Archeologia e Calcolatori
12, 2001, 71-90

PROGETTO CAERE: UN'APPLICAZIONE INTERNET ATTIVA PER L'*INFORMATION RETRIEVAL* DI DOCUMENTI SGML

1. Introduzione

L'Istituto per l'archeologia etrusco-italica del CNR è da tempo impegnato, in seno al Progetto Finalizzato "Beni Culturali", nella realizzazione di un Sistema informativo archeologico per il territorio dell'antica Caere. Oltre a perseguire obiettivi di carattere ideografico e interpretativo del territorio ceretano, specificatamente per i settori ove si sono susseguite le campagne di scavo condotte negli anni 1983-1989, il lavoro si è fin dall'inizio, ed in modo del tutto programmatico, indirizzato verso la sperimentazione di nuove metodologie e nuovi approcci tecnologici nell'ambito dell'archeologia computazionale.

La documentazione dello scavo di Cerveteri, in particolare quella dei primi anni Ottanta, è piuttosto peculiare. Larga parte dell'area interessata dallo scavo era stata sottoposta a profonde arature moderne, e campo d'azione intensiva di precedenti scavi clandestini (Moscati 2001). La stratigrafia del sito appariva agli archeologi sconvolta e confusa, tanto da indurre i responsabili dello scavo ad aggiungere alle schede formalizzate di documentazione una cospicua collezione di testi esplicativi narrativi, che servissero a descrivere quanto non era riconducibile all'interno di schede standard.

Questi testi, che costituiscono nel loro insieme i giornali di scavo, sono apparsi subito, all'inizio del progetto di informatizzazione, una fonte imprescindibile per l'interpretazione archeologica delle strutture emerse. I diari, redatti sul campo dai responsabili dello scavo, piuttosto che essere «sintetici resoconti quotidiani del progredire del lavoro» (Barker 1977, 188), apparivano come estesi documenti descrittivi, all'interno dei quali era contenuta, in forma già parzialmente strutturata, la maggior parte delle informazioni notevoli dello scavo.

Nel realizzare il sistema informatizzato dei dati di scavo si è quindi pensato di associare alla base cartografica numerica, piuttosto che un database organizzato sulle consuete schede catalografiche, il testo informatizzato di questi diari. Ne è derivato un approccio peculiare: la dignità documentaria del giornale di scavo, così profondamente discussa dalle moderne metodologie di documentazione, ha qui recuperato parte del suo antico valore.

Il significato di questo recupero non è certo da intendersi in termini regressivi, ma progressivi, positivi e integrativi. L'integrazione del dato numerico, oggettivo, topografico e sintetico, con i termini della sua interpretazione diacronica, soggettiva, visuale e intuitiva, offre la possibilità di supera-

re certi limiti interpretativi intrinseci all'analisi a posteriori delle informazioni, esponendo assieme ai dati dello scavo anche i pensieri e le ipotesi dell'archeologo che li ha raccolti.

La giornata di scavo, così come registrata nel diario, è un piccolo tassello di un'evoluzione diacronica, che reca al suo interno, oltre ai dati relativi ai reperti, alla topografia, alla grafica e alla documentazione fotografica – evidenziati nel nostro caso in modo univoco da etichette SGML che ne identificano il significato nel testo – anche le vicende del rinvenimento e la storia dello scavo, con le idee dei ricercatori espresse sul campo, le difficoltà, i dubbi, e quelle sfumature descrittive che sono assai difficilmente riattingibili nei modelli formalizzati di organizzazione dei dati, studiati usualmente attraverso metodi d'analisi basati su paradigmi deduttivi.

In un precedente lavoro relativo al Progetto Caere, pubblicato su questa stessa rivista (Moscati, Mariotti, Limata 1999), si sono già mostrate alcune scelte tecnologiche di base del sistema informativo: in particolare l'adozione del sistema di codifica SGML (*Standard Generalized Markup Language*) per i testi dei diari. L'uso di un linguaggio di marcatura comporta, come è noto, la definizione di norme formali relative al contenuto, ma ha il vantaggio di non alterare il testo cui si applica (Bonincontro 1997). L'uso di un sistema di etichettatura testuale permette inoltre di attribuire a identiche parole, rappresentate da identici significanti, un diverso significato locale, determinato dal contesto in cui si trova la parola.

In presenza di un tale tipo di formattazione del testo è del tutto evidente che, ad una semplice ricerca testuale, possa sostituirsi un sistema d'interrogazione più complesso, capace di riattingere, piuttosto che semplici parole, dei particolari significati all'interno di specifici contesti. L'uso di SGML per la codifica testuale permette quindi, in buona sostanza, di costruire un macrotesto, che aggiunga ai dati anche la loro struttura interpretativa. Gli standard di codifica testuale assicurano poi la scambiabilità incondizionata dei documenti, elemento importantissimo e quasi obbligatorio nella prospettiva della loro diffusione (Adamo 1996).

La possibilità di rendere facilmente accessibili, attraverso le tecnologie informatiche, i documenti prodotti dalla ricerca è oggi una delle opportunità più affascinanti per la scienza, ed ogni gruppo di ricerca intenzionato a promuovere verso l'esterno la sua visibilità scientifica dovrebbe considerarne i vantaggi (Guimier-Sorbets 1996).

Su questi vantaggi, del resto, alcune scuole accademiche hanno già largamente investito: quella inglese ad esempio, che produce e promuove importanti iniziative nel settore della diffusione di documenti archeologici in Internet, e quella americana, all'avanguardia nello sfruttamento delle potenzialità della rete per la diffusione di fonti e materiale iconografico. Anche in Italia tuttavia, sempre più spesso, ai sistemi computerizzati per il trattamento

numerico dei dati, gli archeologi informatici affiancano soluzioni multimediali per la diffusione dei risultati. Del resto, sebbene l'indagine archeologica resti per noi l'unico concreto obiettivo, sembra ormai una realtà consolidata e inconfutabile che la multimedialità si proponga quale strumento per una nuova forma di diffusione della cultura e che essa sia una sorta d'imprescindibile linguaggio dei tempi di un futuro assai prossimo (si veda in particolare ORLANDI 1999 e HODDER 1999). Ogni progetto archeo-informatico che miri a produrre applicazioni per il trattamento interattivo dei dati dovrebbe quindi imporsi, nel rispetto dei suoi obiettivi scientifici, l'adozione di tecnologie adatte a sostenere, in ultimo, la diffusione e lo scambio delle informazioni.

La tecnologia che regge le sorti della rete Internet, al centro di un progresso turbinoso e prevedibilmente inarrestabile, non può che suggerire l'adozione dei suoi specifici modelli diffusivi, che contro ogni tendenza "reazionaria" finiranno prevedibilmente con il divenire i modelli di comunicazione della scienza futura (quale portavoce di una corrente critica e revisionista nei confronti delle potenzialità dell'informatica nell'archeologia cfr. HUGGET 2000).

Il Progetto Caere ha richiesto la collaborazione di diverse figure professionali. Della realizzazione di un'applicazione Internet per l'*Information Retrieval* di testi SGML e della sua integrazione su un server web [1] si è occupato il sottoscritto; quanto verrà descritto nelle pagine seguenti è il risultato di questo specifico impegno. Il presente articolo è diviso in due parti: la prima illustra l'applicazione e i concetti su cui essa si basa, la seconda descrive la tecnologia impiegata.

2. L'APPLICAZIONE

2.1 *La codifica SGML dei diari*

La documentazione di base dello scavo della Vigna Parrocchiale di Cerveteri, che è stata utilizzata per realizzare questa applicazione, è costituita da 7 diari, ciascuno relativo ad una campagna di scavo annuale. Ogni diario raccoglie il resoconto di tutte le giornate della campagna di scavo di quell'anno. Tutti i diari sono stati codificati in SGML utilizzando la stessa DTD (*Document Type Definition*): ogni giornata di ciascun diario è quindi codificata in base ad un comune modello formale.

L'elemento più significativo, nella struttura gerarchica della DTD utilizzata, è identificato dalla coppia di tag <GIORNATA><\GIORNATA> che delimita l'inizio e la fine di un blocco d'informazioni redatte nello stesso

[1] Per server web si intende il computer che ospita i file che costituiscono il sito e che risponde alle richieste dei computer client connessi distribuendo le informazioni secondo un preciso protocollo di scambio dati.

giorno (Moscati, Mariotti, Limata 1999, 170-173). Ogni giornata di scavo, nella sua specificità, è legata ad una struttura formale, che è del tutto identica per tutte le giornate di scavo registrate; in base a questa struttura alcuni elementi precedono altri e appaiono sempre in una precisa posizione gerarchica: *i.e.* gli elementi data, giorno, mese, anno, area, settore, risorse…; altri elementi invece rimangono liberi (struttura, reperto, immagine, strato, etc.) e non hanno l'obbligo di risiedere in punti precisi del testo.

Tuttavia, data la relativa omogeneità strutturale, è stato possibile creare un programma d'interrogazione che sapesse considerare occorrenze testuali in relazione ad alcuni elementi gerarchicamente stabili della DTD, considerati come discriminanti per il recupero del testo. Il diario codificato in SGML è divenuto così assimilabile, dal punto di vista concettuale, ad una sorta di database, i cui dati, piuttosto che risiedere nei campi separati di una o più tabelle, sono mantenuti nella forma originale della loro stesura. Il grande vantaggio che deriva da questo approccio è che i diari rimangono pur sempre leggibili come un testo narrativo, inalterato dal momento della sua redazione sul campo.

Per realizzare il sistema d'interrogazione dei diari abbiamo adottato tecnologie ASP (*Active Server Pages*) e VBSCRIPT (*Visual Basic Scripting Edition*), entrambe nate per la rete Internet ma ugualmente utili per realizzare applicazioni client-server per Intranet; attraverso queste tecnologie è stato anche possibile collegare i diari di scavo alla cartografia digitale e, usando specifici collegamenti ipertestuali con parametri associati, visualizzare l'intero sistema informativo attraverso i browser commerciali più diffusi, cioè Explorer e Navigator. È del tutto evidente che questa applicazione, sebbene nata per essere utilizzata prevalentemente in rete locale, è stata progettata per permetterne la diffusione sulla rete Internet.

2.2 *Le strutture d'interrogazione dei diari*

Tutti i diari di scavo, dopo essere stati codificati in formato SGML, sono stati salvati in formato testo (.TXT) e posti in una directory web del server. Sul server è stato inserito anche un database ACCESS, che contiene le strutture d'interrogazione. Tito Orlandi ha recentemente ricordato come sia difficile, nell'uso dei linguaggi di marcatura testuale, rimanere coerenti ad una descrizione dei dati preordinata, e come questa sia obbligatoriamente soggetta, nel corso dello studio, a subire integrazioni o modifiche, che cambiano a volte in modo sostanziale l'interpretazione della documentazione (Orlandi 1999). L'osservazione di Orlandi, sulla scorta dell'esperienza fatta in questo progetto, appare profondamente vera. Posta una generale struttura formale dei documenti, dal punto di vista sostanziale l'interpretazione del testo assume un carattere dinamico, che investe anche il sistema d'interrogazione. La consultazione informatizzata dei diari deve confrontarsi con le disomogeneità formali della terminologia impiegata dai loro redattori, i quali,

nei vari anni di durata dello scavo, possono aver chiamato aree e strutture identiche con nomi diversi, cambiandone la definizione a mano a mano che l'interpretazione dello scavo prendeva forma.

Le sinonimie debbono essere implementate nelle strutture d'interrogazione, poiché le procedure di ricerca testuale debbono permettere di recuperare identità sostanziali a fronte di difformità terminologiche. Per dotare il programma di ricerca testuale di un'ampia flessibilità e adattabilità al testo si è pensato di creare un semplice database relazionale da associare ai documenti SGML, che contenesse in apposite tabelle il valore-significato di certi elementi topografici per i quali la terminologia impiegata nel testo appariva variabile, sia in funzione della cronologia dei diari sia delle varie fasi interpretative.

Consultando questo database, l'applicazione sviluppata provvede a popolare le liste di opzioni di ricerca messe a disposizione dell'utente solo con valori pertinenti al contesto topografico e ai diari di scavo selezionati e a considerare per essi una serie di sinonimi testuali, che vengono aggiunti al *pattern* di ricerca. I dati sono contenuti in più tabelle separate, legate da relazioni reciproche di carattere topografico e cronologico.

Si sono ottenuti in tal modo due risultati:

– La scoperta di una variante terminologica di un elemento topografico[2], dopo essere stata registrata nell'opportuna tabella del database, si riflette sull'applicazione di interrogazione senza che sia necessario modificare il contenuto degli script e il codice HTML su cui essa è basata.

– L'utente può interrogare il testo in modo intelligente: le opzioni di ricerca disponibili sono solo quelle significative in relazione all'area e al contenuto dei diari selezionati.

Questa scelta tecnica ha un ulteriore vantaggio: la semplice sostituzione delle tabelle del database permette di usare l'applicazione su altri documenti sottoposti alla formattazione della stessa DTD.

Seguendo quindi l'indirizzo di Orlandi, la nostra applicazione si è limitata a considerare una struttura "formale" del documento, ché certo deve essere definita, ma il cui contenuto "sostanziale", *i.e.* il possibile contenuto degli elementi, è letto di volta in volta dalle tabelle di un database.

2.3 *Uso e funzionamento dell'applicazione*

L'applicazione, dal punto di vista dell'utente, è una collezione di pagine HTML (*HyperText Markup Language*). Vi si accede digitando nel browser l'indirizzo URL (*Uniform Resource Locator*) del server su cui l'applicazione risiede.

[2] La terminologia adottata nei diari di scavo per la descrizione delle strutture e delle aree è variabile, in quanto legata alle fasi interpretative dello scavo stesso. Per uno stesso significato possono esistere diverse locuzioni.

La prima pagina è di presentazione ed è una semplice pagina statica HTML. Da questa si accede al *front-end* del motore di ricerca (Tav. 1), anch'esso una pagina HTML, divisa in frame. Ogni frame è il target di una pagina attiva. Alcuni frame ospitano delle liste di opzioni per permettere all'utente di impostare i dati per la ricerca; altri ospitano i risultati della ricerca. Il funzionamento è piuttosto semplice: la Fig. 1 mostra un diagramma di flusso della sequenza operativa.

All'apertura della pagina d'interrogazione, l'utente deve inizialmente selezionare da una lista a discesa i diari sui quali intende effettuare le sue ricerche. La selezione viene inviata ad una pagina ".asp" che provvede a leggere sul server i file delle annate selezionate, ad immagazzinare i dati in una variabile dell'oggetto SESSION di ASP e successivamente ad aprire il database delle strutture d'interrogazione. L'applicazione provvede in seguito a popolare le liste di opzioni di ricerca: *struttura, area* e *reperto* messe a disposizione dell'utente con elementi pertinenti ai diari di scavo selezionati. Se l'utente sceglie di restringere la sua ricerca ad un preciso *settore* dello scavo, o/e ad una precisa *area*, le liste di opzioni per la ricerca sono nuovamente aggiornate con valori pertinenti al contesto.

Definiti i diari, e la base topografica di riferimento, l'utente può richiedere l'effettuazione di 5 particolari tipi di ricerca (Tab. 1).

Struttura	Cerca un elemento <STRUTTURA> che contenga il significato selezionato nella lista
Strato	Cerca tutti i riferimenti agli <STRATI> registrati
Reperto	Cerca tutti i reperti del tipo elencato nella lista
Iscrizione	Cerca tutte le iscrizioni rinvenute sui reperti
Testo	Cerca liberamente una parola nei diari (si possono usare *Regular Expressions*)

Tab. 1

Per effettuare le ricerche all'interno del testo SGML viene utilizzato uno script ASP basato su particolari funzioni, dette *Regular Expressions*, che funziona da parser [3]. Questo script è il cuore dell'intera applicazione.

2.4 *Il funzionamento del motore di ricerca: il parser SGML*

Per comprendere il funzionamento del nostro parser si faccia riferimento alla figura presentata in questo stesso volume nell'articolo di I. Boninconтro, che mostra la struttura della DTD SGML utilizzata per la codifica dei diari. Gli elementi SGML all'interno del testo dei diari codificati

[3] Il parser è un programma che provvede a suddividere un testo in blocchi più piccoli e a interpretarli. Ogni browser Internet è basato su un parser HTML.

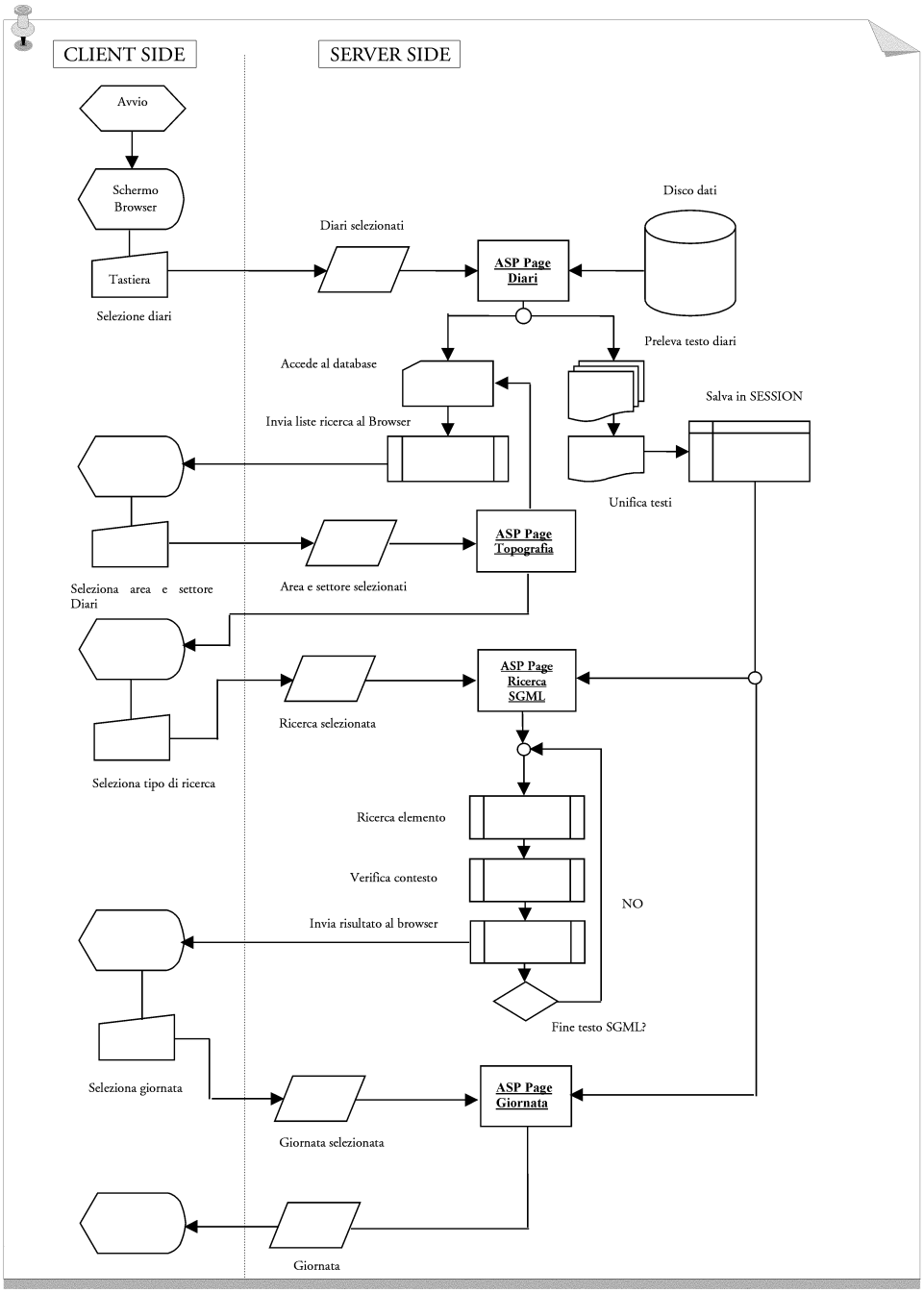


Fig. 1 – Diagramma di flusso semplificato delle sequenze operative e della logica funzionale.

rispettano una precisa sequenza: ogni elemento è contenuto all'interno di altri elementi di rango superiore. Questa è quella che si definisce la "struttura formale" del documento SGML. Posta questa struttura come comune a tutte le giornate di scavo contenute nei diari, un sistema d'interrogazione appositamente studiato ha permesso di recuperare informazioni in base a regole discriminanti multiple.

L'applicazione provvede in prima istanza a ricercare nel testo tutte le occorrenze del tag SGML corrispondente all'elemento cercato (ad esempio <REPMETALLICO>), successivamente ne verifica il contenuto; quando viene identificato un "match", il parser SGML estrae la giornata di scavo dal diario, identificandone i limiti in base ai tag <GIORNATA></GIORNATA>, e verifica se i riferimenti al settore e/o all'area selezionati che precedono il match siano uguali a quelli definiti dall'utente. Quando tutti i criteri sono soddisfatti, il parser provvede ad inviare al browser alcune informazioni riassuntive sul tipo di match incontrato ed i parametri necessari per visualizzare a richiesta per intero il testo della giornata che lo contiene. La procedura viene ripetuta sino alla fine del testo dei diari selezionati (che possono essere più d'uno) e può produrre più risultati. La Tab. 2 mostra i risultati della ricerca, nel diario relativo alla campagna condotta nel 1983, dei reperti metallici di bronzo rinvenuti nell'area ellittica (in sintesi).

n. 1) ***Data:*** 22 giugno 1983, 7 operai --- ***Area:*** [Area ellittica]---***Nome Settore:*** E7IV ρ
Reperto metallico= tracce di bronzo e ferro
Mostra questa giornata per intero
……..
……..
n. 6) ***Data:*** 22 giugno 1983, 7 operai --- ***Area:*** [Area ellittica]---***Nome Settore:*** E7VIII δ
Reperto metallico= anellino di bronzo
Mostra questa giornata per intero

Tab. 2

L'utente può scorrere tutti i risultati ottenuti e decidere di leggere il contenuto di una specifica "giornata" attivando il link ipertestuale **Mostra questa giornata per intero**[4] (Tab. 3). Il contenuto dell'elemento trovato appare evidenziato in giallo.

Se nel testo della giornata ci sono riferimenti ad immagini o ad altra documentazione grafica, questi sono di tipo ipertestuale e, se vengono attivati, permettono di visualizzare le immagini in una finestra.

[4] Il link, se attivato, passa i parametri che identificano univocamente il match ad un'altra pagina ASP, la quale estrae tutta la giornata dai diari e la invia come testo HTML al browser.

Diario della giornata 22 giugno 1983 - 7 operai
settori E7IV μ E7IV ν

[.....]
settore E7VIII δ

[Area ellittica]

Si continua a togliere lo strato superficiale 1 per mettere in luce il piano di tufo, seguendo il taglio nel tufo che corre parallelo all'imposta dei blocchi di fondazione della struttura romana. Materiali frammentari di tegole e lastre tra cui un fr. di cornice con palmetta. Sul piano di tufo a m. 1.45 dal piano di campagna, a m. 3.25 dall'angolo NE del quadrato e a m. 4 dall'angolo della fascia di m. 4, che si era delimitata in precedenza, si rinviene un anellino di bronzo…

Tab. 3

2.5 *Interconnessione con il GIS*

L'interconnessione con la cartografia del GIS è in corso di realizzazione. Si prevede di collegare agli oggetti dei vari layer che costituiscono la base cartografica *hyperlinks* con *querystrings* di parametri da passare alle pagine ".asp". La realizzazione del motore d'interrogazione prescinde, a meno di specifiche e piccole modifiche non rilevanti, dal canale di input dei parametri; per la descrizione più dettagliata del sistema di connessione si rimanda a CECCARELLI, in questo volume.

3. LA TECNOLOGIA IMPIEGATA

La realizzazione di un sistema informativo che integri in un'architettura client-server cartografia digitale e strumenti per l'*Information Retrieval* di documenti testuali non è affatto banale. Inizialmente almeno due considerazioni di carattere tecnologico dovrebbero premettere qualsiasi altra decisione operativa:

– In prima istanza è necessario definire l'insieme delle applicazioni costituenti il sistema informativo: quali funzionalità e quali servizi si offriranno all'utente. Si debbono considerare attentamente i costi del progetto, sia in termini di tempi di realizzazione, sia in termini di risorse economiche.

– In secondo luogo si deve decidere quale sistema operativo si dovrà usare sul server, perché questo è chiaramente un fattore determinante per sapere quali programmi e linguaggi si potranno impiegare per realizzare le applicazioni [5].

[5] Procedendo in senso inverso si può decidere quale server e quale OS acquistare in base alle applicazioni e ai linguaggi di sviluppo che si intendono o si possono produttivamente usare. Questo tipo di approccio è sicuramente più ragionevole, perché tiene conto del know-how e delle risorse che già si possiedono.

3.1 *Le due piattaforme più diffuse per l'hosting di servizi Internet*

Una workstation con sistema operativo Unix può apparire a tutt'oggi la scelta migliore per l'hosting di un web; tuttavia, tra i gestori di servizi Internet, anche il sistema operativo Windows (NT e 2000 Server) è largamente diffuso[6].

Le potenzialità degli ultimi processori Pentium Intel, installati sui più recenti Personal Computer, sono assai significative: la "velocità di clock" degli ultimi modelli presenti sul mercato sfiora ormai i 1500 Mhz. A fronte di investimenti relativamente modesti, si è oggi in grado di avere potenza di calcolo ed efficienza fino a ieri del tutto impensabili.

Anche la disponibilità di software professionale per lo sviluppo di applicazioni sul sistema operativo Windows è oggi ampia; larga parte delle piattaforme GIS che offrono programmi per la pubblicazione on-line di cartografia numerica esistono anche nella versione Windows della Microsoft, ed in questa veste hanno usualmente costi assai ridotti rispetto alle versioni per Unix. Sulla scorta di queste necessarie premesse si possono poi aggiungere delle riflessioni di carattere più specificatamente tecnico.

3.2 *Siti Internet statici e siti interattivi: una profonda differenza*

La realizzazione di un sito Internet statico, costituito da sole pagine di tipo HTML, è di facile realizzazione: i moderni editor HTML o gli ambienti di sviluppo integrati come Frontpage, Netscape Composer, etc. permettono di realizzare e gestire siti con estrema semplicità, anche grazie all'uso di *wizards* e modelli precostituiti. Non altrettanto semplice è la realizzazione di siti con pagine attive: pagine che nascono in base alle scelte operate da un utente.

Per realizzare pagine web interattive esistono oggi diverse soluzioni, il cui grado di complessità è bene conoscere [7]. L'impiego di un server web Unix per l'hosting di pagine attive richiede per la realizzazione dei servizi la stesura di opportuni programmi CGI (*Common Gateway Interface*). Questi programmi sono usualmente scritti in linguaggio PERL. Attraverso gli script CGI un generico browser può comunicare interattivamente con un server: inviare cioè le sue scelte e ricevere in risposta pagine create in base ad elaborazioni di dati residenti sul server del web. Occorre dire, tuttavia, che lo sviluppo dei programmi CGI non è affatto semplice [8].

[6] Un'indagine sulle tecnologie adottate dagli istituti finanziari che offrono servizi di Home Banking rileva che 22 banche delle 33 on-line in Italia lo utilizzano.

[7] Conviene iniziare ad orientarsi sulla Web Technology da http://normandy.sandhills.cc.nc.us/english/shared/html.html.

[8] Un *tutorial* interessante è all'indirizzo http://wdvl.internet.com/Authoring/Scripting/WebWare/

L'impiego di un server Windows NT con estensione IIS 3.0 (o successivo) permette al contrario di sfruttare una tecnologia specifica dei server Microsoft nota come ASP (*Active Server Pages*), sulla quale la realizzazione di servizi Internet attivi ha un grado di complessità di gran lunga inferiore a quello richiesto per la realizzazione di programmi CGI. ASP non è un linguaggio, ma un motore di interpretazione di codice JAVASCRIPT o VBSCRIPT che è stato specificatamente pensato per la realizzazione di pagine attive per il web [9].

3.3 *L'architettura del sistema*

Il nostro Progetto prevedeva il raggiungimento di due obiettivi:

– l'integrazione della cartografia numerica di Caere con i diari di scavo;

– la successiva apertura del sistema verso la rete Internet, poste alcune limitazioni sulla consultazione dei dati necessarie a garantire la sicurezza del sito archeologico.

Si è deciso di utilizzare per realizzare il progetto il sistema operativo Microsoft Windows 2000 Server. Questa scelta è stata fatta per due ragioni: la prima è che questo ci avrebbe permesso di adottare la versione Windows dei pacchetti software GIS di Esri ed Autodesk per la realizzazione della base cartografica; la seconda è che su tale piattaforma avremmo potuto sfruttare ASP per realizzare pagine attive e sarebbe stato assai più semplice sviluppare software client-server.

La fase di sviluppo dell'applicazione di *Information Retrieval* è stata effettuata su un PC in rete del tutto standard, con sistema operativo Windows 98. Sulla macchina di sviluppo è stato installato *Personal WEB Server (PWS)*, un piccolo server per desktop che supporta la tecnologia ASP e permette di testare sul proprio computer il funzionamento delle pagine attive che si realizzano. Il prodotto è presente in qualità di "estensione di installazione" nel CD di Windows 98 e di FrontPage e la sua installazione è del tutto automatica e non difficoltosa.

Tuttavia, poiché l'istallazione di *PWS* espone la macchina di sviluppo sulla rete esterna, è stato anche installato un *personal firewall*. Si è usato *Zone Alarm*, che per utenti individuali e organizzazioni no-profit è un prodotto freeware [10]. Il *firewall* chiude verso l'esterno le porte d'accesso del PC e protegge da visite indesiderate la stazione di sviluppo. Configurando il *firewall* è

[9] Informazioni specifiche sulla tecnologia ASP si trovano nel sito Microsoft http://msdn.microsoft.com, seguendo il percorso MSDN Home > MSDN Library > Web Development> Server Technologies >Active Server Pages>. Più semplicemente in rete nelle centinaia di siti di riviste on-line dedicate alla tecnologia web.

[10] Si può scaricare da http://www.zonelabs.com

possibile accordare l'accesso alla macchina dalla rete solo ad alcuni utenti abilitati.

Al termine della fase di sviluppo l'intero sistema è stato integrato su un PC con sistema operativo Windows 2000 Server, processore da 1 Ghz, 256 Mbyte di Ram, HD da 40 Giga, unità di backup realizzata con un masterizzatore commerciale.

3.4 *La tecnologia ASP*

Nelle pagine seguenti si illustreranno alcune caratteristiche specifiche della tecnologia su cui è basata l'applicazione che presentiamo. Una pagina HTML è, nella sua più semplice accezione, un documento testuale che contiene, oltre alle norme per la sua rappresentazione, i collegamenti attivi ad altri documenti preformati. Per introdurre in una pagina HTML la capacità di eseguire delle operazioni su insiemi di dati è necessario utilizzare lo *scripting*.

Com'è noto, uno script è un blocco di istruzioni inserito in un documento ospite "scritto" in un linguaggio di programmazione ben specifico. Il motore di interpretazione degli script risiede nell'applicazione che gestisce il documento ospite; tutti i browser Internet implementano motori per il linguaggio JAVASCRIPT, derivato del più complesso JAVA; il browser Microsoft Explorer supporta anche lo scripting nel linguaggio VBSCRIPT, perché questo è un linguaggio Microsoft. Purtroppo Netscape per motivi commerciali rifiuta di implementare tecnologia di proprietà Microsoft nei suoi browser e questo crea qualche problema agli sviluppatori, che progettando applicazioni Internet debbono evitare di usare VBSCRIPT per lo scripting sul lato client.

Gli script contenuti in una pagina HTML possono essere tuttavia eseguiti sia dal server sia dal client. A dichiarare dove debbano essere eseguiti gli script è un'istruzione che li precede nel documento ospite. In mancanza di questa dichiarazione il tipo di file che li contiene è un esplicito indicatore per un server Microsoft su come interpretare lo *scripting code* della pagina.

Una pagina sviluppata per un sito Internet ha usualmente l'estensione ".html": *i.e.* "index.html". Se una pagina ".html" contiene script, un server Microsoft non li esegue ma li invia al client, lasciando a lui il compito di eseguirli (*client side scripting*). Se la pagina ha l'estensione ".asp", *i.e* "index.asp", gli script sono invece (a meno di esplicite dichiarazioni) eseguiti dal server (*server side scripting*) [11]. Il diverso trattamento degli script in base al tipo di pagina che li ospita è dovuto al motore ASP del server, che interviene ogni volta che in una transazione è coinvolta una pagina con estensione ".asp".

[11] Numerosi esempi possono essere tratti da siti specifici sulla tecnologia ASP quali http://www.asp101.com/

Realizzare una pagina ASP, dal punto di vista procedurale, è piuttosto semplice: si aggiungono ad una pagina ".html" gli script necessari (JAVASCRIPT o VBSCRIPT), alcune semplici dichiarazioni, e si salva la pagina con l'estensione ".asp". Per definire in ASP uno script su un documento HTML si usa la coppia di tag "<%...%>". All'interno dei tag si scrive il codice VBSCRIPT opportuno: ad esempio istruzioni per operare su database, file, matrici numeriche; istruzioni per interfacciare lo script ai form [12] che appaiono sulle pagine HTML inviate all'utente o più in generale codici per eseguire tutte quelle operazioni che una normale pagina HTML non permette di fare, come cambiare il suo aspetto ed i suoi contenuti in base a certe scelte operate dall'utente. La Fig. 2 mostra un semplice esempio di scripting creato per prelevare il valore di una casella di testo posta su una pagina HTML e restituire all'utente una nuova pagina (ASP) che usa questo valore.

ASP quindi è un ambiente per l'esecuzione server side di script su server web Microsoft che può essere utilizzato per realizzare applicazioni Internet dinamiche. I vantaggi della programmazione server side sono molto importanti nella prospettiva di una diffusione Internet dell'applicazione. Con la tecnologia ASP e il linguaggio VBSCRIPT è possibile realizzare applicazioni che non richiedono particolari capacità al browser dell'utente e sono facili da realizzare. Il linguaggio VBSCRIPT[13] è infatti una versione semplificata del VISUAL BASIC, un linguaggio di programmazione potente ed universalmente diffuso, che è di fatto lo standard di programmazione professionale per l'ambiente Windows. VBSCRIPT è molto facile da apprendere e altrettanto facile da usare.

In sintesi: l'applicazione realizzata su ASP è del tutto indipendente dalle capacità del client di eseguire gli script, perché questi sono eseguiti dal server; questa strategia permette di raggiungere il massimo numero di utenti ed è affidabile. La possibilità di sviluppare programmi in VBSCRIPT semplifica poi notevolmente il lavoro di realizzazione dell'applicazione, poiché chiunque, dopo un minimo investimento di tempo per l'autoapprendimento del linguaggio, è in grado di ottenere rapidi risultati. Nella Fig. 3 mostriamo un piccolo quadro riassuntivo dei componenti di questo progetto. Sul server

[12] I form sono recinti logici contenenti particolari elementi HTML (controlli): pannelli, pulsanti, liste di opzione, caselle di testo. Posti su una pagina web forniscono l'interfaccia grafica per l'interazione con l'utente: i dati ad essi associati devono essere gestiti da procedure specifiche all'interno degli script di una pagina ASP.

[13] Informazioni specifiche sul linguaggio sono nel sito Microsoft http://msdn.microsoft.com seguendo il percorso MSDN Home> MSDN Library> Web Development> Documentation> Scripting> Windows Script Technologies> VBScript>. L'ambiente di sviluppo può essere semplicemente Notepad di Windows o qualsiasi altro strumento per scrivere testi. In FrontPage si possono creare gli script direttamente nella pagina in visualizzazione HTML. Per eseguire e testare gli script (se sono per ASP) si deve pubblicare il sito sul server. Un ambiente di sviluppo più evoluto per lo scripting è Microsoft Script Editor, che è parte di Office. Più complesso ancora Microsoft Interdev, che è il migliore per il web-editing professionale.

Pagina HTML (PAGINA_1.htm)

```
<html>
    <head>
        <title>PAGINA_1.htm</title>
    </head>
<body>
    <FORM METHOD=POST ACTION=PAGINA_2.asp>
    Inserisci il tuo nome:<INPUT NAME="Nome Utente">
    <INPUT TYPE=SUBMIT>
    </FORM>
</body>
</html>
```

Il form dichiarato nel codice HTML dell'esempio presenta al client una casella di testo editabile e un bottone per la Convalida dei dati. Quando l'utente preme il bottone di convalida, il contenuto della casella di testo è inviato con il metodo "POST" a una pagina ASP del server (PAGINA_2.asp).

Pagina ASP (PAGINA_2.asp sul server)

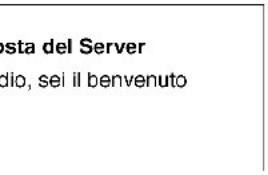
```
<%Option Explicit%>
    <html>
        <head>
            <title>PAGINA_2.asp</title>
        </head>
    <body>
        <b> Risposta del Server </b>
        <%
            Dim Nome as string
            Nome=REQUEST.Form("Nome_Utente")
            RESPONSE.Write ("Ciao " & Nome & ", sei il benvenuto")
        %>
    </body>
    <html>
```

La pagina ASP, quando viene attivata, provvede a richiedere al server la variabile Nome_Utente presente nella collection "Form" dell'oggetto ASP REQUEST (il dato vi è stato immagazzinato quando l'utente ha premuto il tasto di convalida). Il server costruisce poi la stringa di risposta e con il metodo "Write" dell'oggetto ASP RESPONSE la invia al client.

Pagina ASP (PAGINA_2.asp inviata all'utente)

La pagina che il client riceve in risposta è così costituita:

```
<html>
    <head>
        <title>PAGINA 2.asp</title>
    </head>
<body>
    <b> Risposta del Server </b>
    Ciao Claudio, sei il benvenuto!!
</body>
</html>
```



L'esempio illustra il processo di interazione client-server in una transazione ASP. Sebbene qui si mostri solo una delle innumerevoli modalità di scambio delle informazioni tra client e server, il processo, anche per operazioni ben più complesse, è sempre analogo: gli script sono eseguiti dal server; il resto del codice HTML della pagina ASP viene invece inviato direttamente al browser.

Fig. 2 – Esempio di scripting in VBSCRIPT e logica di funzionamento di ASP.

risiedono il database delle strutture d'interrogazione, i file SGML dei diari e le pagine ASP che eseguono gli script per l'interrogazione. Al client, attraverso l'intervento del motore ASP del server, giunge solo codice HTML statico: tutte le operazioni dinamiche sono svolte dal server.

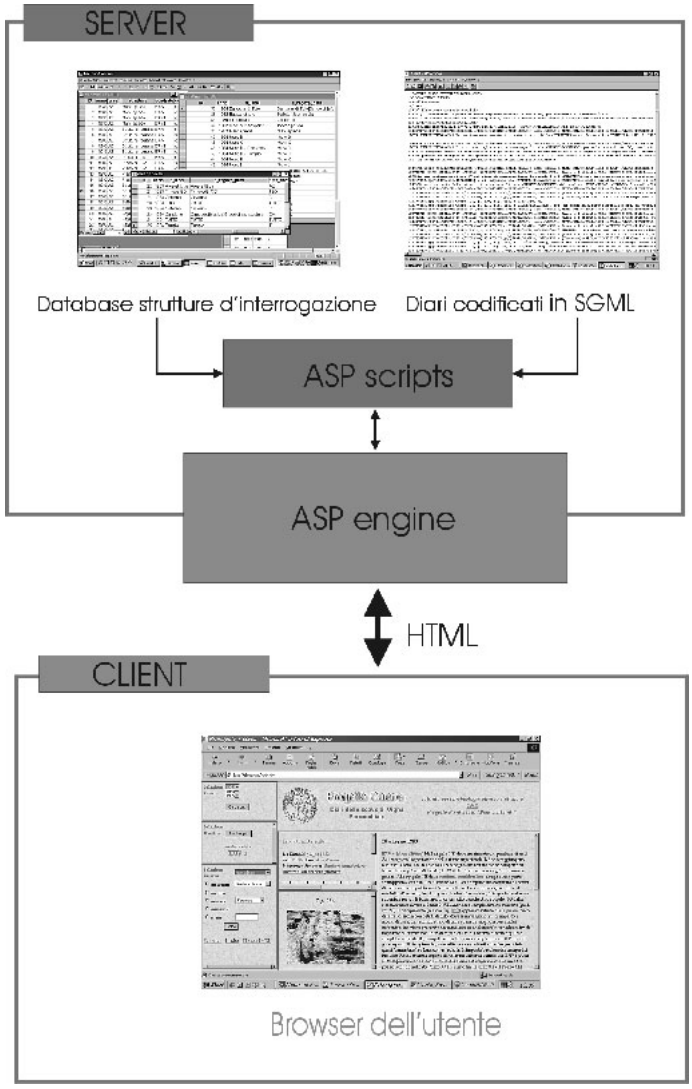


Fig. 3 – Quadro di riferimento dell'architettura di sistema.

3.5 *Interazione con l'utente attraverso l'ambiente ASP*

Per l'interazione con l'utente l'ambiente ASP mette a disposizione 5 oggetti precostituiti (un "oggetto" è un insieme di dati e funzioni che può essere considerato come un'unità) (Tab. 4). Gli oggetti di ASP possono essere facilmente impiegati dagli *scripting languages*.

Oggetto REQUEST	Preleva informazioni da un client
Oggetto RESPONSE	Invia informazioni ad un client
Oggetto SERVER	Controlla l'ambiente d'esecuzione del motore ASP
Oggetto SESSION	Immagazzina dati relativi alla sessione di un client
Oggetto APPLICATION	Permette di condividere dati tra tutti i client connessi

Tab. 4

Prelevando i dati contenuti nell'oggetto REQUEST il server può conoscere le scelte selezionate dall'utente. Attraverso l'oggetto RESPONSE, il server può inviare informazioni ad un client. L'oggetto SERVER viene utilizzato per gestire dallo script le risorse del server, mentre l'oggetto SESSION permette di immagazzinare i dati per un client specifico: dati che rimangono validi fintanto che l'utente è connesso. L'oggetto APPLICATION permette di condividere dati con tutti gli utenti connessi al sito.

Ogni oggetto ASP contiene diverse *collections* di dati ma anche alcuni *methods*, funzioni specifiche per operare su di essi. Per apprendere il modo di gestire gli oggetti ASP è utile far riferimento alla vastissima documentazione e alla gran mole di esempi presentata dai siti web che trattano di *Information Technology*: basta digitare la parola ASP su un qualsiasi motore di ricerca per ottenere gli indirizzi di riviste on-line e di siti accademici e professionali che presentano manuali e articoli tecnici abbastanza chiari da cui trarre procedure e brani di codice da personalizzare. La guida del programmatore VBSCRIPT, che è "obbligatorio" avere sul proprio computer, si può scaricare dal sito Microsoft in forma di file autoexpanding (seguire il percorso MSDN Home > MSDN Downloads > web Development > Windows Script > Microsoft Windows Script 5.6 Documentation). Una sorta di manuale on-line di ASP è ASP ROADMAP: un'utilissima guida di riferimento che dovrebbe essere installata sul proprio computer ma di cui si trovano anche versioni accessibili dalla rete. Per approfondire ci si può collegare al sito ufficiale della Microsoft, nella sezione Scripting.

3.6 *Le* Regular Expressions

L'applicazione di *Information Retrieval* qui presentata utilizza, per le procedure di ricerca testuale, un particolare oggetto software implementato nel linguaggio VBSCRIPT. Quest'oggetto contiene particolari strutture dati e alcune funzioni molto potenti per operare sul testo; il nome è REGEXP e il suo scopo è quello di permettere l'uso delle *Regular Expressions* all'interno del programma .

Cercare del testo generico all'interno di un documento può apparire cosa semplice: tutte le applicazioni di Windows che gestiscono testi dispongono infatti di un comando ("Trova") che permette di cercare stringhe del documento su cui si lavora. In una ricerca di questo tipo si deve scrivere, in

una casella, esattamente la parola che si vuole cercare. Tutti i linguaggi per lo scripting supportano questo tipo di ricerca, che può essere facilmente inserita in un'applicazione di elaborazione di testi: essa tuttavia manca di flessibilità ed è ben poco utile per interrogare documenti SGML, dove si cercano dei concetti piuttosto che delle parole. Tuttavia, oltre a queste funzioni standard, sia VBSCRIPT che JSCRIPT supportano le *Regular Expressions* [14]. Utilizzate per la prima volta sull'editor *qed* di Unix, le *Regular Expressions* sono poi state estesamente impiegate in altri contesti e implementate infine dalla Microsoft nei suoi linguaggi per lo scripting attraveso l'oggetto REGEXP. Con queste strutture è possibile creare pattern di ricerca testuale complessi, congiungenti metacaratteri (segni che indicano funzioni speciali) e caratteri alfanumerici.

Una *Regular Expression* è usualmente costruita con una notazione che ricorda molto da vicino quella algebrica. Un po' complesse da usare, una volta messe a punto svolgono efficientemente qualsiasi tipo di ricerca testuale. Per fare un esempio: si possono cercare in un testo, con un unico comando, tutte le occorrenze della parola "terra" e di quella "tufo" che siano precedute dalla parola "battuto" e siano inserite tra due etichette SGML <STRUTTURA> <\STRUTTURA>.

Il pattern di ricerca di una *Regular Expression* per questo esempio sarebbe:

"<STRUTTURA>.*?battuto.*?terra|tufo.*?<\STRUTTURA>"

e troverebbe indifferentemente nel testo sia "battuto di tufo" sia "battuto irregolare di tufo" sia "battuto di terra scura" e altre simili locuzioni.

Attraverso l'oggetto REGEXP è possibile conoscere la posizione di ogni singolo risultato incontrato nel testo (match) e il suo contenuto specifico. Usando *methods* e *collections* specifiche dell'oggetto è possibile operare delle sostituzioni di parole, memorizzare in array tutti i risultati ottenuti, o la parte di essi che soddisfa alcuni requisiti. L'oggetto REGEXP si è rivelato quindi uno strumento fondamentale per gestire i testi marcati. Il parser SGML di questa applicazione è stato progettato attraverso un suo esteso impiego. Le *Regular Expressions* si sono rivelate infatti strutture fondamentali e insostituibili per la ricerca testuale condizionata.

3.7 *Un metodo per l'applicazione di stile HTML al testo SGML*

HTML è un sottoinsieme di SGML. La DTD di HTML è nota a tutti i programmi che permettono la navigazione sulla rete Internet, poiché essa è

[14] Per apprendere l'uso delle *Regular Expressions* si veda il sito Microsoft http://msdn.microsoft.com e si segua questo percorso nei menu del sito: MSDN Home> MSDN Library> Web Development> Documentation> Scripting> Windows Script Technologies> VBScript> User's Guide> Introduction to Regular Expressions (JScript.NET).

"contenuta" all'interno di queste applicazioni. Un generico browser conosce tuttavia solo e soltanto la DTD dell'HTML e ignora tutte le etichette (*i.e.* <MIOELEMENTO> <\MIOELEMENTO>) che non vi sono elencate. Le etichette SGML definite dall'utente sono quindi ignorate nei browser Internet quali Navigator, Explorer, etc. Per superare questo ed altri limiti intrinseci dell'HTML, il mondo Internet sta lentamente migrando verso il nuovo standard di codifica XML (*Extensible Markup Language*) [15] (BONINCONTRO, in questo volume), ma in attesa di questa rivoluzione tecnologica il problema deve essere risolto con altri mezzi.

Se si vogliono visualizzare specifici elementi di un testo SGML con uno stile particolare negli attuali browser Internet, la soluzione più semplice è aggiungere al testo SGML, prima di inviarlo all'utente, delle etichette HTML opportune.

Si ponga il caso di voler rappresentare alcuni elementi di un testo SGML in particolari colori, o con particolari font*:* tralasciando soluzioni più complesse, la tecnologia ASP e il linguaggio VBSCRIPT permettono di risolvere il problema abbastanza semplicemente. Ci sono due possibilità. La più semplice consiste nell'effettuare con uno script un parsing preventivo sul server del testo SGML, e sostituire le etichette SGML con etichette HTML opportune. Un po' più complesso, ma tuttavia ugualmente praticabile, se non si vuole modificare il testo, è "aggiungere" internamente agli elementi SGML un etichettatura HTML conveniente. Il browser dell'utente continuerà ad ignorare le etichette degli elementi SGML definiti dall'utente, ma interpreterà correttamente gli elementi HTML aggiunti dal parser nella pagina ASP.

3.8 *Font embedding in pagine HTML*

Nella realizzazione di un sito web l'uso di font di caratteri particolari può impedire a molti utenti di visualizzare correttamente i dati contenuti nelle pagine. Un browser che riceve un pagina HTML, se non possiede un particolare font in essa indicato, cerca di sostituirlo con quello che considera il più adatto tra quelli a sua disposizione. In un sito di carattere archeologico il ricorso a font "desueti" è quasi obbligatorio: ad esempio il font "GREEK" non è universalmente diffuso ed il suo impiego potrebbe causare degli errori di visualizzazione. Come risolvere il problema? La soluzione di segnalare all'utente la necessità di scaricare il font richiesto da un apposito link FTP del sito risulta piuttosto sconveniente. La soluzione migliore è inserire i font impiegati all'interno del sito, associandoli alle pagine che ne fanno uso. La tecnica Microsoft si chiama *Open Type Font Embedding* e si basa sui fogli di stile CSS.

[15] Si veda http://www.w3.org/XML. Per un buon manuale cfr. L.A. PHILLIPS, *Usare XML*, Milano 2000, Mondadori Informatica.

Microsoft a questo scopo distribuisce un *tool freeware* specifico, Microsoft WEFT [16], che analizza tutte le pagine di un sito web e genera dei font compressi da includervi. Nel nostro caso abbiamo usato WEFT per utilizzare il font greco impiegato per trascrivere iscrizioni rinvenute sui reperti e per rappresentare correttamente i quadrati della griglia di scavo.

4. Conclusioni

Questo articolo, d'intenzione fondamentalmente tecnologica, ha voluto mostrare una soluzione per l'interrogazione in rete dei diari di scavo codificati in formato SGML. Una soluzione, s'intende, che non pretende d'essere la migliore, ma che ha il pregio di funzionare bene e di essere di facile realizzazione. A fronte della progressiva complicazione delle tecnologie Internet e del moltiplicarsi esponenziale dei possibili approcci al problema della diffusione dei dati in rete, questo lavoro ha l'intento di mostrare gli strumenti per la realizzazione di una soluzione praticabile, semplice, efficiente, che qualsiasi gruppo di archeologi informatici possa considerare interessante per le sue necessità di interrogazione e diffusione di documenti testuali.

Claudio Barchesi
Istituto per l'archeologia etrusco-italica
CNR - Roma

BIBLIOGRAFIA

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

Barker P. 1977, *Tecniche dello scavo archeologico*, Milano, Longanesi (ed it. a cura di B. d'Agostino).

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

Guimier-Sorbets A.-M. 1996, *Le traitement de l'information en archéologie: archivage, publication et diffusion*, «Archeologia e Calcolatori», 7, 985-995.

Hodder I. 1999, *Archaeology and global information systems*, «Internet Archaeology», 6 (http://intarch.ac.uk/journal/issue6/Hodder/toc.html)

Hugget J. 2000, *Computers and archaeological culture change*, in G. Lock, K. Brown (eds.), *On the Theory and Practice of Archaeological Computing*, Oxford, Oxbow Books, 5-22.

Moscati P. 2001, *Scavi archeologici e scavi clandestini : il caso ceretano,* in M.P. Guermandi (ed.), *Rischio Archeologico se lo conosci lo eviti. Atti del Convegno di studi su cartografia archeologica e tutela del territorio (Ferrara 2000)*, Firenze, All'Insegna del Giglio, 361-367.

[16] Si può scaricare da http://www.microsoft.com/typography/web/fonts/default.htm

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

ORLANDI T. 1990, *L'ambiente Unix e le applicazioni umanistiche*, «Archeologia e Calcolatori», 1, 237-251.

ORLANDI T. 1999, *Multimedialità e archeologia*, «Archeologia e Calcolatori», 10, 145-158.

ABSTRACT

In the process of creating an archaeological information system of the excavations in Cerveteri, the decision was made not only to use a more traditional database, but also to develop a recording methodology that connects the text of the excavation diaries, encoded by the application of a mark-up language (SGML), with the cartographic data. In order to query all the excavation diaries, an information retrieval application is required, with the aim to retrieve not only words but also specific meanings and contexts.

In this paper the author describes the creation of an internal software application for providing information retrieval from SGML texts and of its subsequent implementation on a Web server. The paper is divided into two parts: the first describes the application itself and the concepts on which it is based and the second part discusses the technology that has been applied and the results achieved. In order to construct a querying system for the content of the excavation diaries, both ASP and VBSCRIPT technologies have been used, as they are particularly useful for constructing client-server applications for an Intranet. Through applying such technologies, it has been possible to connect the textual sources with the digital cartography through specific hypertext links, allowing the visualisation of the search results in a browser such as Explorer or Netscape Navigator. This application has also been designed to allow data diffusion through the Internet.