Archeologia e Calcolatori
8, 1997, 139-149

UNA IPOTESI PER L'ARCHIVIAZIONE DI DATI TESTUALI NEL SETTORE ARCHEOLOGICO. L'IMPIEGO DELLO STANDARD GENERALIZED MARKUP LANGUAGE PER LA CODIFICA DELLE INFORMAZIONI

1. Introduzione

L'archiviazione di dati testuali nel campo dell'archeologia ha da sempre presentato notevoli problemi e resta argomento centrale di riflessione. Negli ultimi anni si sono moltiplicate le iniziative mirate alla costituzione di banche dati che raccogliessero documenti epigrafici, intendendo il termine in senso molto ampio e cioè includendovi anche le legende monetali o altre iscrizioni trovate su diversi tipi di supporto. Le soluzioni adottatte sono state molteplici e hanno avuto esiti più o meno positivi. Certamente soddisfacente appare in genere l'architettura dell'archivio (ormai quasi sempre di tipo relazionale) per la registrazione di tutte quelle informazioni legate al supporto del documento, in quanto reperto di interesse archeologico; per fare qualche esempio, si pensi ai dati relativi al contesto di rinvenimento o al luogo di conservazione, o ancora ai dati relativi alle caratteristiche fisiche del supporto stesso (dimensioni, forma, materiale, etc.)[1].

La trascrizione del testo resta invece un problema (Donati-Giacomini 1994, 143), specialmente quando tale testo deve diventare oggetto di ricerche lessicali o in genere di analisi linguistiche (ricerca di termini o formule significativi; indicizzazione con conseguente normalizzazione dei termini da indicizzare; etc.). Nell'ambito di molti progetti non si è forse riflettuto a sufficienza sul fatto che la memorizzazione dei documenti è un problema da affrontare da un punto di vista generale, e cioè nel campo dell'archeologia allo stesso modo che in altri settori delle discipline umanistiche quali ad esempio la paleografia, la filologia, la linguistica. Le scelte devono essere fatte sulla base degli scopi per i quali un testo viene memorizzato, non in base alla tipologia del documento. Se ad esempio l'intento è semplicemente quello di mostrare il documento associato a determinate chiavi di ricerca disponibili in una banca dati, potrebbe essere sufficiente memorizzare le immagini dei testi. Naturalmente un testo archiviato come immagine è da considerare sullo stesso piano di un disegno, non può cioè essere oggetto di nessun tipo di operazione che non sia la semplice visualizzazione.

[1] Tra le numerose banche dati per testi epigrafici costituite negli ultimi anni ricordiamo a titolo esemplificativo quella del progetto P.E.T.R.A.E. Hispaniarum (Université Michel de Montaigne-Bordeaux III, Centre Pierre Paris, Talence - Cedex); del progetto CAIE (Istituto per l'Archeologia etrusco-italica, CNR Roma - Pandolfini 1996); Epigraph, Cd-rom contenente le epigrafi del *C.I.L.* vol. VI, edito da J. Yuan, Bryan Mawr College.

Per poter operare sul contenuto del documento si deve produrre un testo elettronico opportunamente codificato (GIGLIOZZI 1987), ma ciò complica le procedure di ricerca. Numerose sono state le soluzioni adottate. L'editore del volume VI del *C.I.L.* (cfr. nota 1) ha scelto ad esempio di duplicare tutte le epigrafi in modo da disporre di due versioni diverse del testo:

- una trascritta sciogliendo le legature e trascurando tutte le particolarità epigrafiche o le annotazioni editoriali, utile per eseguire ricerche di parole o frasi;
- un'altra versione con una serie di codici che identificano lettere di forma insolita, legature ed altre caratteristiche epigrafiche significative che in tal modo possono diventare oggetto di ricerche indipendentemente dal contesto.

Di certo questa può essere una soluzione, ma evidentemente richiede molto più spazio disco per memorizzare i dati; la duplicazione delle informazioni è inoltre un'operazione che è preferibile evitare poiché accresce la possibilità di errore.

Un altro esempio di banca dati epigrafica per la quale si è affrontato il problema della codifica del testo è quella del *Cornell Greek Epigrahpy Project*. Lo scopo principale del progetto, come dichiarato dai suoi ideatori, era quello di rendere disponibile una banca dati di testi nella quale si potessero fare ricerche rapide ed efficienti utilizzando un'ampia gamma di programmi su basi hardware differenti. A tale scopo è stato adottato per la codifica dei testi il Beta Code (PERILLI 1995, 51 ss.). Si tratta di una codifica elaborata da D. Packard nell'ambito del progetto *Thesaurus Linguae Grecae* e basata su un codice binario a 7 bit. Al gruppo centrale di segni associati ai 100 codici numerici disponibili, si aggiunge una notevole quantità di combinazioni di tali segni necessarie per rappresentare ogni altro segno o simbolo. Questo tipo di codifica permette l'utilizzo dei dati su qualsiasi piattaforma; tuttavia in assenza dei font opportuni capaci di visualizzare i simboli, l'utente vede direttamente i codici. Il Beta Code presenta comunque dei limiti derivanti soprattutto dal fatto che si tratta di un sistema nato molti anni fa, quando le capacità dell'hardware e del software disponibile costringevano a operare scelte spesso metodologicamente poco corrette ma inevitabili per ragioni di ordine pratico.

Il problema della codifica va analizzato a diversi livelli. A ogni testo elettronico è necessario associare per lo meno una codifica, quella per la memorizzazione dei singoli caratteri, ciascuno dei quali è associato a un codice numerico (HAMMING 1986). Una codifica standard comune a tutte le piattaforme è quella del set ASCII 7 bit, costituito dai caratteri alfanumerici (lettere dell'alfabeto latino, cifre e alcuni segni speciali); i set "estesi" che comprendono i segni speciali degli alfabeti nazionali non sono immediatamente esportabili; lo stesso discorso vale per i set, o font, che vengono apposita-

mente preparati per visualizzare altri segni speciali. Per risolvere il problema si può usare in alternativa un'opportuna combinazione dei segni alfabetici di base per ogni segno speciale; in tal modo si ha la certezza di poter trasferire le informazioni senza perdite; l'insieme di tali combinazioni costituisce a sua volta un sistema di codifica, basato non su combinazioni di bit come nel caso del set ASCII (a ogni carattere alfanumetico è infatti associato un byte costituito da una combinazione di 7 bit), bensì basato su combinazioni di byte.

In un file-documento ai semplici caratteri del testo possono essere associate informazioni o istruzioni di formattazione per una corretta visualizzazione ed eventuale stampa, e tali istruzioni costituiscono un'ulteriore codifica; in genere si tratta di istruzioni formulate in una sorta di linguaggio proprietario, che possono essere interpretate solamente dal programma che ha generato il file. Ogni programma di videoscrittura, ad esempio, produce dei file di questa natura, e per poter passare da una programma a un altro è necessario convertire il formato. Un testo elettronico prodotto da questo tipo di programmi, oltre a non essere immediatamente trasferibile su altri sistemi, non permette l'estrazione automatica delle informazioni contenute a meno di non sfruttare, quando possibile, la formattazione dei caratteri e dei paragrafi come segno distintivo di determinati "oggetti"; ma ciò comporta evidentemente un lungo lavoro di analisi e non dà per altro alcuna garanzia che gli elementi estratti sulla base del formato siano sempre quelli realmente cercati: se per esempio il formato corsivo viene usato per i titoli, chi cerca i titoli potrebbe tentare di estrarre tutte le sequenze di caratteri corsivi; ma se in qualche caso il formato corsivo è stato associato ad altri elementi (per esempio, a termini in lingua diversa da quella del documento), anche questi verranno estratti insieme ai titoli.

Per poter lavorare su un documento come se fosse un archivio, cercando ed estraendo dati presenti al suo interno, è necessario introdurre una codifica relativa non tanto all'aspetto formale del documento, quanto al suo contenuto (Doedens 1994) e tale codifica non deve essere "privata" di un certo software, se si pensa di esportare il documento senza perdita di informazioni. In altre parole è auspicabile il ricorso a degli standard universalmente validi.

La codifica dei contenuti è dunque il passo da compiere perché un documento trasformato in testo elettronico possa essere sottoposto ad analisi semi-automatiche e/o automatiche finalizzate all'individuazione di particolarità sintattiche, di sistemi abbreviativi, etc. Operazione preliminare è la scelta di un sistema e di un modello di codifica adatto al *corpus* di documenti oggetto delle proprie ricerche. Fase successiva è l'analisi dei documenti al fine di individuare quelle informazioni che è importante selezionare e il loro ordinamento all'interno della struttura generale del documento.

Esistono modelli di codifica elaborati appositamente per codificare la struttura dei documenti e le informazioni sul contenuto; non necessariamen-

te tuttavia tali modelli sono adattabili a tutti i tipi di documento. Ecco perché può rivelarsi molto più vantaggioso l'uso di un sistema di regole elaborate per costruire modelli di codifica adatti ai documenti, quale è appunto lo Standard Generalized Markup Language.

2. Come nasce lo Standard Generalized Markup Language: breve storia

Il titolo **Standard Generalized Markup Language**, normalmente sostituito dall'acronimo **SGML**, riassume in modo eccellente le principali caratteristiche di questo linguaggio di codifica per dati testuali. Si tratta appunto di un linguaggio di codifica (Markup Language) riconosciuto dall'International Standard Organization (Standard), basato su un particolare tipo di codifica, che verrà spiegato tra breve, definibile come "generica" (Generalized).

I principî di base dello SGML sono stati pubblicati dall'ISO con il titolo "Information Processing -- Text and Office System -- Standard Generalized Markup Language (SGML)" e il numero 8879-1986.

In quanto elaborato dall'ISO, SGML è un sistema "non proprietario", soggetto solamente al controllo dell'ISO, e questo è un primo importante vantaggio di tale linguaggio, che ne garantisce l'applicabilità su ogni piattaforma hardware e software.

All'origine dello SGML è l'intuizione di separare il contenuto di un documento dal suo aspetto formale e di applicare appunto una "codifica generica" che tenga conto solo delle informazioni contenute nel testo, non del modo in cui vengono presentate. Alla fine degli anni Sessanta venne esposta quest'idea nel corso di un convegno organizzato presso il Canadian Government Printing Office; la prima applicazione fu il tentativo di elaborare una serie di codici utilizzabili per determinati tipi di documenti, con struttura simile.

Il gruppo di ricerca organizzato presso l'IBM nel 1969, guidato da C. Goldfarb, pensò invece di abbandonare l'idea di un elenco di codici universalmente validi e di inventare una sintassi per la descrizione di tipologie di documenti, tipologie identificabili in base alla struttura interna del testo. In pratica un metodo con il quale si potesse codificare qualsiasi documento, individuando di volta in volta il set di codici adatto alla particolare struttura in esame. Questa prima fase di studio portò alla nascita di un sistema di codifica denominato GML (Generalized Markup Language) dal quale fu poi ricavato lo standard.

3. Cos'è SGML: concetti di base

SGML serve a creare schemi descrittivi adatti al documento da codificare. Non è dunque un sistema di codifica in quanto non fornisce i codici da inserire nel testo; fornisce invece delle regole per elaborare un sistema di codici adatto a determinati tipi di documenti. Queste regole costituiscono la

sintassi SGML; il dizionario definisce la sintassi di un linguaggio come «la parte della grammatica che contiene le regole di combinazione degli elementi lessicali e significativi, e quindi di formazione delle fras». La sintassi SGML non contiene infatti modelli descrittivi preordinati, ma le regole per costruire i modelli.

Questo è un punto fondamentale nella valutazione del linguaggio di codifica; la possibilità di scegliere il modello descrittivo adatto al documento in esame e alle operazioni che si intendono compiere su tale documento rende a mio parere questo linguaggio particolarmente idoneo al campo delle scienze umane, nonostante sia nato nel settore dell'editoria e in esso trovi frequentemente applicazione.

La libera scelta del modello di codifica assume un'importanza decisiva in un ambito in cui i documenti che sono oggetto di analisi rispondono necessariamente a schemi di volta in volta diversi e imprevedibili.

4. Sommario delle regole fondamentali

Nella sintassi SGML intervengono alcuni concetti fondamentali che vanno compresi in modo corretto per poter utilizzare al meglio il linguaggio. Un primo punto è il "documento" che rappresenta l'oggetto primario su cui applicare il linguaggio di codifica; rispetto al normale significato del termine, il "documento SGML" è qualcosa di più complesso. Esso va inteso come un archivio di "oggetti" ordinati per lo più in modo gerarchico. Questi oggetti, o "elementi", vengono identificati con l'inserimento nel testo di un codice di apertura (*open-tag*) e di uno di chiusura (*end-tag*). Dal momento che non esiste un modello di documento prestabilito, è necessario definire per ogni documento l'insieme degli elementi presenti e la gerarchia in base alla quale vanno ordinati.

Il documento SGML è dunque costituito da una descrizione o definizione della propria struttura, elaborata in base alle regole della sintassi, chiamata **Document Type Definition**, e da un testo codificato. Naturalmente la stessa DTD può essere associata a più documenti di struttura simile.

La Document Type Definition o DTD deve essere elaborata secondo regole precise affinché rispetti lo standard e sia comprensibile a qualsiasi software SGML. La DTD contiene infatti la descrizione della struttura interna del documento e serve ai programmi per verificare l'esatto posizionamento dei codici all'interno del testo. Elaborare una DTD può essere un compito estremamente difficile, che non devono necessariamente eseguire tutti coloro che codificano documenti con SGML; nelle organizzazioni di tipo commerciale o negli uffici vengono normalmente applicate ai documenti delle DTD preparate da tecnici esperti, in modo che la documentazione sia uniforme quanto a organizzazione interna del contenuto. Quando però si applica il

linguaggio nella codifica di opere letterarie o di fonti storiche o ancora di testi epigrafici, è evidente che non si può affidare l'incarico di elaborare il modello ad un informatico; ciò presupporrebbe da parte di quest'ultimo una conoscenza di tutte quelle caratteristiche significative in un'analisi testuale scientifica. L'elaborazione del modello diventa quindi compito del ricercatore che studia i documenti.

Torniamo per un momento al concetto di documento SGML. Lo si è definito dunque come un archivio di "oggetti", ma quali sono questi oggetti? In generale gli oggetti presenti in un documento sono di due tipi: "elementi" ed "entità".

Un "elemento" è una porzione di testo che contiene determinate informazioni e che si ritiene utile poter identificare automaticamente; ad esempio il capitolo, la nota, etc. Un elemento risponde a determinate caratteristiche dal punto di vista del contenuto; tutte le porzioni di testo di contenuto affine costituiscono un'occorrenza di un particolare elemento, in altre parole ogni capitolo di un libro è un'occorrenza dell'elemento "capitolo". Un elemento può essere un oggetto complesso, cioè contenere al suo interno altri elementi; il capitolo contiene ad esempio il paragrafo. Il livello basso di questa gerarchia di elementi non è fisso, viene al contrario stabilito da chi elabora la DTD, il quale a seconda delle proprie esigenze stabilirà quali sono gli elementi non ulteriormente divisibili. A un elemento possono essere associati degli "attributi" per definire particolari caratteristiche di ogni singola occorrenza; ad esempio, in genere la nota ha un attributo che definisce il numero di nota, importante identificativo da usare per costruire un richiamo alla stessa nota all'interno del testo. Non è corretto usare gli attributi per associare agli elementi istruzioni per la formattazione; tali istruzioni sarebbero infatti necessariamente scritte in un linguaggio proprietario, comprensibile a un solo programma e ciò va chiaramente contro la filosofia dello standard, finalizzato alla produzione di documenti codificati che siano indipendenti dall'hardware e dal software.

Le "entità" sono stringhe di caratteri delimitate dai due segni particolari (& e ;) alle quali possono essere associati contenuti di vario genere, dalle immagini a segni alfabetici a stringhe di caratteri. Il caso più semplice è quello dell'acronimo: l'acronimo, per esempio &CNR;, costituisce l'entità alla quale viene associato lo scioglimento dell'acronimo stesso, Consiglio Nazionale delle Ricerche. Questo sistema permette di introdurre brevi stringhe all'interno del testo e di sostituirle con altro testo quando necessario. L'entità SGML potrebbe ricordare il concetto di variabile nei linguaggi di programmazione, ma a differenza della variabile, alla quale viene assegnato un contenuto di volta in volta diverso durante l'esecuzione del programma, all'entità viene assegnato un unico valore fisso all'interno della DTD. Esistono insiemi di entità standard pubblicati dall'ISO usate per rappresentare caratteri grafici; per esempio il cosiddetto set Latin-1 comprende tutti i segni alfabetici con

diacritici associati nel set ASCII esteso ai codici numerici maggiori di 127.

5. Le ragioni che hanno portato alla nascita di SGML

Si potrebbero presentare diversi argomenti per spiegare perché è stata prodotta una "codifica generica", ma quello a mio parere fondamentale è la necessità di facilitare lo scambio di documenti.

I programmi di word processing producono di norma file con formati proprietari; finché i dati vengono utilizzati e manipolati all'interno del sistema che li ha generati, ciò non rappresenta un problema. Un testo viene dunque archiviato in un file come sequenza di caratteri e di codici (propri del programma), che determinano l'aspetto di visualizzazione e stampa del testo stesso. Cosa succede però a questo documento se il file viene trasportato su un sistema differente? Sebbene le versioni più recenti dei programmi prevedano la possibilità di importare file nei formati più diversi, non è mai garantita al 100% la precisione del trasferimento e ciò si traduce in una perdita costante di informazioni. Ora che le comunicazioni tra computer sono state incrementante dall'uso delle reti, lo scambio di dati rappresenta una necessità costante. Di qui l'importanza di una codifica che sia indipendente dai programmi e pertanto universalmente comprensibile.

6. Vantaggi di SGML

Numerosi sono i vantaggi che derivano dall'utilizzo della codifica generica. In primo luogo l'aspetto esteriore di un documento può essere continuamente modificato; è sufficiente infatti associare nuove istruzioni di impaginazione ai codici presenti nel documento per ottenere diversi esiti tipografici. Questo vuol dire che non è necessario intervenire sui testi codificati. Quando si elabora un testo con un programma di editoria elettronica tradizionale, oltre a digitare il testo vero e proprio, si decide contestualmente l'aspetto del documento: in altre parole si stabilisce dove usare il sottolineato, il corsivo o il grassetto, se i paragrafi devono avere un rientro in prima linea, se le note vanno stampate in fondo alla pagina, e così via. Nel caso in cui l'autore si accorga ad esempio che le citazioni si presentano meglio in corsivo anziché in sottolineato, dovrà intervenire direttamente nel file e cambiare i codici di formattazione-carattere da sottolineato a corsivo in tutti i casi in cui una citazione compare nel testo.

Ci si rende facilmente conto di quale vantaggio presenti il sistema della codifica generica in casi di questo tipo; se l'autore del documento assegna come codice alle citazioni la stringa "cit", potrà in un secondo tempo associare tale stringa al formato carattere sottolineato o corsivo secondo i casi, senza più aprire il file documento, con un evidente risparmio di tempo. Un altro vantaggio di SGML deriva dal fatto che questo linguaggio è del tutto indi-

pendente dal software e dall'hardware, garantisce anzi la trasferibilità dei file su qualsiasi sistema. Si tratta infatti di file ASCII, basati su una tabella internazionale di rappresentazione dei caratteri, quella ASCII a 7 bit, che qualsiasi programma è in grado di importare e leggere indipendentemente dall'ambiente in cui lavora.

Per quanto riguarda la produzione di nuovi documenti, la scelta di codificarli secondo le regole dello standard comporta la necessità di definirne a priori la struttura interna; questa progettazione, che a prima vista potrebbe sembrare un aggravio di lavoro, si traduce invece in un vantaggio. L'organizzazione del testo infatti deve forzatamente seguire delle regole, e i dati contenuti nel documento finale saranno completi e logicamente ordinati. Condizione essenziale è di certo una corretta definizione della struttura e della Documenti Type Definition che la descrive. D'altro canto la presenza di una codifica strutturata permette di riconoscere immediatamente l'organizzazione del testo e agevola il recupero di informazioni da parte di chi legge.

L'aspetto più innovativo è tuttavia la possibilità di recuperare automaticamente le informazioni, come se il documento si trasformasse in un archivio. I codici identificano unità significative nel testo e tramite questi stessi codici un programma può automaticamente estrarre tutte quelle porzioni di testo che appartengono a una determinata categoria. Ad esempio, si potrebbero identificare tutte le citazioni bibliografiche di un articolo ed estrarle per costituire la bibliografia del volume che lo contiene. In casi più complessi la codifica generica può rivelarsi essenziale. Si consideri un testo multilingue; la gestione può avvenire in modo intelligente assegnando un attributo "lingua" ai codici, in modo che il programma sia in grado di distinguere gli oggetti testuali in base alla lingua in cui sono scritti.

Un vantaggio derivante dalla particolare natura non commerciale di questo standard sta nel fatto che esso permette una comunicazione veloce ed economica tra persone che partecipano ad attività concomitanti. Nell'ambito di progetti di ricerca l'abbattimento dei costi materiali (sia in termini di tempo che di denaro) legati all'acquisto di nuovi programmi o alla conversione dei file in formati compatibili con i programmi in uso, solo per fare alcuni esempi, non è di secondaria importanza.

Questi sono soltanto alcuni dei vantaggi immediati che derivano dalla scelta di utilizzare il linguaggio standard di codifica generica dei testi, e che sono indubbiamente fondamentali nel campo della ricerca. In particolare vanno valutati due fattori che conviene ribadire per la loro importanza, forse non sempre e non da tutti riconosciuta: la totale **indipendenza** dall'hardware e dalle applicazioni software, che si traduce nella possibilità di trasferire i file anche tra macchine completamente diverse; l'enorme **flessibilità** di questa sintassi, che permette di elaborare di volta in volta un modello di codifica adatto alla particolare struttura del testo in esame.

```
<!ENTITY % ISOlat1 PUBLIC "ISO 8879-1986//ENTITIES Added Latin 1//EN" >
%ISOlat1;

<!ENTITY % ISOgrk1 PUBLIC "ISO 8879-1986//ENTITIES Greek Letters//EN" >
%ISOlat2;

<!ENTITY % global    "
                id              ID              #IMPLIED
                n               CDATA           #IMPLIED ">

<!--ELEMENT          MIN     CONTENT -->

<!ELEMENT fontes     - -     (liber+) >
<!ELEMENT liber      - -     (title, editor?, sez1+) +(gr) >
<!ELEMENT sez1       - -     (title?, ((scheda+, sez4*) | sez2+)) >
<!ELEMENT sez2       - -     (title?, ((scheda+, sez4*) | sez3+)) >
<!ELEMENT sez3       - -     (title?, ((scheda+, sez4*) | sez4+)) >
<!ELEMENT sez4       - -     (title?, scheda+) >
<!ELEMENT title      - O     (#PCDATA) >
<!ELEMENT editor     - -     (#PCDATA) >

<!-- gr delimita il testo greco -->
<!ELEMENT gr                 - -     (#PCDATA) >

<!ELEMENT scheda     - -     (locus, testo, nota?)       +(fnref | index) >
<!ELEMENT locus      - -     (#PCDATA, cf?) >
<!ELEMENT cf                 - -     (locus, testo) >
<!ELEMENT testo      - -     (#PCDATA) >
<!ELEMENT nota       - -     (#PCDATA)  -(fnref)>
<!ELEMENT fnref      - O     EMPTY >

<!--index segna il punto dove punta l'indice. Il lemma va inserito -->
<!--come attributo dell'elemento -->
<!ELEMENT index      - O     EMPTY >

<!ATTLIST    liber           %global; >
<!ATTLIST    sez1            %global; >
<!ATTLIST    sez2            %global; >
<!ATTLIST    sez3            %global; >
<!ATTLIST    sez4            %global; >
<!ATTLIST    scheda          %global; >
<!ATTLIST    testo           lang            (latin,greek)   latin >
<!ATTLIST    nota            id              ID              #REQUIRED >
<!ATTLIST    fnref           target          CDATA           #REQUIRED >
<!ATTLIST    index           level           CDATA           #REQUIRED
                             index           CDATA           #IMPLIED >
```

Tab. 1

7. Una DTD SGML elaborata per codificare fonti epigrafiche

Presso l'Istituto di Topografia Antica dell'Università di Roma "La Sapienza" è in corso di realizzazione la riedizione e il completamento dei volumi dei *Fontes ad topographiam veteris urbis Romae pertinentes*, editi da G. Lugli; per la codifica dei dati già pubblicati si è scelto lo SGML ed è stata elaborata una DTD che descrive la struttura dei volumi. In questa fase iniziale si sta procedendo ad acquisire per mezzo di uno scanner tutti i testi del primo volume. Per i testi in latino viene utilizzato Omni Page PRO che ha dato buoni risultati sui testi campione; per i testi in greco è stato necessario ricorrere a un altro tipo di software, Optopus della Makrolog, un ICR in grado di effettuare il riconoscimento di segni diversi da quelli dell'alfabeto latino, dopo uno specifico addestramento.

Il testo memorizzato e corretto deve essere codificato sulla base della DTD; quella a Tab. 1 è ancora un prototipo che viene provato sui primi dati disponibili. Nella scelta dei nomi degli elementi si è cercato di rispettare le convenzioni della pubblicazione, anche per facilitare il lavoro di inserimento dei tag. Si è preso in esame il singolo volume dei *Fontes* che rappresenta nel modello l'elemento di primo livello; esso è costituito da capitoli (liber), ciascuno dei quali è diviso in sezioni e sottosezioni (sez). Ogni sezione ha un titolo e contiene una sottosezione o direttamente le fonti (scheda). Le singole fonti sono costituite da un riferimento bibliografico (locus) che identifica la provenienza del testo, dalla trascrizione del testo e da eventuali note. Per la redazione degli indici si è inserito un'elemento (index) che può stare in qualsiasi posizione all'interno della struttura e che come attributo ha la forma normalizzata del termine da indicizzare; in questo modo non si devono operare cambiamenti nel testo mentre sulla base dell'attributo si possono eseguire ricerche.

Le fonti così codificate potranno essere stampate nuovamente e soprattutto diventare oggetto di ricerche che faciliteranno l'opera di riedizione e di aggiornamento dell'edito.

Ilaria Bonincontro
CISADU
Università di Roma "La Sapienza"

BIBLIOGRAFIA

Bryan M. 1988, *SGML. An Author's Guide to the Standard Generalized Markup Language*, Reading (Mass.), Addison Wesley.

Doedens, Chr.-J. 1994, *Text Databases. One Database Model and Several Retrieval Language*, Amsterdam, Rodopi.

Donati Giacomini P. 1994, *La strutturazione dei dati epigrafici*, «Archeologia e Calcolatori», 5, 141-146.

Gigliozzi G. (ed.) 1987, *Studi di codifica e trattamento dei testi*, Informatica e Discipline Umanistiche, 1, Roma, Bulzoni.

GOLDFARB C. F., RUBINSKY Y. 1990, *The SGML Handbook*, Oxford, Clarendon Press.

HAMMING R.W. 1986, *Coding and Information Theory*, Englewood Cliffs, Prentice-Hall, II ed.

International Standard ISO 8879-1986. Information Processing, Text and Office Systems, Standard Generalized Markup Language (SGML) = Traitement de l'information, systèmes bureautiques, langage standard généralisé de balisage (SGML), I ed. International Organization for Standardization (ISO), Genf, Schweiz, 15. Oktober 1986.

PANDOLFINI M. 1996, *Il progetto CAIE*, in P. MOSCATI (ed.), *III Convegno Internazionale di Archeologia e Informatica (Roma, 22-25 novembre 1995)*, «Archeologia e Calcolatori», 7, 795-801.

PERILLI L. 1995, *Filologia computazionale*, Roma, Accademia Nazionale dei Lincei.

RIEGER W. 1995, *SGML für die Praxis*, Berlin, Springer.

VAN HERWIJNEN E. 1994, *Practical SGML*, Norwell, Kluwer Academic Publisher Group, II ed.

ABSTRACT

Main topic of this article is the problem of textual data filing in archaeological field in case they have to be analyzed and processed automatically. This is actually a problem that involves strictly another topic, namely the text encoding. An electronic text may contain more than one markup level. The first one is the character encoding, usually the ASCII - 7 bit set, that offers 127 bit combination to represent letters, digits and few other signs. The ASCII code has the advantage to be a standard valid for every operating system. It is therefore suggested, to encode other signs not included in the ASCII set, the use of a combination of the 127 available characters rather than a combination of 8 bit, i.e. 256 possible combinations, because the number codes greater than 127 are used for different signs on different systems. A document-file may contain another markup that encodes format parameters. To transform a simple document in a "database" where information can be searched and retrieved, the conceptual components must be encoded. The Standard Generalized Markup Language appears to be a good tool to produce files that are software and hardware independent, easy to be managed and first of all ready to be automatically analyzed by a software in order to retrieve pieces of information.