

OPEN SOURCE, FREE SOFTWARE E  
OPEN FORMAT NEI PROCESSI DI  
RICERCA ARCHEOLOGICI

ATTI DEL I WORKSHOP  
(GROSSETO, 8 MAGGIO 2006)

a cura di

ROBERTO BAGNARA E GIANCARLO MACCHI JÁNICA

## Indice

Introduzione: Open Source, Free Software e Open Format nei processi di ricerca archeologici ROBERTO BAGNARA, GIANCARLO MACCHI JÁNICA	3
ArcheOS 1.1.6 Akhenaton: la nuova release della prima distribuzione GNU/Linux per archeologi ALESSANDRO BEZZI, LUCA BEZZI, DENIS FRANCISCI, RUPERT GIELT	9
L <sup>A</sup> T <sub>E</sub> X <sub>e</sub> natbib itarcho: un formato scientifico per il testo archeologico EMANUEL DEMETRESCU	19
A monte del software libero. Riflessioni sulla progettazione di architetture aperte per la gestione del dato archeologico VITTORIO FRONZA	35
OS vs “Tradizione”: un punto di vista archeologico CARLO CITTER	49
I linguaggi standard del W3C e gli strumenti Open Source per la gestione dei dati archeologici ANDREA D’ANDREA, ACHILLE FELICETTI, SORIN HERMON, FRANCO NICOLUCCI, TOMMASO ZOPPI	53
XML ed il Web per la gestione di dati archeologici EMILIANO SCAMPOLI	69
Iosa.it: una risorsa online per l’incontro tra archeologia e software libero STEFANO COSTA	89

GRASS, un potente GIS per archeologi MICHAEL BARTON, ALESSANDRO BEZZI, LUCA BEZZI, DENIS FRANCISCI, RUPERT GIETL, MARKUS NETELER	97
Utilizzo di tecnologie OpenSource e FreeSoftware per l'implementazione di web GIS per la consultazione di dati archeologici: il caso della Spina Verde di Como STEFANO ALIVERNINI, MARIA ANTONIA BROVELLI, DIEGO MAGNI	105
ARIS: un sistema per la gestione del dato archeologico FRANCESCA CUNIOLO	125
P.I.C.A. (Portale Informatico Culturale delle Alpi occidentali): un portale Open Source per i Beni Culturali DANILO DEMARCHI, GIORGIO DI GANGI, CHIARA MARIA LEBOLE	139
OS/FS nella pubblicazione di banche dati topografiche: il caso del Atlante dei Siti Fortificati della Toscana GIANCARLO MACCHI JANICA, MICHELA SERRAGLI, ANDREA VICHI	155
Esperienze di predisposizione e gestione della banca dati e del sito web relativi all'Insula del Centenario a Pompei ALBERTO CUSTODI, LINO SCIORTINO	183
Tecnologie Open Source e servizi web per la condivisione di conoscenze sulla preistoria e la protostoria in Italia ANDREA BONOMI, MAURIZIO CATTANI, GLAUCO MANTEGARI, GIUSEPPE VIZZARI	203
“Revolution OS” in archeologia: esempi di interfacce web per l'archeologia del paesaggio CARLO CAMPORESI, AUGUSTO PALOMBINI, SOFIA PESCARIN	225
Il caso di studio relativo alla documentazione di scavo di Villa di Villa, Cordignano LUCA BEZZI, STEFANO BOARO, GIOVANNI LEONARDI, DAMIANO LOTTO	237

# I linguaggi standard del W3C e gli strumenti Open Source per la gestione dei dati archeologici.

Andrea D'Andrea<sup>1</sup>, Achille Felicetti<sup>2</sup>, Sorin Hermon<sup>2</sup>,  
Franco Nicolucci<sup>3</sup>, Tommaso Zoppi<sup>2</sup>

SOMMARIO. La gestione dei dati archeologici richiede formati aperti e adeguati strumenti di gestione. I linguaggi *XML-based* hanno la flessibilità necessaria per garantire ai dati trasparenza, compatibilità e velocità di diffusione e l'Open Source dimostra particolare robustezza e versatilità nella produzione di software *XML-oriented*. Il sistema da noi sviluppato (MAD) utilizza il connubio tra Open Source e tecnologie XML per realizzare un applicativo realmente *open* in grado di gestire basi di dati XML in formato nativo, compatibile con standard internazionali e pronto per essere integrato e utilizzato in diversi scenari.

*The management of archaeological data needs open formats and satisfactory management tools. The XML-based languages have all the needed flexibility to give data transparency, compatibility and rapidity of diffusion and Open Source is powerful and flexible enough for the production of good XML-oriented software. The system we developed (MAD) takes advantage of the perfect integration between Open Source and XML technologies for creating a real open application able to manage sets of XML data in native format, compatible with international standards and ready to be integrated and used in a wide range of scenarios.*

## 1. Introduzione

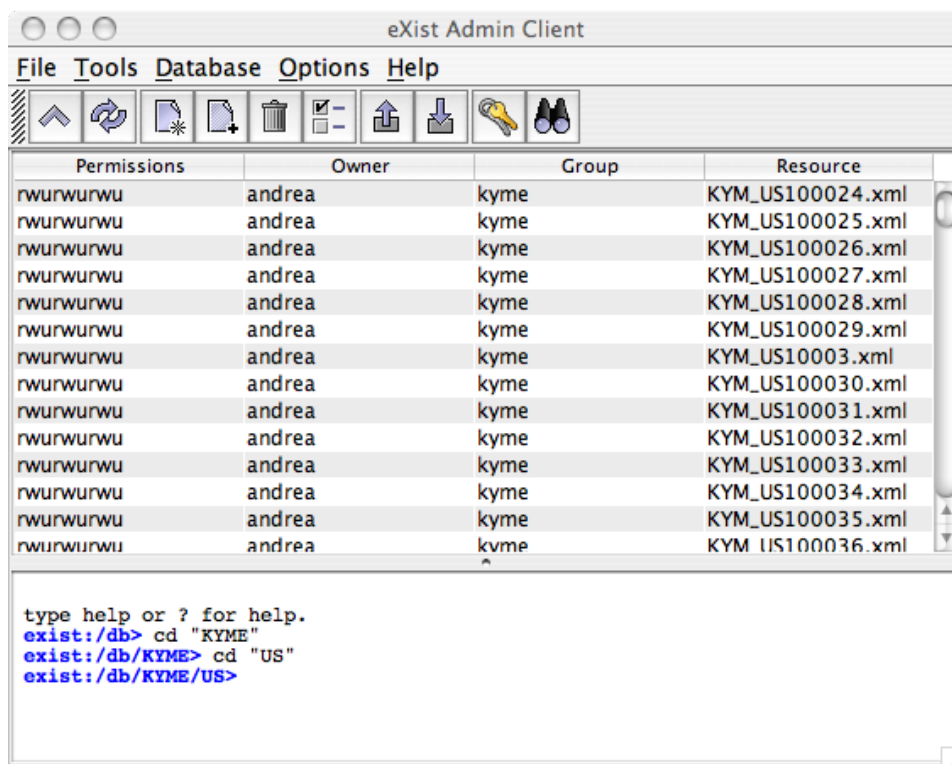
La complessità e la mole dei dati prodotti dalla ricerca archeologica moderna richiedono la messa a punto di nuovi e più adeguati strumenti, la creazione di nuovi sistemi e la definizione di standard che consentano

---

<sup>1</sup>CISA, Università degli Studi di Napoli "L'Orientale"

<sup>2</sup>PIN, Università degli Studi di Firenze, Prato

<sup>3</sup>Università degli Studi di Firenze



**Figura 1.** L'interfaccia grafica di eXist per la gestione locale del database XML

organizzazione razionale e interoperabilità fra informazioni provenienti da contesti spesso diversi. I formati binari risultano spesso non adeguati per soddisfare le esigenze di gestione dei dati di una disciplina che diventa sempre più articolata e complessa: gli archeologi necessitano di formati flessibili, facilmente editabili e che consentano la conservazione delle informazioni nel lungo periodo (D'Andrea, Niccolucci 2001).

## 2. I linguaggi W3C e i dati archeologici

I linguaggi elaborati dal *World Wide Web Consortium* (organismo internazionale il cui principale obiettivo è la definizione di tecnologie intese ad espandere le potenzialità del Web) e, in particolare, quelli basati su XML offrono un valido punto di partenza per garantire l'interoperabilità dei dati essendo per loro stessa natura aperti (non proprietari), flessibili e facilmente integrabili in diversi contesti d'uso.

Il documento XML presenta notevoli vantaggi rispetto ai classici database relazionali tradizionalmente usati in archeologia perché mette a disposizione dell'archeologo uno strumento semplice ma potente per il *recording* e l'integrazione dei dati e garantisce allo stesso tempo trasparenza, compatibilità e velocità di diffusione delle informazioni. XML è un linguaggio cross-platform, indipendente da qualsiasi software e hardware, la cui peculiare natura rende possibile lo scambio di dati anche fra sistemi incompatibili. I documenti XML, a differenza dei dati indicizzati nei comuni database, sono totalmente *portabili*, possono cioè essere utilizzati come "base di dati" da differenti tipi di applicazioni senza dover subire alcun processo di esportazione o riconversione, poiché le informazioni sono descritte utilizzando semplici file di testo e nessun particolare software è richiesto per l'editing dei documenti. La caratteristica struttura ad albero implementata da XML consente, inoltre, la definizione di relazioni più articolate fra i singoli elementi, grazie al modo esplicito di descrivere informazioni dato dalla combinazione di testo libero e marcatori: i documenti così strutturati risultano facilmente leggibili sia direttamente dall'utente - tramite un semplice software di produzione testuale quali NotePad e Microsoft Word - sia dalla macchina attraverso specifici applicativi (Niccolucci 2002).

La grande flessibilità di questo tipo di struttura assicura anche la possibilità di archiviare particolari tipi di documenti molto utilizzati in ambito archeologico: i cosiddetti documenti *non strutturati*, nei quali solitamente i dati non seguono un particolare ordine, ma sono localizzati in forma descrittiva all'interno di un testo che fluisce in modo libero (diari di scavo, relazioni di scavo); è fondamentale, per questo tipo di documenti, non alterare il flusso delle informazioni come sarebbe, invece, necessario fare se si intendesse archiviare il loro contenuto in un comune database relazionale: essi necessitano, infatti, di essere archiviati preservando la loro struttura testuale per individuare le informazioni in modo non distruttivo (Crescioli, D'Andrea, Niccolucci 2002).

Un aspetto determinante per l'utilizzo di tecnologie *XML-based* in ambito archeologico risiede nella loro capacità di descrivere la natura stessa dei dati e di individuare particolari rapporti semantici grazie ai quali è possibile eseguire ricerche razionali a partire da specifiche condizioni e relazioni interne mediante l'utilizzo di motori di ricerca "intelligenti" capaci di discernere informazioni semantiche. Si tratta del primo passo verso l'implementazione di quello che comunemente viene definito *Semantic Web*, uno scenario in cui lo scambio di informazioni poggia sulla loro "comprensione" da parte della macchina,

oramai perfettamente in grado di fornire risultati precisi grazie ad articolate definizioni poste a corredo dei dati stessi e di eseguire in modo automatico ricerche complesse oggi realizzabili, con grande dispendio di tempo, solo manualmente da parte degli utenti (Niccolucci 2002).

Specifici modelli complessi per rendere i dati “comprensibili” da parte di tool pensanti sono in via di sviluppo (Ontologie) mentre nuovi standard internazionali per potenziare l’integrabilità e l’intercomunicazione fra sorgenti di dati diverse sono in corso di definizione (CIDOC-CRM): i linguaggi usati per la loro realizzazione sono tutti basati su XML (RDF, OWL), la cui estensibilità risulta particolarmente adatta per la definizione di relazioni complesse fra i dati di un determinato dominio (D’Andrea 2006).

### 3. I formati aperti e l’Open Source

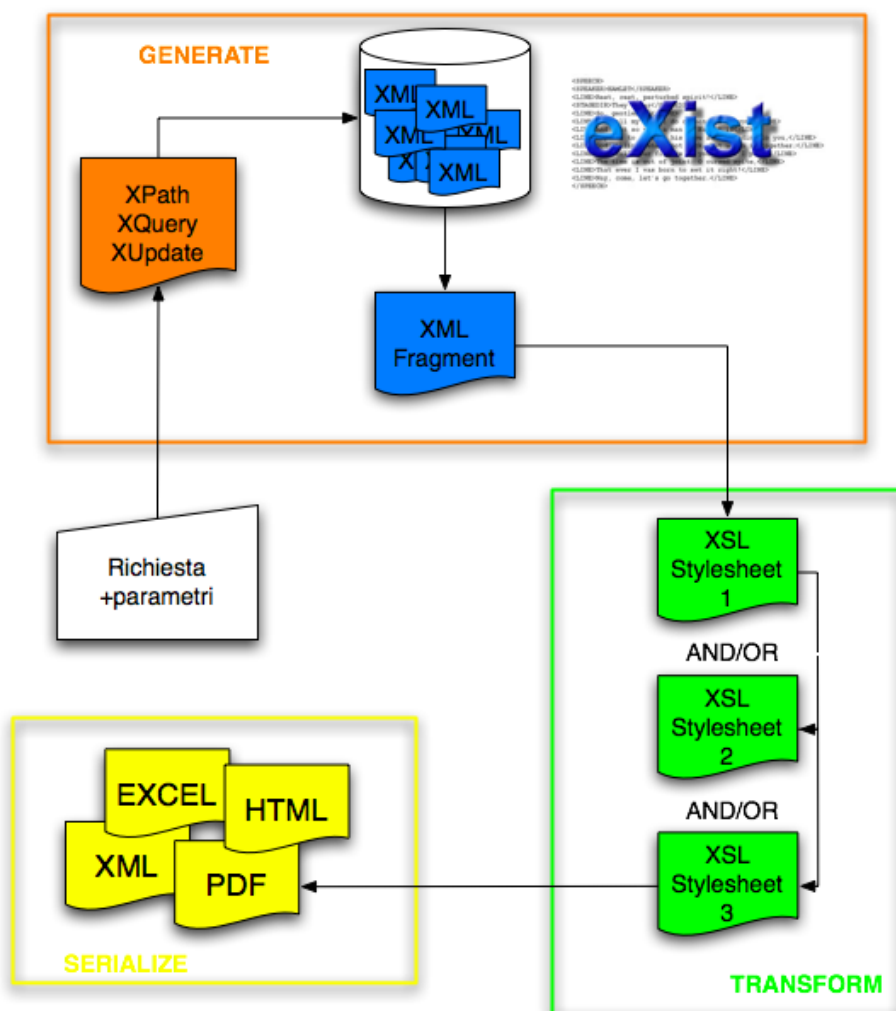
Fino a qualche tempo fa gli Open Format venivano raramente integrati nei software di uso più comune, nella maggior parte dei casi costruiti su formati proprietari; oggi la crescente esigenza di applicativi in grado di interagire con questo tipo di dati richiede la messa a punto di strumenti sempre più efficienti e in grado di adeguarsi rapidamente ai nuovi standard. L’Open Source possiede per sua natura tutte le caratteristiche necessarie per fornire supporto e interazione ai formati aperti in tutti gli scenari che prevedono il loro utilizzo. I tipici software Open Source sono, infatti, multiplatforma, di facile installazione ed utilizzo, aggiornati da una comunità di utenti/sviluppatori con cadenza periodica e spesso integrabili fra di loro per la costruzione modulare di applicativi in modo estremamente efficiente e potente, grazie anche ad una documentazione a corredo spesso chiara e completa. La possibilità di modificare il codice sorgente semplifica ulteriormente il processo di creazione e consente di adattare il software per risolvere specifici problemi. Tutte queste caratteristiche peculiari rendono l’Open Source estremamente versatile nella produzione di software *XML-oriented* e in grado di gestire tutte le fasi di organizzazione delle informazioni, dalla creazione di interfacce *stand-alone* ed *on-line* per l’acquisizione dei dati, allo sviluppo di database Open Source capaci di operare su documenti XML in modo nativo (*Xindice*, *Tamino*, *eXist*), alla creazione di software per la gestione di relazioni complesse e di ontologie (*Protégé*). La crescente diffusione di applicativi Open Source sta offrendo terreno fertile per la piena espressione delle potenzialità di tecnologie basate su XML e sui linguaggi da esso derivati.

#### 4. XML e database

Nuovi standard e nuovi formati richiedono anche nuovi sforzi da parte degli sviluppatori di software. I database tradizionali risultano spesso inadeguati per la gestione di particolari tipi di dati, quali i documenti non strutturati, soprattutto a causa della estrema rigidità della struttura relazionale che presuppone una organizzazione altrettanto rigida delle informazioni. I database più evoluti, molti dei quali sviluppati in ambito Open Source, rimangono comunque strumenti molto potenti ed affidabili soprattutto per la gestione di grandi archivi, tanto che recentemente molti tentativi sono stati fatti per renderli in grado di gestire dati XML nel loro formato nativo (Chaundri, Rashid, Zicari 2003). Spesso, tuttavia, i compromessi necessari per creare un punto di incontro fra i due mondi sono di per sé sufficienti a scoraggiare un tale tipo di implementazione: si pensi solo al continuo processo di codifica dal formato XML a quello proprio del database cui sono soggetti i dati o alla difficoltà di gestione delle relazioni spesso complesse esistenti fra i dati, soprattutto quando si ha a che fare con grandi collezioni di documenti, o ancora all'inadeguatezza dei meccanismi di query tipici del database per effettuare ricerche su documenti XML (Abiteboul, Buneman, Suciù 1999).

La necessità di uno strumento affidabile e facilmente implementabile, capace di gestire collezioni eterogenee di documenti XML, richiede l'integrazione di tecnologie di tipo database e di concetti tipici dell'*information retrieval*. Il W3C ha sviluppato una serie di linguaggi in grado di interrogare documenti XML partendo dalla loro stessa struttura: i più famosi ed utilizzati sono XQuery ed alcuni linguaggi di supporto in grado di eseguire operazioni complesse su collezioni di documenti anche molto grandi. I linguaggi di query non sono di per sé in grado di eseguire operazioni sui dati testuali: per essere applicati ai documenti necessitano, infatti, di essere interpretati da specifici software in grado di gestire tutto il processo di archiviazione e ricerca. In ambito Open Source un numero sempre crescente di strumenti di questo tipo ha visto la luce nel corso degli ultimi anni: si tratta di applicazioni in grado di gestire XML in modo nativo (*eXist* e *Xindice*) e di usare collezioni di documenti come "base di dati" in senso lato, sulle quali eseguire operazioni tipiche dei tradizionali database relazionali con un notevole grado di affidabilità. Si tratta dei cosiddetti *database XML nativi*, creati specificamente per gestire documenti XML strutturati e non strutturati senza alterarne in alcun modo la struttura.





**Figura 2.** Il meccanismo client/server e il flusso logico della pipeline di *cocoon*

## 5. L'applicativo: MAD

MAD (*Managing Archaeological Data*) è il sistema web da noi sviluppato con lo scopo di dimostrare come il software Open Source possa essere perfettamente integrabile con i linguaggi W3C per consentire una gestione razionale, flessibile e a lungo termine dei dati archeologici, rendendoli immediatamente disponibili anche online in maniera più efficiente rispetto alle soluzioni proprietarie. Lo scopo principale di

questo progetto è la creazione di un motore Open Source per la gestione di dati archeologici archiviati in un formato aperto, indipendente da qualsiasi architettura (XML), perfettamente compatibile con standard internazionali quali CIDOC-CRM ed ISO19100 e particolarmente versatile per essere utilizzato in diversi scenari, quali la creazione di archivi distribuiti consultabili online, lo sviluppo di siti web semantici, la costruzione di applicativi Web GIS.

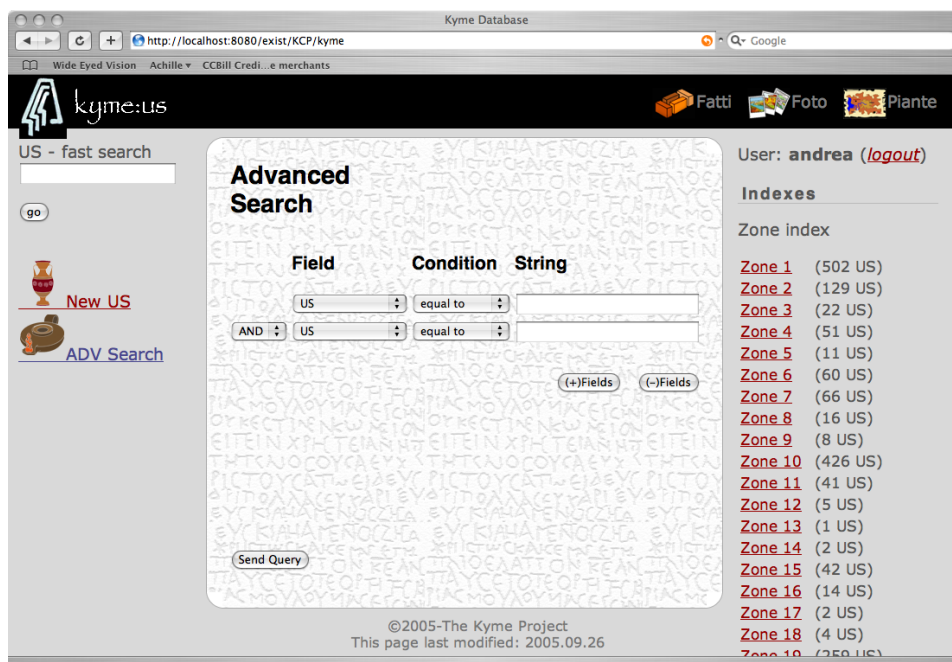
Il suo sviluppo è stato possibile grazie all'utilizzo di combinato di diversi *framework*, originariamente sviluppati dalla Comunità Open Source per svolgere specifiche funzioni, perfettamente integrati fra di loro grazie alla grande flessibilità e modularità che li caratterizza.

L'applicativo nasce dalla collaborazione fra il VAST-LAB, laboratorio di ricerca per l'utilizzo di tecnologie informatiche con sede presso il PIN (Servizi didattici e scientifici per l'Università degli Studi di Firenze - Polo Universitario di Prato) e il CISA (Centro Interdipartimentale di Servizi di Archeologia - Università di Napoli "L'Orientale") nell'ambito del progetto europeo EPOCH, un network di oltre 90 partners che ha come obiettivo la comunicazione del patrimonio culturale mediante l'utilizzo delle nuove tecnologie.

Il VAST-LAB svolge, nell'ambito del PIN, un'ampia attività di ricerca e sviluppo di soluzioni informatiche per i beni culturali, in particolare applicazioni multimediali, gestione di banche dati archeologiche, definizione ed utilizzo attivo di nuovi standard.

**5.1. eXist.** La base di tutto il sistema di gestione è il database XML nativo *eXist* (<http://exist-db.org>) adatto al nostro scopo per la sua estrema versatilità e semplicità di utilizzo. *eXist* è un Open Source scritto in Java e sviluppato da Wolfgang Meier alla Darmstadt University of Technology come un software *core* da integrare in applicativi di gestione XML sia stand-alone sia sul web. Fornisce un metodo di indicizzazione ed archiviazione di documenti organizzati in cartelle e sottocartelle e la possibilità di eseguire XQuery su documenti o parti distinte di documento, su intere cartelle o su tutto l'archivio (Meier 2003).

I dati possono, inoltre, essere validati mediante l'utilizzo di uno o più *XML Schema* o *DTD* che, una volta inclusi nel database, vengono dinamicamente assegnati mediante l'utilizzo di namespaces. Le diverse interfacce offerte dall'applicazione consentono di interrogare l'archivio utilizzando un'ampia serie di protocolli fra i quali, oltre al comune HTTP, generalmente utilizzato per le applicazioni web, anche XML-RPC, SOAP ed XML:DB, protocolli più evoluti per la consultazione di basi di dati remote. Una comoda interfaccia grafica consente inoltre



**Figura 3.** L'interfaccia web di ricerca del database di Cuma implementata con MAD

la creazione di cartelle e sottocartelle per l'archiviazione e l'organizzazione dei documenti e fornisce funzioni di backup/restore, di gestione utenti e permessi e di tutte le operazioni necessarie a rendere i dati immediatamente disponibili per successive operazioni (Fig. 1).

**5.2. Le interfacce di gestione online.** Un database XML nativo con documenti XML archiviati in una struttura a cartelle e sottocartelle si comporta esattamente allo stesso modo di un database relazionale con i dati archiviati in tabelle e record. La realizzazione di un'applicazione web richiede, tuttavia, un set di interfacce che consenta agli utenti online di lavorare con i dati in archivio. Database web-oriented come *MySQL* o *PostgreSQL* interagiscono alla perfezione con linguaggi come PHP, PERL e JSP che, combinati con CSS/XHTML e sempre più spesso con Javascript, consentono la creazione di interfacce complete e semplici da usare. XPath e XQuery sono, invece, il naturale complemento dei dati gestiti dal database XML per chi voglia realizzare una applicazione web XML-based, soprattutto grazie alla loro perfetta integrazione nel framework CSS/XHTML, con il quale condividono lo stesso modello; ad essi possono essere agilmente affiancati i fogli di stile XSLT, capaci di manipolare i dati e trasformare i contenuti XML in

qualunque altro formato. XQuery è un linguaggio molto simile a SQL sotto l'aspetto della sintassi, ma essendo stato disegnato specificamente per operare su documenti XML presenta una notevole versatilità se impiegato in strutture gerarchiche ad albero; è basato su XPath e ne supporta le stesse funzioni e operazioni. XPath fonda la sua funzionalità su sofisticate capacità di individuare e selezionare specifiche parti della struttura di un documento XML in base ad una serie di criteri stabiliti, in modo analogo alla selezione di file e cartelle in un tipico file system UNIX/LINUX. XPath è uno standard W3C dotato di una serie di circa 100 funzioni che lo rende insostituibile nella costruzione di documenti di stile XSLT per la trasformazione di documenti XML. XQuery eredita da XPath tutte le funzioni per l'identificazione di nodi, ma la sua sintassi comprende anche espressioni tipiche di linguaggi SQL che, consentendo al programmatore di scrivere query concise e facilmente leggibili, ne fanno il linguaggio più completo per la gestione di documenti XML. Nel seguente frammento di XQuery, ad esempio, si può notare come una query abbastanza potente possa essere realizzata con poche e semplici istruzioni:

```
for $i in /schedaUS
where us-princ/US = "100031"
return $i
```

la prima istruzione, infatti, esegue ricorsivamente una azione di selezione ramo XML (XPath), la seconda stabilisce la condizione di selezione e la terza restituisce il risultato all'applicazione chiamante.

Un sistema di query, per quanto complesso, non riesce comunque a fornire tutte le caratteristiche necessarie per la realizzazione di una applicazione *web-oriented* completa. L'utente necessita non solo di eseguire ricerche sui dati, ma anche di inserirne di nuovi, modificare quelli esistenti, gestire ed esportare i contenuti. eXist offre la possibilità di implementare queste funzioni in modo semplice e veloce, fornendo le interfacce per XUpdate, altro potente linguaggio basato su XPath capace di sostituire frammenti di documento con altri, di aggiungere nuove sezioni e di creare nuovi documenti. eXist include, inoltre, alcune funzioni integrabili in XQuery che consentono la creazione di nuove cartelle e l'organizzazione dei documenti in esse contenuti direttamente tramite l'interfaccia web.

**5.3. Il flusso logico e la “pipeline”.** Uno dei problemi più comuni del processo di scrittura di una applicazione web basata su contenuti XML è la separazione di fatto esistente fra il contenuto, la sua presentazione e la gestione logica di questo processo. I documenti XML,

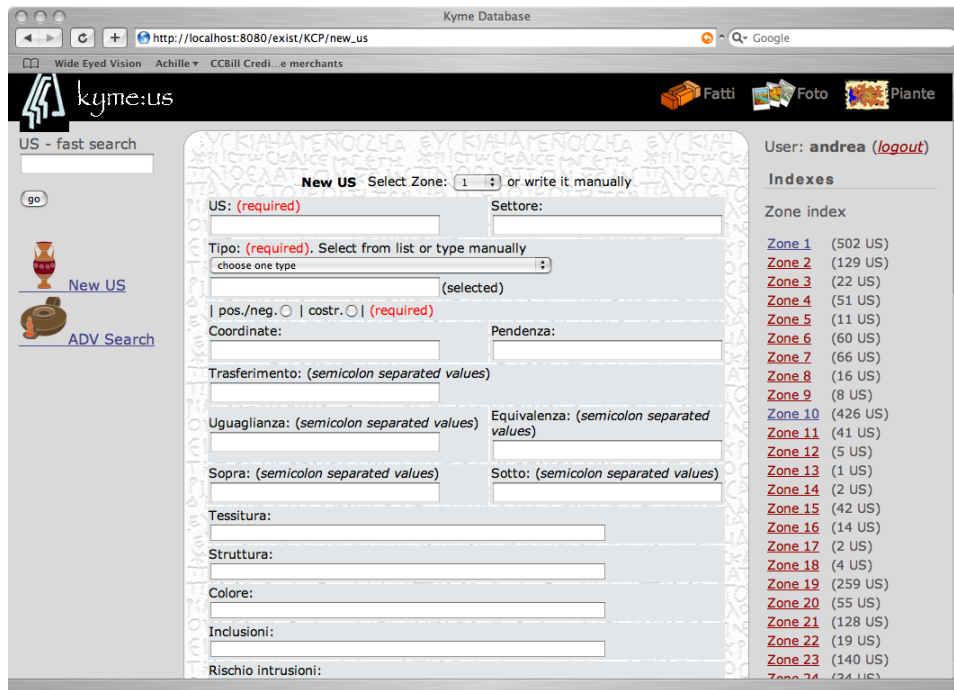
pur presentando notevoli somiglianze sintattiche con HTML - linguaggio espressamente disegnato per la presentazione di contenuti sul web - si limitano a riportare informazioni sul contenuto dei dati e nessuna specifica di stile. La presentazione avviene solitamente associando al documento XML un foglio di stile CSS o XSLT in grado di formattare i dati in base ad una serie di parametri. L'associazione di uno o più fogli di stile a documenti o a frammenti di documenti XML deve avvenire in modo dinamico secondo il tipo di contenuto da presentare o secondo specifiche indicazioni fornite dall'utente: la presentazione in una pagina web del risultato di una query e i dettagli di un documento codificati in formato PDF, ad esempio, richiedono due tipi totalmente diversi di trattamento del contenuto. Ancora, una pagina che mostri una visione tabellare di un documento necessita di un diverso foglio di stile per mostrare lo stesso documento in formato RDF. La logica che presiede all'associazione dinamica di uno o più fogli di stile al contenuto XML viene solitamente regolata da una serie di script che si occupano anche di gestire il processo di interazione client-server fra utente e database.

Il nostro applicativo utilizza il concetto di *pipeline* per implementare l'intero processo logico di intercettazione ed interpretazione della richiesta utente (HTTP Request) e di produzione, trasformazione ed invio del risultato. Il processo logico della pipeline è stato implementato utilizzando il framework *cocoon* (<http://cocoon.apache.org>) un potente motore basato su moduli Java e file di configurazione XML che consente una gestione estremamente raffinata del flusso richiesta/risposta. La pipeline di *cocoon* si basa su quattro componenti fondamentali

- un *matcher* che cerca una corrispondenza fra la stringa HTTP inviata dall'utente e una particolare entry della pipeline in modo da indirizzare il processo in una specifica direzione;
- un *generator* usato per creare un frammento XML attraverso una o più operazioni (principalmente esecuzione di XQuery);
- un *transformer* utilizzato per mappare una struttura XML di ingresso con un'altra struttura XML di uscita attraverso l'utilizzo dei fogli di stile XSLT;
- un *serializer* che spedisce il contenuto XML all'utente nel formato richiesto.

A ciascuno di questi elementi corrisponde un modulo Java che esegue fisicamente l'operazione seguendo le indicazioni della pipeline contenute in un file di configurazione (`sitemap.xmap`). Utilizzando questo file una tipica richiesta inviata da un utente tramite il browser, come la seguente

```
http://www.vast-lab.org/exist/advanced_search.xml
```



**Figura 4.** Implementazione MAD del database di Cuma: l'interfaccia per l'inserimento e la modifica di documenti

anziché produrre la visualizzazione diretta del documento `advanced_search.xml` verrà intercettata dal frammento XML della pipeline specificata dal `map:match` seguente

```
<map:match pattern="advanced_search.xml">
  <map:generate src="advanced_search.xq" type="xquery"/>
  <map:transform type="xslt"
    src="StyleSheets/advanced_search.xsl"/>
  <map:serialize type="html"/>
</map:match>
```

che innescherà l'esecuzione della corrispondente XQuery (come specificato nel `map:generate`), l'associazione dinamica di un foglio di stile al contenuto (`map:transform`) e l'invio del risultato presentato in HTML al browser (`map:serialize`): ogni successiva operazione avviene utilizzando come punto di partenza il risultato dell'operazione precedente.

In questo modo la richiesta da parte dell'utente non fa altro che attivare uno specifico set di funzioni da applicare *on-demand* al contenuto così da presentare i risultati in modo dinamico ma senza alterare la natura originaria dei dati archiviati.

La pipeline è in grado di gestire tutto il processo di comunicazione fra utente (browser), database (eXist) e script procedurali incaricati di svolgere le query e la presentazione del contenuto utilizzando XML anche per la definizione delle direttive. Questo peculiare tipo di funzionamento “a percorso” per l'estrazione del contenuto dal database e la sua trasformazione dinamica tramite fogli di stile, conferisce all'applicazione la sua caratteristica integrabilità in diversi sistemi e la capacità di erogare il contenuto in tempo reale e nella forma ottimale per soddisfare qualunque tipo di richiesta ricevuta. La potenza dei fogli di stile XSLT garantisce, infatti, la trasformazione del contenuto da XML in qualsiasi altro formato richiesto, dal semplice PDF per la distribuzione e la stampa di documenti, al nuovo formato X3D utilizzabile per la realizzazione di modelli grafici tridimensionali, all'SVG per la creazione di Web GIS basati su XML (Fig. 2).

## 6. MAD e il database archeologico di Cuma

Per testare MAD abbiamo utilizzato un set di dati archeologici forniti dal CISA acquisiti durante lo scavo della città greca di Cuma (Napoli) effettuato negli anni novanta (D'Andrea 2005).

Le informazioni relative alle unità stratigrafiche di questo scavo e tutta la documentazione a corredo erano state informatizzate utilizzando *Syslat*, un sistema di archiviazione di dati archeologici basato su *HyperCard*, un celebre programma sviluppato dalla Apple Computer. *Syslat* era sostanzialmente un software di gestione database in grado di archiviare e gestire i dati in modo semplice e flessibile grazie anche ad *HyperTalk*, un potente linguaggio di scripting con il quale era possibile creare semplici e complete interfacce. Sfortunatamente l'applicativo era privo di qualsiasi tipo di compatibilità “esterna”, essendo basato su un formato proprietario, e di qualunque possibilità di consultazione online a causa dell'assenza di interfaccia web. *HyperCard* venne inoltre ritirato dal mercato nel marzo 2004 dopo una serie di alterne vicissitudini (D'Andrea, Niccolucci 2002).

La prima necessaria operazione doveva quindi essere la riconversione in XML del vecchio archivio dello scavo archeologico, operazione che ha consentito non solo il recupero dei dati, prima confinati al solo ambito locale del computer che li ospitava, ma anche una loro codifica secondo un nuovo ed efficiente schema dati e l'inserimento nel database XML nativo eXist. Lo schema XML è stato disegnato con l'intento di rimappare la vecchia struttura *Syslat* con una nuova struttura XML che fosse compatibile con uno degli standard attualmente più importanti di

The screenshot shows the 'Kyme Database' web interface. The browser address bar displays 'http://localhost:8080/exist/KCP/us\_details?us=1032'. The page header includes the 'kyme:US' logo and navigation icons for 'Fatti', 'Foto', and 'Piante'. The user is logged in as 'andrea (logout)'. The main content area is titled 'US - fast search' and contains a search bar with a 'go' button. Below the search bar are icons for 'New US' and 'Search'. The central part of the page displays a table with the following data:

us: 1032	creusement de fosse	Tipo: POS-NEG
Settore: 2	Coordinate: Z:5.55-5.34/5.09-5.06	Pendenza: nord-sud
Sopra:	1027	
Sotto:	1023	
Anno:	1994	

The 'Descrizione:' field contains a detailed text description of the archaeological feature. Below the description, there is a table with the following data:

Sezioni:	1S33
Piante:	1P43
Autore:	F. Fratta
Foto:	17
Data:	Lun

At the bottom of the page, there are icons for 'RDF (CIDOC-CRM)', 'XML XML', and 'PDF'. On the right side, there is a 'Zone index' section with a list of zones and their corresponding US counts:

- 1 (502 US)
- 2 (129 US)
- 3 (22 US)
- 4 (51 US)
- 5 (11 US)
- 6 (60 US)
- 7 (66 US)
- 8 (16 US)
- 9 (8 US)
- 10 (426 US)
- 11 (41 US)
- 12 (5 US)
- 13 (1 US)
- 14 (2 US)
- 15 (42 US)
- 16 (14 US)
- 17 (2 US)
- 18 (4 US)
- 19 (259 US)
- 20 (55 US)
- 21 (128 US)
- 22 (19 US)
- 23 (140 US)
- 24 (34 US)

**Figura 5.** Dettagli di un documento XML del database di Cuma ed opzioni per la trasformazione del contenuto

codifica ed organizzazione dati, CIDOC-CRM, in modo da renderli immediatamente disponibili anche per i futuri scenari del Semantic Web. CIDOC-CRM è uno standard ISO internazionale sviluppato per la gestione della documentazione prodotta nell'ambito dei beni culturali, sviluppato da un gruppo interdisciplinare dell'*International Committee for Documentation of the International Council of Museums* (CIDOC/ICOM) sotto la direzione scientifica dell'ICS-FORTH. L'esportazione fisica dei dati da Syslat e la mappatura al nuovo schema sono state realizzate mediante una serie di script HyperTalk grazie ai quali è stato possibile ottenere un set di circa 3000 documenti nel nuovo formato testuale (XML), successivamente importati in eXist ed organizzati in una serie di cartelle. Lo schema XML è stato utilizzato anche per la creazione semi-automatica delle interfacce, degli script e dei fogli di stile specifici per la gestione di questo set di dati.

Una volta terminate l'archiviazione dei dati e la creazione delle interfacce secondo lo schema, l'applicazione è pronta per essere messa online ed essere utilizzata tramite un semplice browser web: la consultazione può avvenire attraverso una serie di indici creati in tempo reale



da potenti script XQuery oppure utilizzando campi di ricerca semplice o complessa mediante specifiche chiavi di ricerca ed espressioni booleane (Fig. 3). L'utente può effettuare ricerche anche su tutto l'archivio attraverso un campo presente in ogni pagina web. L'interfaccia offre inoltre la possibilità di inserire nuovi documenti e di aggiornare quelli esistenti (Fig. 4). L'interfaccia web mette anche a disposizione tutte le *features* necessarie per trasformare istantaneamente il contenuto XML in un qualsiasi altro formato semplicemente associando dinamicamente ad esso uno o più fogli di stile mediante il meccanismo della pipeline: è possibile, ad esempio, codificare un documento in PDF o trasformare l'intero contenuto in RDF includendo la descrizione completa dei dati e delle loro relazioni (Fig. 5).

## 7. Conclusioni

MAD ha dimostrato di essere un tool veloce ed affidabile in ogni fase del processo di gestione dati, soprattutto grazie alla potenza e alla solidità del database XML nativo sul quale è costruito (eXist). Il connubio fra Open Source e linguaggi standard del *World Wide Web Consortium* ha consentito la creazione di un sistema totalmente indipendente da formati proprietari, non solo per quanto riguarda il codice in cui è scritto il software utilizzato, ma anche e soprattutto per il formato realmente *open* con cui i dati sono archiviati. XML garantisce, infatti, trasparenza e completa portabilità delle informazioni per qualsiasi uso diverso da quello proposto, liberando per sempre i dati gestiti da ogni rischio di obsolescenza ed inutilizzabilità future. Gli sviluppi previsti per l'applicazione comprendono una serie funzioni per la resa grafica di alcune relazioni che l'attuale struttura dei dati è già in grado di individuare (quali ad esempio relazioni spaziali fra unità stratigrafiche) mediante l'integrazione con altri software Open Source quali *Jnet*; l'ottimizzazione del supporto per la creazione di basi di dati distribuite sul web consentirà, inoltre, una migliore integrazione fra informazioni provenienti da archivi diversi.

## 8. Bibliografia

- S. ABITEBOUL, P. BUNEMAN, D. SUCIU 1999, *Data on the Web: From Relations to Semistructured Data and XML*, Morgan Kaufmann Publishers, San Francisco, CA.
- A. B. CHAUNDRI, A. RASHID, AND R. ZICARI 2003, *XML Data Management – Native XML and XML-Enabled Database Systems*, Addison-Wesley.

- M. CRESCIOLI, A. D'ANDREA, F. NICCOLUCCI 2001, *Archaeological applications of fuzzy spatial databases and GIS*, in Z. Stancic, T. Veljanovski (eds.), *Computing Archaeology for Understanding the Past. CAA 2000. Computer Applications and Quantitative Methods in Archeology*, Proceedings of the 28 Conference, Ljubljana April 2000, BAR International Series 931, Oxford, pp. 107-115.
- M. CRESCIOLI, A. D'ANDREA, F. NICCOLUCCI 2002, *XML Encoding of Archaeological Unstructured Data*, in G. Burenhault (ed.), *Archaeological Informatics: Pushing the envelope*, Proceedings of the 29 CAA Conference, Gotland April 2001, BAR International Series 1016, Oxford, pp. 267-275.
- A. D'ANDREA, F. NICCOLUCCI 2001, *L'Informatica dell'archeologo: qualche istruzione per l'uso*, in *Archeologia e Calcolatori*, 12, pp. 199-220.
- A. D'ANDREA, F. NICCOLUCCI 2002, *Database di scavo ed Internet. L'accesso remoto all'archivio Syslat*, in B. d'Agostino, A. D'Andrea (a cura di), *CUMA. Nuove Forme di Intervento per lo Studio del Sito Antico*, Atti della Giornata di Studio, Napoli 12 Febbraio 2001, AION ArchStAnt, Quad. 14, Napoli, pp. 167-175.
- A. D'ANDREA 2004, *'L'entropia dell'Archeologia Computazionale. Ovvero dall'ordine al disordine'*, in *Archeologia e Calcolatori*, 15, pp. 219-238.
- A. D'ANDREA 2005, *Metodologie informatiche integrate per lo studio dell'insediamento antico di Cuma*, in B. d'Agostino, F. Fratta, V. Malpele (a cura di), *CUMA. Le Fortificazioni. 1. Lo scavo 1994-2002*, AION ArchStAnt, Quad. 15, Napoli, pp. 251-265.
- A. D'ANDREA 2006, *A preliminary Ontology-based model applied to the description/interpretation of the archaeological excavation*, in W. von Hahn, C. Vertan (Eds.), *Proceedings of the First International Workshop on "Ontology Based Modelling in The Humanities"*, 7-8 April 2006, University of Hamburg, Bericht 264, pp. 38-46.
- W. MEIER 2003, *eXist: An Open Source Native XML Database*, in *Web, Web-Services and Database Systems, NODE 2002, Web and Database-Related Workshops*, Erfurt, Germany, October 7-10, 2002, pages 169-183, Lecture Notes in Computer Science 2593, Springer 2003.
- NICCOLUCCI F. (ED.) 2002, *Dalla Fonte alla Rete: Il linguaggio XML e la codifica dei documenti storici, archeologici e archivistici*. Numero monografico del Bollettino del CRIBECU - Scuola Normale Superiore di Pisa, 12.

- NICCOLUCCI, F. 2002, *XML and the Future of Humanities Computing*, Applied Computing Review, vol. 10, n.1, 43-47.