Archeologia e Calcolatori
12, 2001, 55-69

PROGETTO CAERE: PROSPETTIVE DI APPLICAZIONE DEGLI STANDARD INTERNAZIONALI PER LA CODIFICA DEI DATI TESTUALI

1. La scelta dello Standard Generalized Markup Language (SGML)

Uno degli aspetti di maggiore originalità nell'ambito del Progetto Caere è stata la scelta di informatizzare i diari di scavo utilizzando la codifica SGML; un procedimento che non solo ha costituito una sperimentazione innovativa ma che ha aperto anche numerose prospettive di sviluppo nelle applicazioni informatiche alla ricerca archeologica. Quando si è deciso di analizzare i diari manoscritti redatti nel corso delle campagne di scavo condotte a Caere nell'area della Vigna Parrocchiale tra gli anni 1983 e 1989 per estrarne i dati utili allo studio della topografia e dei materiali, è stato subito evidente che non era possibile inserire tutte le informazioni in essi contenute all'interno di un database. La struttura rigida del record, infatti, non si adattava alla natura inevitabilmente non sistematica dei diari che sono stati scritti in linguaggio naturale, nel corso dello scavo, rispettando solo parzialmente un modello o uno schema di codifica predefiniti. Di qui la decisione di trasformare i diari in documenti elettronici senza modificarne la struttura interna, garantendo allo stesso tempo la possibilità di interrogarli come se i dati fossero inseriti in un database.

A questo scopo sono stati valutati diversi formati largamente utilizzati per l'informatizzazione di documenti e la pubblicazione su supporto digitale. Il loro limite, dal nostro punto di vista, è quello di essere tutti mirati alla codifica "tipografica"; in altre parole i programmi consentono di impaginare e riprodurre le caratteristiche tipografiche dell'originale cartaceo mediante una codifica proprietaria che non tiene conto del contenuto, cioè del valore semantico delle stringhe di caratteri. Le interrogazioni possibili in questi documenti sono di due tipi: cercare una stringa di testo indipendentemente dal contesto in cui è inserita; ovvero cercare caratteristiche tipografiche. Nel primo caso si potrebbe ottenere una risposta ridondante, specialmente qualora si interroghino documenti lunghi; nel secondo caso il risultato potrebbe essere inutilizzabile, in quanto non esiste una correlazione univoca tra stile tipografico e contenuto semantico [1].

Lo SGML si è dimostrato invece lo strumento più adatto al raggiungimento dell'obiettivo perseguito (Adamo 1996; Bonincontro 1997). Si tratta

[1] Il corsivo, ad esempio, può essere assegnato al titolo di un articolo, alla citazione tratta da un altro testo, ad una parola straniera. Una ricerca che ha per chiave il "corsivo" avrà dunque come risultato tre informazioni diverse.

infatti di un linguaggio di codifica per documenti elettronici, finalizzato prima di tutto alla descrizione della struttura interna dal punto di vista dei dati e delle informazioni contenute nel testo, non dell'aspetto tipografico che gli stessi dati devono assumere nel momento della pubblicazione. In un documento SGML stringhe di testo con diverso contenuto semantico hanno una codifica diversa anche se condividono le medesime caratteristiche tipografiche.

Si è dunque stabilito di elaborare un modello descrittivo dei diari secondo le regole di questo standard. Ciò ha richiesto un lungo periodo di analisi dei testi, che ha prodotto un primo modello piuttosto semplice al quale sono stati apportati modifiche e miglioramenti in interventi successivi, procedendo in parallelo con la trascrizione e la codifica dei diari. L'analisi preliminare, infatti, non era stata sufficiente a individuare tutti gli elementi strutturali interessanti ai fini dell'indagine archeologica.

Sullo SGML è oggi disponibile una discreta bibliografia e molto materiale disperso su numerosi siti Internet, ufficiali [2] e non. Fino a non molti anni fa esisteva solamente il documento ufficiale ISO [3]; furono poi pubblicati i primi manuali, in particolare quello di Charles Goldfarb (1990), il "padre" di SGML, che guidò per anni il gruppo di ricerca al cui lavoro si deve l'elaborazione di un primo sistema di codifica denominato Generalized Markup Language, dal quale fu poi ricavato lo standard. Recentemente la situazione è cambiata perché SGML è diventato uno strumento conosciuto grazie alla diffusione mondiale di una sua applicazione, l'HyperText Markup Language (HTML), cioè il linguaggio di codifica utilizzato per la produzione delle pagine web.

La sintassi SGML è molto complessa, per lo meno in certi suoi aspetti, e ciò ha inevitabilmente costituito un ostacolo alla diffusione di questo linguaggio, anche se le sue potenzialità sono ora comunemente riconosciute. La complessità si riflette nella limitata disponibilità di programmi in grado di gestire documenti SGML; programmi per altro molto costosi e dunque diffusi più in contesti commerciali che negli ambienti di ricerca, da sempre penalizzati dalla carenza di fondi [4].

[2] http://www.sgmlopen.org/; http://www.oasis-open.org/cover/sgml-xml.html; http://etext.lib.virginia.edu/sgml.html; http://www.sgmlsource.com/

[3] L'International Standard Organization è responsabile della pubblicazione dello standard. Il primo documento ufficiale fu pubblicato nel 1986 con il titolo *Information Processing - Text and Office System - Standard Generalized Markup Language (SGML)*, documento n. 8879-1986.

[4] Non è possibile in questa sede presentare in modo esaustivo la sintassi SGML. Si rimanda pertanto ai manuali specifici. Per consentire anche a coloro che non hanno alcuna conoscenza di questo linguaggio di seguire la descrizione del presente progetto, si ricorda che un documento SGML è costituito da un modello descrittivo della sua struttura interna, chiamato Document Type Definition (DTD), e dal testo codificato. La DTD contiene le dichiarazioni di tutti i codici usati per una determinata tipologia di documenti; gli stessi codici vengono inseriti nel testo in modo da delimitarne e quindi definirne le componenti strutturali. I codici sono chiamati *tag* nella sintassi SGML; le componenti strutturali del documento che vengono delimitate dai tag sono chiamate *elementi*.

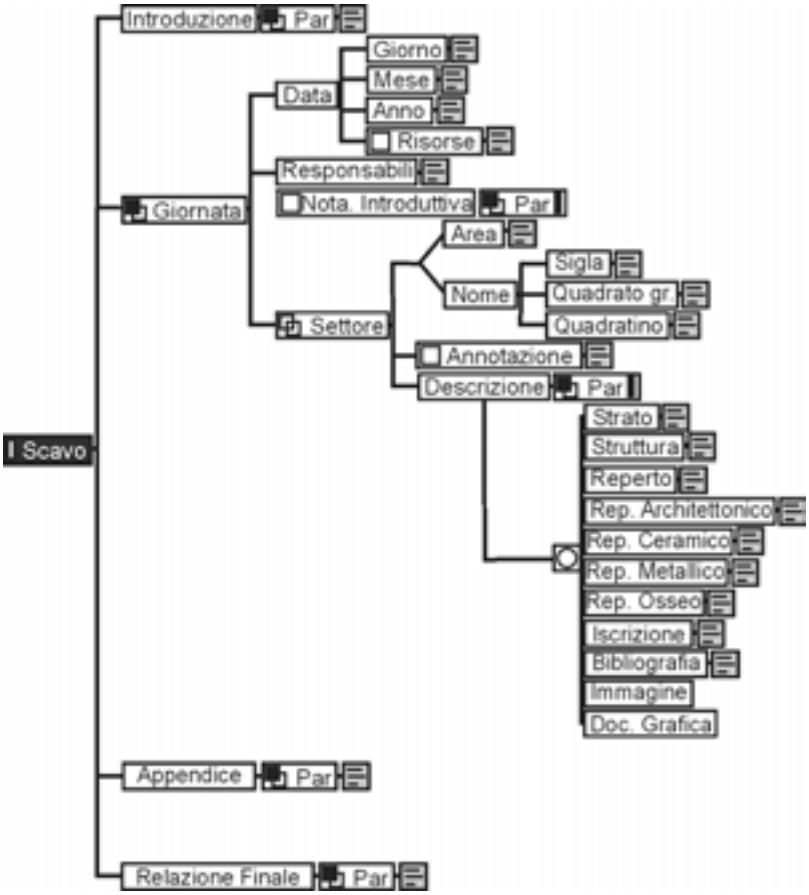

Fig. 1 – Schema riassuntivo della Document Type Definition (DTD).

2. Elaborazione di una DTD per i diari di scavo

Il modello messo a punto per l'informatizzazione dei diari di scavo non è molto complesso (Fig. 1). La struttura principale è la giornata di scavo, al cui interno è contenuta la descrizione del settore di scavo, delle strutture murarie rinvenute e dei reperti (Moscati, Mariotti, Limata 1999). Poiché la descrizione non è schematica ma discorsiva e i riferimenti alle strutture e ai reperti sono inseriti liberamente all'interno del testo, è stato previsto un certo numero di "elementi liberi" nella struttura del documento. Si definiscono "liberi" gli elementi che non sono inseriti nella struttura gerarchica che caratterizza ogni documento SGML, in quanto essi ricorrono senza un ordine preciso. Nel caso presente ogni "giornata" contiene, all'interno della "descri-

zione" del lavoro svolto, elementi liberi quali, ad esempio, "reperto" o "struttura muraria", che possono ricorrere senza un ordine preciso nei paragrafi che costituiscono la "descrizione" stessa.

Come si è detto, più volte sono state apportate modifiche alla DTD, perché nel corso del lavoro sono emerse caratteristiche testuali nuove, non presenti nei documenti campione utilizzati per disegnare il modello. Inoltre, per consentire l'inserimento dei diversi elementi in qualsiasi contesto, è stata elaborata una DTD con struttura piuttosto libera e di conseguenza gestibile con maggiore difficoltà dai programmi di codifica e di interrogazione. Si tratta di una situazione piuttosto frequente quando si codificano documenti preesistenti, soprattutto quando essi sono stati composti senza una struttura rigida di riferimento. In casi particolarmente complessi può risultare addirittura impossibile disegnare una DTD; ciò si verifica quando la struttura interna non è coerente e dunque non riducibile ad una seppure minima gerarchia di elementi.

Questo modo di procedere (elaborazione di un modello iniziale e successive modifiche) comporta a volte il rischio di perdere di vista la struttura generale del documento e di puntare l'attenzione solamente sulle singole informazioni presenti nei diversi contesti. Si deve, invece, tenere sempre presente che in un testo possono esserci numerose ripetizioni e che le informazioni ripetute possono avere diverso peso a seconda del contesto che le racchiude: un dato ripetuto in più luoghi non deve essere codificato ogni volta. Si consideri inoltre che l'operazione di codifica può diventare quasi impossibile se si evidenzia ogni microstruttura presente nel documento in modo acritico: di qui la necessità che il codificatore sia al contempo un archeologo, il cui contributo risulta di importanza primaria ai fini dell'esattezza delle procedure.

Altra conseguenza di una codifica estremamente fitta riguarda l'analisi successiva del documento; la risposta ad una ricerca infatti può essere eccessivamente lunga e ridondante. Per chiarire meglio il problema possiamo fare un confronto con l'indice analitico di un volume; la scelta dei termini da introdurre nell'indice deve essere ragionata perché sia di qualche utilità: ad esempio, se in una parte del testo si dà qualche informazione relativa ad una località inserita nell'indice topografico, sarà opportuno aggiungere tale località nell'indice analitico; se invece la stessa località è solo citata per un parallelo con altre località, sarà meglio escluderla.

3. Prime prove di interrogazione

Nel caso ceretano, una volta completata la trascrizione e la codifica dei diari delle prime campagne di scavo, è stata verificata la funzionalità del modello SGML provando ad interrogare il documento elettronico sulla base

Fig. 2 – Primo esempio di interrogazione on-line dei dati di scavo.

di alcuni elementi (Fig. 2). Il sistema operativo UNIX è stato molto utile; il linguaggio di programmazione disponibile nella shell di lavoro può essere facilmente sfruttato per scrivere brevi programmi richiamabili direttamente dalle pagine web. In questo modo si è verificato che i risultati delle estrazioni di dati in base alla codifica SGML erano coerenti e significativi e dunque costituivano un nuovo percorso di lettura dei diari. La codifica SGML, infatti, trasforma naturalmente il testo in un ipertesto, moltiplica le chiavi di accesso alle informazioni, pone in luce connessioni non evidenti nella lettura tradizionale di tipo sequenziale, consente di associare informazioni presenti in luoghi diversi del documento.

Per realizzare un sistema di interrogazione dei documenti SGML più completo e più duttile, adatto a gestire la complessità delle relazioni degli elementi del testo, capace di integrarsi con i programmi di gestione della cartografia digitale, pronto a divenire veicolo di diffusione delle informazioni oltre che di studio, è stata poi realizzata un'applicazione Internet basata sulla tecnologia ASP (Active Server Pages).

L'applicazione è un sito web attivo (cfr. BARCHESI, in questo volume), in grado di gestire l'insieme dei testi dei diari SGML e operarvi ricerche complesse sulla base di filtri cronologici e topografici. Per meglio adattarsi alla mutevole terminologia adottata nei testi in riferimento a determinate aree di scavo o strutture, il motore di ricerca sfrutta un database Access collegato, nelle cui tabelle sono descritte le relazioni cronologiche e topografiche di tutti gli elementi SGML interrogabili e la forma con cui essi appaiono nel testo dei diari. Tutto il sistema – programma di gestione della cartografia e applicazione per l'Information Retrieval dai testi – comunica interattivamente sulla base della tecnologia Internet ASP. L'applicazione può essere utilizzata con qualsiasi Browser Internet commerciale.

4. EXTENSIBLE MARKUP LANGUAGE (XML)

L'informatizzazione dei diari di scavo fa parte di un più ampio progetto di studio dell'area archeologica della Vigna Parrocchiale, situata nel cuore dell'antica Caere. L'archivio testuale è stato quindi affiancato da un archivio grafico e fotografico e da un GIS, che consente la distribuzione di mappe dinamiche attraverso la rete (cfr. CECCARELLI, in questo volume). La scelta della rete come canale preferenziale per la diffusione delle informazioni ha reso necessario uno studio di fattibilità per una eventuale conversione dei documenti SGML già realizzati in documenti XML.

L'XML si sta infatti affermando come nuovo linguaggio per lo scambio di informazioni via rete e negli ultimi anni tende a sostituire SGML, dal quale deriva; inoltre, i più diffusi software GIS sono in grado di riconoscere e gestire documenti XML. L'evoluzione tecnologica in corso non può certo essere ignorata, anche se non si ritiene necessario operare immediatamente una conversione dell'archivio testuale. Nella sostanza XML non comporta grandi modifiche nel formato dei documenti codificati; il nuovo linguaggio è sostanzialmente una semplificazione del precedente SGML, rispetto al quale ha perso alcune caratteristiche che risultavano difficili da gestire automaticamente. I diari di scavo in formato SGML sono compatibili con il nuovo formato; la conversione non richiede dunque nessuna modifica strutturale della codifica interna. Essa comunque è stata prevista nell'eventualità in cui si decida di rendere disponibili sul sito web i documenti elettronici comprensivi del markup effettuato, utilizzando XML come standard di interscambio con altre istituzioni.

All'interno del gruppo di ricerca che lavora al progetto si è dunque deciso che l'informatizzazione dei diari può proseguire utilizzando il formato SGML e l'editor scelto per digitare e codificare i testi, così da conservare procedure ormai testate e stabilizzate. I documenti potranno poi essere convertiti in una fase conclusiva, al termine dell'informatizzazione di tutte le campagne di scavo. Per quanto riguarda la DTD è già pronta una versione

XML che conserva la medesima struttura gerarchica della DTD SGML; sono state solo apportate alcune modifiche allo scopo di allineare tutte le dichiarazioni di elementi con la sintassi XML.

Ci si può chiedere comunque cosa abbia determinato la necessità di un nuovo linguaggio di codifica per il mondo web. Nel corso degli anni gli utenti Internet hanno richiesto sempre più efficienza; tutti cercano, infatti, un sistema in grado di capire la natura delle informazioni che la rete veicola, in modo da rispondere alle richieste in modo selettivo. HTML non è stato disegnato per questo scopo; esso è piuttosto un linguaggio WYSIWYG, pensato per la composizione di pagine web, non per automatizzare procedimenti di indagine attraverso gli archivi disponibili in rete. Per questo si è ritenuto necessario definire un insieme di codici che descriva la natura delle informazioni, cioè un linguaggio che descriva i dati dal punto di vista della loro tipologia e non dell'aspetto grafico che essi devono assumere sulla pagina web.

Uno standard utile a questo fine già esisteva: SGML. La sua eccessiva complessità ha però indotto il World Wide Web Consortium (W3C) a sviluppare tra il 1996 e il 1998 un nuovo linguaggio in grado di descrivere le informazioni che viaggiano attraverso la rete. La pubblicazione della prima versione dell'XML è del 1998 (W3C XML 1.0 Recommendation). Successivamente si è lavorato su aspetti particolari del nuovo linguaggio di codifica, in particolare sui fogli di stile e sull'implementazione di un linguaggio per interrogare documenti XML (XML query language).

XML è un linguaggio di codifica derivato dallo SGML allo scopo di semplificare e quindi diffondere l'uso di questo tipo di codifica nel campo dell'editoria elettronica. XML si è inoltre rivelato un ottimo formato di scambio tra applicativi diversi e questa sua caratteristica ne sta determinando una sempre maggiore diffusione. Le principali doti di XML non sono pertanto una novità per quanti hanno utilizzato in passato SGML come linguaggio di codifica e pubblicazione di documenti elettronici; inoltre, alcune qualità attribuite a XML possono facilmente essere ricondotte a SGML:

1) **XML consente la pubblicazione di documenti elettronici indipendenti dai media di trasmissione e memorizzazione.**

   SGML in quanto standard pubblicato dall'International Standard Organization, è un formato *pubblico* che nessuno può modificare e adattare alle proprie esigenze. Un documento SGML è sempre accompagnato dalla dichiarazione della sua struttura interna (DTD) formulata secondo le regole dello standard in modo che chiunque possa riconoscere tale struttura e gestirla automaticamente mediante programmi appositamente sviluppati. Gli utilizzatori di SGML, desiderosi di svincolarsi dal controllo delle grandi holding produttrici di hardware e di software, hanno da sempre puntato su questa fondamentale caratteristica.

2) **XML consente di implementare protocolli per lo scambio dei dati, indipendenti dalle piattaforme soprattutto nel settore del commercio elettronico.**
   Anche in questo caso non si tratta di una novità: se per protocolli intendiamo in effetti modelli secondo i quali strutturare i dati da esportare/importare, un software può essere progettato in modo tale da gestire la DTD per acquisire i dati in arrivo e convertirli nel suo formato proprietario o viceversa. Questa possibilità era già offerta da SGML.
3) **XML consente di mostrare le informazioni con un'impaginazione libera e personale grazie all'utilizzo dei fogli di stile.**
   Lo standard SGML è finalizzato alla codifica dei documenti dal punto di vista delle informazioni in essi contenute, indipendentemente dall'aspetto grafico che devono presentare. Mediante i fogli di stile uno specifico software può assegnare ad ogni componente strutturale del documento le caratteristiche tipografiche che l'utente ritiene più adatte. L'utente può cambiare quando vuole l'aspetto del documento senza dover intervenire sulla codifica.
4) **XML consente di inserire nel documento *metadati*, cioè informazioni ulteriori relative ai dati contenuti nel documento ma non presenti nel documento stesso.**
   Il meccanismo che permette tale inserimento è quello degli attributi che contengono qualificazioni utili a distinguere le numerose occorrenze di un medesimo elemento. Gli attributi sono naturalmente presenti nello standard SGML.

XML sta avendo oggi una diffusione notevole, decisamente maggiore rispetto allo standard SGML da cui deriva e dal quale ha ereditato le sue principali caratteristiche, proprio grazie alla semplificazione delle complesse regole dello SGML, che lo ha reso adatto al mondo web. La vera novità è quindi che XML consente di sviluppare software in grado di gestire informazioni distribuite via web.

È importante sottolineare che XML non può essere posto sullo stesso piano dello HTML, il linguaggio correntemente utilizzato per la realizzazione delle pagine web. Come si è detto, HTML è soltanto un'applicazione dello SGML, cioè un modello costruito secondo le regole dello standard, ma non è un linguaggio; tutti i browser sono in grado di riconoscere, gestire, impaginare un documento HTML in quanto conoscono la sua DTD. XML è invece un linguaggio nuovo, un insieme di regole utili per la costruzione di modelli di documenti e solo i browser più recenti sono in grado di interpretare documenti XML.

Analizziamo ora con più attenzione le differenze tra i due linguaggi SGML e XML. Esistono differenze nel formalismo che regola la scrittura di una DTD e differenze nel tipo di dichiarazioni previste. Quando si converte una DTD da SGML a XML, si possono eseguire modifiche automatiche per risolvere alcune di queste differenze:

a) Omissione dei tag

In un documento SGML ogni oggetto testuale significativo viene codificato delimitandolo con due codici: il tag di apertura (open-tag) e il tag di chiusura (end-tag). La sintassi SGML prevede dei casi in cui si possono omettere l'uno o l'altro o entrambi. Ad esempio, secondo la dichiarazione seguente, l'elemento "autore" è costituito da una sequenza di due elementi, "cognome" e "nome", che devono ricorrere al suo interno una sola volta e nell'ordine indicato:

```
<!ELEMENT AUTORE -- (COGNOME, NOME) -->
```

Il tag di chiusura dell'elemento "cognome" può essere omesso dal momento che la fine di questo elemento è implicita nell'apertura dell'elemento successivo "nome"; secondo la dichiarazione, infatti, "nome" segue "cognome". Il linguaggio XML, al contrario, non consente omissione di tag.

b) Dichiarazioni di tipi di elemento o di attributi raggruppati

In una DTD SGML è prevista la possibilità di dichiarare attributi una sola volta per un gruppo di elementi, in modo da non ripetere più volte dichiarazioni identiche.

```
                ELEMENTS        ATT_NAME          VALUE
<!ATTLIST REPERTO, STRUTTURA DATAZIONE CDATA #IMPLIED>
```

Nell'esempio l'attributo "datazione" è valido per i due elementi "reperto" e "struttura". Il suo valore deve essere costituito da *caratteri* (CDATA = *character data*) e non deve essere obbligatoriamente inserito per ogni occorrenza degli elementi "reperto" o "struttura" (# IMPLIED). Nella DTD XML è invece necessario inserire una dichiarazione diversa per ogni elemento.

c) Commenti inseriti nelle dichiarazioni

I commenti consentono di inserire nella DTD informazioni sulla codifica che devono essere ignorate dal programma che la elabora. I commenti possono comparire in qualunque punto della DTD. La sintassi SGML consente di inserire un commento anche in una dichiarazione; nella DTD XML i commenti devono invece essere separati e inseriti in apposite dichiarazioni di commento. Ad esempio, nella DTD SGML la dichiarazione dell'elemento "struttura", dichiarato come #PCDATA (= *parsed character data*), può essere spiegata da una nota contenuta nella stessa riga:

```
<!ELEMENT STRUTTURA -- (#PCDATA) -- si intende struttura muraria -->
```

Nella DTD XML si devono separare le due informazioni:

```
 <!ELEMENT STRUTTURA -- (#PCDATA) >
<!-- si intende struttura muraria -->
```

Vi sono tuttavia altre differenze tra le due sintassi che non possono essere risolte automaticamente. L'autore della DTD deve intervenire manualmente.

a) Tipi di attributo previsti da SGML, eliminati in XML.

Nella sintassi dello standard esistono delle parole riservate che possono essere usate in una dichiarazione di elemento o di attributo, che però sono state eliminate nella sintassi XML. NAME indica una stringa che può contenere caratteri, numeri, trattini, caratteri di sottolineatura, due punti o punti ma deve cominciare sempre con una lettera. NMTOKEN (name token) indica lo stesso tipo di stringa con la differenza che può cominciare anche con un punto, un trattino, un numero. NUMBER indica una stringa che può contenere numeri, trattini, punti ma deve cominciare sempre con un numero.

La sintassi XML non consente di definire il contenuto di un attributo come NAME e NAMES, NUMBER e NUMBERS. In caso di conversione da un sistema all'altro si deve valutare caso per caso come sostituire questo tipo di valori.

Nella dichiarazione di attributo si può inserire un valore predefinito. Ad esempio la dichiarazione:

```
<!ATTILIST REPERTO TIPO CDATA "CERAMICO">
```

assegna un atributo "tipo" all'elemento "reperto" il cui valore deve essere costituito da *caratteri*. Se non viene definito l'attributo, il suo valore sarà per default "ceramico". La sintassi SGML consente di inserire la definizione #CURRENT in questa colonna:

```
<!ATTILIST REPERTO STRATO CDATA #CURRENT >
```

Secondo questa dichiarazione l'elemento "reperto" ha un attributo "strato" il cui valore deve essere costituito da *caratteri*; se non viene definito l'attributo per un reperto, si intende che il suo valore sia l'ultimo definito. Questo tipo di dichiarazione non è ammessa in XML; in genere si sostituisce #REQUIRED a #CURRENT indicando la definizione dell'attributo come obbligatoria. Tuttavia la regola non è sempre valida.

b) Dichiarazioni di contenuto per un elemento, non più utilizzabili.

Altre parole riservate della sintassi SGML sono CDATA *(character data)* e RCDATA (*replaceable character data)* che indicano due possibili contenuti di una dichiarazione di elemento. XML non ammette nella dichiarazione di un elemento questi valori. Nella maggior parte dei casi è possibile sostituirli con #PCDATA (*parsed character data)*.

c) Inclusione ed esclusione.

Nel linguaggio SGML è possibile definire un elemento come "inclusive" rispetto ad un nodo dell'albero gerarchico; ciò significa che tale elemento può

ricorrere in qualsiasi posizione nel nodo e in tutti i livelli sotto tale nodo. Al contrario un elemento definito come "exclusive" rispetto ad un nodo dell'albero non può essere usato all'interno del medesimo nodo. I due meccanismi non sono presenti in XML e devono essere sostituiti; in certi casi l'eliminazione è facile, in altri richiede una modifica della struttura generale della DTD[5].

d) Operatore AND.

In una dichiarazione di elemento XML non è ammesso l'operatore AND (&). Questo operatore collega più elementi senza definire un ordine obbligatorio di sequenza; se gli elementi collegati sono pochi si possono inserire nella dichiarazione tutte le combinazioni possibili:

```
<!ELEMENT DESCRIZIONE -- (REPERTO & STRUTTURA) -->
```

diventa

```
<!ELEMENT DESCRIZIONE--(((REPERTO, STRUTTURA)|(STRUTTURA, REPERTO))*)-->
```

La conversione non è possibile se l'operatore AND connette un numero troppo grande di elementi in quanto le combinazioni possibili diventano troppe e ingestibili. Proprio questo è del resto il motivo per il quale l'operatore AND è stato eliminato dalla sintassi XML; dunque sarà necessario risolvere diversamente il problema, definendo un ordine più rigido tra gli elementi.

Torniamo brevemente sul caso delle inclusioni e delle esclusioni. Come eliminarle da una DTD SGML nella fase di conversione verso XML? Le inclusioni a livello basso in una DTD SGML si possono eliminare, introducendo nella dichiarazione del nodo un modello misto che comprenda anche l'elemento precedentemente definito come "inclusive". Prendendo l'esempio della DTD implementata per i diari di scavo di Caere, esistono elementi che possono essere liberamente usati nei paragrafi di testo che costituiscono la descrizione di una giornata di scavo: tra questi elementi vi sono ad esempio i vari tipi di reperti. Al posto della dichiarazione SGML.

```
<!ELEMENT PAR -- (#PCDATA) + REPERTO_METALLICO, REPERTO_CERAMICO -->
```

inseriamo una nuova dichiarazione conforme a XML

```
<!ELEMENT PAR--((#PCDATA|REPERTO_METALLICO|REPERTO_CERAMICO)*)-->
```

Le esclusioni possono essere semplicemente eliminate confidando sul fatto che certe situazioni non si verifichino se un documento viene elaborato secondo buone norme editoriali, anche se non viene esplicitamente vietato l'uso di determinati elementi all'interno di altri. Altrimenti si possono chiamare gli stessi elementi in due modi diversi secondo la loro posizione, per

[5] Cfr. il paragrafo successivo per un approfondimento sulla risoluzione di casi di elementi "inclusive" nella conversione di una DTD SGML nel formato XML.

evitare che l'uso di certi sottoelementi in posizioni logicamente non ammissibili sia lecito.

Un esempio: se un paragrafo è costituito da semplice testo (#PCDATA) e annotazioni, e un'annotazione a sua volta contiene uno o più paragrafi, il parser ammette che all'interno di un'annotazione venga inserito un paragrafo e al suo interno un'annotazione. Converrà differenziare con due nomi diversi i paragrafi che costituiscono il testo principale da quelli che costituiscono un'annotazione e dichiarare l'annotazione solamente nel contenuto del paragrafo principale.

Come si è già detto, la DTD SGML per i diari di scavo presenta una struttura facilmente convertibile in XML; sono state verificate tutte le dichiarazioni e, se necessario, si potrà intervenire sui testi già codificati con minime correzioni per renderli del tutto compatibili con XML. Operare la conversione in questa fase ci costringerebbe a cambiare il programma usato attualmente per la codifica; per questo si è ritenuto più economico procedere senza cambiamenti almeno fino a che tutti i diari non saranno trascritti e codificati.

5. Future applicazioni della DTD dei diari di scavo

Il progetto di codifica dei diari di scavo non si ferma a questo risultato, di per sé importante, che permette di rileggere i testi secondo nuove prospettive e di metterli in stretta relazione sia con l'indagine topografica del sito sia con lo studio dei reperti. Proprio lo studio dei reperti costituisce l'occasione per un approfondimento della ricerca (Moscati c.s.). Nei diari i reperti vengono spesso elencati nella descrizione dei lavori di una giornata, ma non descritti in modo accurato. Disponiamo tuttavia di molti dati specifici sui materiali elencati grazie alla catalogazione che è stata successivamente eseguita. Si tratta quindi di mettere in relazione le schede descrittive dei materiali con il testo dei diari.

Accanto a una procedura tradizionale di schedatura informatizzata dei dati all'interno di un database relazionale, si prevede al contempo l'elaborazione di altre DTD che consentano di informatizzare e codificare la descrizione dei singoli reperti. Nel caso specifico dello scavo della Vigna Parrocchiale è prevista la realizzazione di un archivio testuale contenente documenti "interrogabili", con struttura diversa ma legati tra loro mediante un sistema di link attraverso il quale l'utente potrà navigare approfondendo, quando lo ritiene utile, l'analisi dei materiali insieme allo studio topografico del sito scavato. L'archivio testuale si dovrà dunque configurare come un ipertesto; grazie all'inserimento delle immagini e di un ulteriore collegamento con il GIS si otterrà un'applicazione multimediale funzionale alla conoscenza del sito archeologico e all'analisi sia delle strutture murarie sia dei materiali rinvenuti.

Torniamo ora all'idea di un sistema di modelli diversi per la codifica di documenti; questi documenti saranno collegati tra loro e i collegamenti po-

```
<!ELEMENT Reperto.ceramico - - (US, classe?, forma?, tipo?, descrizione?,
                        argilla?, rivestimento?, motivo.soggetto?, tecnica?, iscrizione?, stato?,
                        misure?, datazione?, disegno*, foto*, diapositiva*, bibliografia?, note?,
                        responsabile?, collocazione?, restauri?) -- >
<!ATTLIST Reperto.ceramico
                         inv.nr NUMBER  #IMPLIED
                         id.nr ID  #IMPLIED
                          -- >
<!ELEMENT US - - (#PCDATA) -- Unità stratigrafica -->
<!ELEMENT classe - - (#PCDATA) -- Classe ceramica -->
<!ELEMENT forma - - (#PCDATA) -- Forma del reperto-->
<!ELEMENT tipo - - (#PCDATA) -- Tipologia -->
<!ELEMENT descrizione - - (#PCDATA) -- Descrizione libera-->
<!ELEMENT argilla - - (colore | (#PCDATA))+ -- Dati sull'argilla-->
<!ELEMENT colore - - (#PCDATA) -- Definizione del colore-->
<!ELEMENT rivestimento - - (tipo | colore | (#PCDATA))+ -- Dati sul rivestimento -->
<!ELEMENT decorazione - - (motivo.soggetto | tecnica | posizione)+ -- Dati sulla decorazione -->
<!ELEMENT motivo.soggetto - - (#PCDATA) -- Soggetto decorativo-->
<!ELEMENT tecnica - - (#PCDATA) -- Tecnica di esecuzione -->
<!ELEMENT posizione - - (#PCDATA) -- Posizione sulla superficie -->
<!ELEMENT iscrizione - - (lingua | tecnica | posizione | tipo | note)+ -- Dati sull'iscrizione -->
<!ELEMENT lingua - - (#PCDATA) -- Lingua dell'iscrizione -->
<!ELEMENT note - - (#PCDATA) -- Annotazioni -->
<!ELEMENT stato - - (#PCDATA) -- Stato di conservazione -->
<!ELEMENT misure - - (#PCDATA) -- Misure del reperto -->
<!ELEMENT datazione - - (#PCDATA) -- Datazione del reperto -->
<!ELEMENT disegno - - (#PCDATA) -- Disegno del reperto -->
<!ATTLIST disegno
                         id.nr ID  #IMPLIED
                          -- >
<!ELEMENT foto - - (#PCDATA) -- Fotografia del reperto -->
<!ATTLIST foto
                         id.nr ID  #IMPLIED
                          -- >
<!ELEMENT diapositiva - - (#PCDATA) -- Diapositiva del reperto -->
<!ATTLIST diapositiva
                         id.nr ID  #IMPLIED
                          -- >
<!ELEMENT bibliografia - - (citazione.bibl+) -- Elenco di riferimenti bibliografici -->
<!ELEMENT citazione.bibl - - (#PCDATA) -- Singolo riferimento bibliografico -->
<!ATTLIST citazione.bibl
                         anno NUMBER  #IMPLIED
                          -- >
<!ELEMENT responsabile - - (#PCDATA) -- Autore della scheda -->
<!ELEMENT collocazione - - (#PCDATA) -- Collocazione del reperto -->
<!ELEMENT restauri - - (#PCDATA) -- Eventuali restauri del reperto -->
```

Fig. 3 – Proposta di DTD per l'elemento radice "reperto.ceramico".

tranno ricalcare i meccanismi previsti dal sistema relazionale di gestione dei dati. Consideriamo a titolo esemplificativo un reperto ceramico. Se nel testo dei diari troviamo semplicemente un elenco di reperti rinvenuti in una determinata <giornata> di scavo, la catalogazione dei reperti ci consente di compilare con precisione la scheda descrittiva in formato SGML articolandola come previsto nella DTD per il <reperto.ceramico>[6] (Fig. 3).

Le schede in formato SGML potranno naturalmente essere numerose, tante quanti sono i reperti per i quali disponiamo di dati completi.

Per quanto concerne il rapporto tra la DTD dei diari e le nuove DTD di descrizione dei reperti, in effetti si tratta dell'elaborazione di nuove ramificazioni descrittive per elementi presenti nella DTD dei diari ma definiti come "solo testo". Gli elementi in questione sono quelli contenuti nella <descrizione> della giornata di scavo e utilizzati per codificare i singoli reperti elencati (Reperto_architettonico, Reperto_ceramico, Reperto_metallico, Reperto_osseo). Ogni occorrenza di questi elementi potrà costituire un link verso un documento SGML che descrive in modo approfondito il singolo reperto. Tra la DTD "madre" disegnata per i diari di scavo e le DTD "figlie" disegnate per la codifica delle schede di reperto si viene a configurare una relazione uno-a-molti: il documento principale costituito dal diario di una giornata potrà infatti essere collegato a molte schede di materiali. L'utente potrà navigare nell'archivio testuale leggendo i documenti e seguendo i link interni; oppure potrà interrogare l'archivio costruendo documenti virtuali che combinano informazioni sullo scavo e sui reperti rispondenti ai criteri di ricerca scelti.

Al momento le DTD per la descrizione dei materiali sono in una fase prototipale ma contiamo di poter sperimentare il funzionamento dell'ipertesto e presentarlo in tempi piuttosto brevi. Sarà comunque necessario un periodo di test per verificare sia la struttura delle singole DTD sia l'efficacia dei link che costituiscono i percorsi di navigazione all'interno dell'archivio.

Ilaria Bonincontro
Istituto Storico Italiano per il Medioevo

BIBLIOGRAFIA

Adamo G. 1996, *Edizione e analisi informatica di testi: standard internazionali per la codifica dei dati testuali,* «Archeologia e Calcolatori», 7, 721-734.

[6] La DTD presenta una struttura piuttosto lineare. L'elemento radice è il "reperto.ceramico"; esso è costituito da una serie di elementi, tutti facoltativi come indica il simbolo "?"; alcuni elementi possono essere ripetuti più volte ("*"). All'elemento radice sono stati assegnati due attributi che ne consentono l'identificazione univoca: sono il numero di inventario assegnato al momento della schedatura e un numero univoco da assegnare nella fase di informatizzazione. L'attributo identificativo è stato assegnato anche agli elementi "disegno", "foto", "diapositiva" per identificare in modo univoco ciascuno di questi oggetti.

BONINCONTRO I. 1997, *Archiviazione di dati testuali nel settore archeologico*, «Archeologia e Calcolatori», 8, 139-149.

GOLDFARB CH. 1990, *The SGML Handbook*, Oxford, Clarendon Press.

MOSCATI P. c.s., *Dal dato al modello: l'approccio informatico alla ricerca archeologica sul campo*, in T. ORLANDI (ed.), *I modelli nella ricerca archeologica: il ruolo dell'informatica*, *Atti del Convegno (Roma 2000)*, Accademia Nazionale dei Lincei, in corso di stampa.

MOSCATI P., MARIOTTI S., LIMATA D. 1999, *Il "Progetto Caere": un esempio di informatizzazione dei diari di scavo*, «Archeologia e Calcolatori» 10, 165-188.

ABSTRACT

As part of the Caere Project, the author describes the diverse stages that have characterised the acquisition and encoding in a digital format of the excavation diaries through the application of SGML. This encoding language for electronic documents is focused mostly on describing the internal structure of the data and the information contained in the text. The SGML syntax in some aspects is complex, and inevitably this has been an obstacle to the diffusion of the language.

The transcription and the encoding of the diaries have been completed and a flexible querying system of the SGML documents has been created. The decision to use the Internet in order to distribute information has also implied a study of the viability of converting SGML documents into XML, which in the last few years has been replacing SGML, from which it derives. The completion of the encoding project of the excavation diaries does not represent the final stage; in fact, it is the new phase that it has initiated which is important: further DTDs will be created which will allow the acquisition and encoding of the descriptions of every find. The user will be able to navigate and explore the textual data and, where a more detailed study is required, analyse the objects together with the topographical information.