal progetto MINERVA all'interoperabilità semantica*, in «Bollettino d'Informazioni. Centro di ricerche informatiche per i beni culturali della Scuola Normale Superiore», XII, 2, 2002, pp. 9-23.

Simpson 2001 = J.E. Simpson, *Namespace nuances*, <http://www.xml.com/pub/a/2001/07/05/namespaces.html>.

Taglioli, Valgimogli 2002 = *La guida on-line dell'Archivio di Stato di Firenze*. Atti della presentazione (Firenze, 3 dicembre 2002), a cura di M. Taglioli e L. Valgimogli, in «Bollettino d'informazioni. Centro di ricerche informatiche per i beni culturali della Scuola Normale Superiore», XII, 2002, pp. 121-198.

Tola, Castellani 2004 = *Futuro delle memorie digitali e patrimonio culturale*. Atti del Convegno internazionale, Firenze, 16-17 ottobre 2003 / Istituto centrale per il catalogo unico delle biblioteche italiane e per le informazioni bibliografiche, Istituto di studi per la tutela dei beni archivistici e librari dell'Università degli studi di Urbino, a cura di V. Tola e C. Castellani, Roma, ICCU 2004.

Vitali 1994 = S. Vitali, *Il dibattito internazionale sulla normalizzazione della descrizione: aspetti teorici e prospettive in Italia*, in «Archivi & Computer», IV, 1994, pp. 303-323.

Vitali 2003a = *Un'indagine sui programmi di inventariazione archivistica*, a cura di S. Vitali, in «Archivi & Computer», XIII, 2003, pp. 7-75.

Vitali 2003b= S. Vitali, *La seconda edizione di ISAAR (CPF) e il controllo d'autorità nei sistemi di descrizione archivistica*, in *Authority Control* 2003, pp. 139-152.

Weber 2003 = J. Weber, *LEAF. Collegare ed esplorare gli authority file*, in *Authority Control* 2003, pp. 179-186.