HPS: Un sistema di presentazione ipertestuale

N. Aloia, S. Giuliano, G.A. Romano
CNUCE-CNR, Via S. Maria 36, 56126 Pisa
E-mail (N.Aloia,G.A.Romano)@cnuce.cnr.it

CNUCE C96-015

Pisa, 5 luglio 1996

Introduzione

Il lavoro di ricerca e recupero dell'informazione coinvolge sempre più spesso l'uso contemporaneo di più sorgenti informative con diversi contenuti e meccanismi di accesso. Un obbiettivo comune al lavoro attuale sui protocolli di accesso all'informazione è quello di fornire un accesso uniforme a diverse sorgenti di documenti. In questo contesto sono già stati fatti tentativi per ampliare e migliorare l'interazione fra utenti e informazione offrendo un adeguato ambiente informativo.

Tra le proposte di questi ultimi anni, la ormai diffusa affermazione di WWW (con il suo protocollo, standard di fatto, HyperText Transfer Protocol o http) come strumento di accesso all'informazione e lo sviluppo di browser multiprotocollo tipo "Netscape", "Mosaic", "Linkx" etc., conosciuti anche come WWW browser, hanno notevolmente contribuito all'attuale successo di Internet. Questi sistemi hanno consentito di agevolare in maniera significativa l'attività di ricerca e presentazione di documenti, in quanto hanno reso fruibile, in maniera semplice ed intuitiva, la navigazione tra le varie sorgenti e nel contempo hanno reso gradevole la visualizzazione integrata di dati di natura diversa (testo, immagini, voce, etc) mediante un modello ipertestuale. Tutti i WWW browser sono infatti in grado di interpretare il linguaggio HTML (Hypertext Markup Language) che permette di definire documenti ipertestuali.

Non tutti i documenti reperibili sulla rete, per esempio via FTP o via Wais, sono descritti tramite il linguaggio HTML. Tali documenti vengono presentati dai WWW browser con un *look* abbastanza povero e per di più non consentono la navigazione verso ulteriori altri siti, contenenti documenti correlati, se non attraverso la formulazione di una nuova richiesta.

Inoltre, per varie ragioni, difficilmente sormontabili, non esiste uniformità tra le varie sorgenti nella definizione dei documenti con analogo contenuto informativo, sia per quanto riguarda la loro strutturazione che per il linguaggio utilizzato, per cui diventa complessa se non impossibile la loro interpretazione al fine della presentazione; per questi motivi spesso l'utilizzatore rimane disorientato dalla risposta ottenuta in conseguenza di una sua ricerca.

Questo lavoro si propone due obbiettivi principali:

1. Fornire uno strumento, per l'utente finale, che renda il più possibile semplice e gradevole la navigazione in tutte le sorgenti che non supportano un modello ipertestuale.

2. Fornire uno strumento per gli amministratori di server http che consenta di definire in maniera semplice e flessibile, le modalità di acquisizione e di presentazione di documenti di sorgenti differenti.

Proponiamo quindi un sistema, sviluppato in ambiente WWW, per il recupero e la presentazione ipertestuale di documenti distribuiti su Internet attraverso i protocolli standard (WAIS, Gopher, HTTP, FTP....).

Per quanto riguarda il primo obbiettivo il sistema fornisce all'utente:

- un unico meccanismo di accesso a sorgenti distribuite attraverso i protocolli standard forniti su internet rendendo trasparente l'eterogeneità di tali protocolli;

- una presentazione ipertestuale di tutti i documenti recuperati dalle varie sorgenti basata su uno schema di documento definito dall'amministratore del sistema;

- una presentazione uniforme di documenti semanticamente simili. Il concetto di similitudine semantica si basa esclusivamente sulla percezione dell'utilizzaore e viene definito a suo esclusivo uso.

Per quanto riguarda il secondo obiettivo il sistema che viene proposto ha le seguenti caratteristiche:

- può interrogare database differenti;

- fornisce un semplice linguaggio dichiarativo per la definizione di sorgenti di documenti;

- fornisce un semplice linguaggio dichiarativo per la definizione di un modello di presentazione per un certo insieme di documenti. Questa caratteristica permette di vedere un documento non più come un oggetto statico pubblicato in un certo momento ed in base ad un modello definito staticamente, ma come un oggetto dinamico, che ammette un modello di presentazione che soddisfi il bisogno informativo di diversi gruppi di utenti.

Il meccanismo di acquisizione di presentazione dei documenti si basa sui concetti di schema origine, schema canonico e di preferenze di presentazione:

- lo *schema origine* è una descrizione strutturale dei documenti forniti da una certa sorgente. Per ogni sorgente possono essere definiti più schema origine diversi che corrispondono ad una diversa visione che si vuole fornire a diversi gruppi di utenti.

- lo *schema canonico* incorpora le caratteristiche fondamentali di un insieme di schema origine di sorgenti con analogo contenuto informativo. Lo schema canonico permette di uniformare la presentazione di documenti semanticamente simili.

- le *preferenze di presentazione* permettono di migliorare la qualità della presentazione dei documenti. Le preferenze di presentazione determinano, assieme con gli schema sopra descritti, il modello di presentazione che verrà adottato per un certo tipo di documenti.

Il sistema proposto si compone essenzialmente di due parti.

La prima (Multi Archive Access Engine o MAAE) funziona da motore di accesso alle sorgenti, si occupa quindi della ricerca e del recupero dei documenti richiesti dall'utente. Questa parte è strutturata a moduli, ciò facilita la possibilità di inserire nuovi moduli client capaci di recuperare documenti da qualsiasi tipo di sorgente. Attualmente è stato implementato solo un modulo client WAIS in quanto ritenuto il più complesso. Viene, comunque, fornita la specifica delle interfacce da utilizzare per implementare client di tipo diverso.

La seconda parte è costituita da un singolo modulo (Hypertext Presentation System o HPS) che esegue la traduzione dei documenti dal modello di origine nel modello di presentazione opportuno.

Il client WAIS implementato è capace di eseguire tre tipi di operazioni:

- Search: esegue la ricerca di documenti basandosi su una specifica interrogazione. All'utente viene fornita in risposta una lista pesata di "titoli" (document headers) di documenti attinenti all'interrogazione. Un generico titolo è un riferimento ipertestuale all'intero documento.

- Get: recupera il documento di cui viene fornito l'identificatore (DocumentID) e lo passa al modulo di presentazione per essere tradotto. All'utente verrà presentato il documento nel modello di presentazione opportuno.

- Search and Get: esegue la ricerca dei documenti attinenti all'interrogazione proposta e li passa al modulo presentazione per essere tradotti. All'utente verrà fornita in riposta la versione integrale dei documenti trasformati secondo il modello di presentazione opportuno.

Supporremo in seguito che il lettore conosca l'ambiente di sviluppo WWW:
- il sistema di indirizzamento (URL)[1];
- il protocollo HTTP[2];
- il meccanismo CGI;
- il linguaggio HTML[3];

[1][rfc 1738] T. Verners-Lee, L. Masinter, M. McCahill, "Uniform Resource Locators (URL)", dicembre 1994.

[2]M. Handley, J. Crowcroft, "World Wide Web: Beneath the Surf", UCL Press, 1994, ISBN: 1-85728-435-6, URL: `http://www.cs.ucl.ac.uk/staff/join/book/book.html`
D. M. Chandler, B. Kirkner, J.Minatel, "Running a Perfect Web Site", Que, ISBN 0-7897-0210-X.

[3][rfc 1866] T. Berners-Lee, MIT/W3C, D. Connolly, "Hypertext Markup Language -2.0", novembre 1995.

Motivazioni e problemi affrontati

L'affermazione di Internet e la diffusione di sorgenti di documenti su vaste aree dell'informazione sta incidendo sempre più pesantemente nelle attività lavorative e organizzative di ognuno di noi. Per esempio l'Area Scientifica Pisana ha reso disponibili cataloghi di biblioteche trasformando il loro ambiente gestionale in quello di accesso all'informazione distribuita. Attualmente è quindi possibile utilizzare una workstation collegata a Internet per effettuare la ricerca di un certo articolo scientifico anziché spostarsi fisicamente nelle pur vicine biblioteche. Dopo un pò di pratica ad ogni utilizzatore viene spontanea l'abitudine di rivolgere interrogazioni su un certo argomento ad un insieme di sorgenti, che la sua esperienza permette di classificare come sorgenti di documenti semanticamente simili.

La definizione di similitudine deriva essenzialmente dalla percezione soggettiva dell'oggetto ritrovato. Per esempio se si effettua una ricerca di articoli sul tema "User interfaces" sul catalogo delle pubblicazioni dell'Istituto di Elaborazione dell'Informazione (IEI) e sul catalogo delle pubblicazioni dell'Istituto CNUCE, il risultato che viene fornito in risposta è percepito come un insieme di documenti simili (sono tutti articoli su un certo argomento). Ha poca importanza se la struttura dei due sottoinsiemi è diversa (per esempio uno contiene informazioni circa la data di pubblicazione e l'altro no, oppure le informazioni compaiono in un ordine diverso, etc.), per quello che riguarda l'obiettivo della ricerca i due sottoinsiemi di oggetti sono percepiti nella stessa maniera perché hanno lo stesso contenuto informativo. In figura 1 e 2 sono presentati due documenti di esempio ottenuti in risposta all'interrogazione proposta, il documento di figura 1 proviene dal catalogo delle pubblicazioni dell'IEI, l'altro proviene dal catalogo delle pubblicazioni del CNUCE.

Il concetto espresso di documenti semanticamente simili, è un concetto strettamente personale e non può essere generalizzato, in quanto ognuno percepirà questa *somiglianza* sulla base degli obiettivi che si è posto nella formulazionme dell'interrogazione.

```
IEI-CNR, PISA-ITALY, DBN=IEI PUBBLICATIONS, ID=A85-09
AU:      BERTINO, E.
TI:      DESIGN IUSSES IN INTERACTIVE USER INTERFACES
NO:      INTERFACES IN COMPUTING, 3 (1985) 37-53.
PY:      1985.
ND:      A85-09.
```

Figura 1. Un documento proveniente dal catalogo delle pubblicazioni dell'IEI.

```
CNUCE/CNR, Pisa-Italy, Dbn=CNUCE-Internal-Reports, id=c88-041
DT: CNUCE c88-041
TI: User interface: aspetti metodologici e progettuali
AU:      Scopigno R.
PY: 1988
```

Figura 2. Un documento proveniente dal catalogo delle pubblicazioni del CNUCE.

Ammesso quindi che si possano classificare sorgenti di documenti in base ad un modello astratto di somiglianza, sarebbe comodo avere uno strumento che permettesse di inoltrare una richiesta del tipo "trova tutti gli articoli sul tema *User Interface*" ad un certo numero di sorgenti semanticamente simili e presentasse i documenti ottenuti con la stessa veste grafica e uniformando il modello di presentazione (per esempio imponendo lo stesso nome ai paragrafi di entrambi gli insiemi di documenti ed omettendo i paragrafi che non sono comuni alle due sorgenti).

Un'altra difficoltà, nell'interpretazione del risultato di una ricerca di documenti, deriva spesso dalle caratteristiche fisiche degli oggetti ritrovati. In Internet i documenti descritti tramite il linguaggio HTML sono ormai universalmente accettati e graditi in quanto, oltre alla possibilità di fornire una presentazione accattivante, facilitano enormemente la possibilità di navigazione tra le diverse sorgenti. Non tutti i documenti rintracciabili nelle varie sorgenti sono però documenti di tipo HTML, per cui spesso la presentazione che se ne ottiene è povera sia dal punto di vista della resa visiva che, per le possibilità di interazione (non è

possibile seguire eventuali riferimenti del documento se non tramite la formulazione di una nuova richiesta).

Si propone, quindi, un sistema di presentazione ipertestuale che tenga conto dell'eventuale "similitudine" delle sorgenti utilizzate dall'utente finale e sia capace, dove le sorgenti non restituiscano documenti in formato HTML, di tradurre questi ultimi ed eventualmente aggiungere i riferimenti ipertestuali che competono all'insieme delle sorgenti "simili".

Per raggiungere l'obiettivo proposto si devono risolvere tre tipi di problema:

1. deve essere possibile effettuare la traduzione di documenti da un formato ASCII ad un formato HTML;

2. si deve capire fino a che punto sia possibile uniformare la presentazione di documenti che hanno un modello di origine diverso;

3. deve essere fornito un meccanismo di costruzione di riferimenti ipertestuali.

Il problema della traduzione

Il primo problema che deve essere affrontato è quello della traduzione dei documenti da un modello ad un altro. Bisogna innanzitutto distinguere due tipologie di documenti:

- quelli con una struttura esplicita;

- quelli in cui la struttura è implicita.

I documenti strutturati esplicitamente sono quelli per cui esistono dei comandi in linea o comunque delle etichette che permettono di distinguerne i diversi paragrafi. Per questo tipo di documenti è possibile pensare ad una traduzione automatica.

Per i documenti non strutturati la traduzione è invece complessa e richiede l'intervento umano.

Il metodo di traduzione che verrà proposto è applicabile a documenti esplicitamente strutturati attraverso l'uso di etichette che permettono di individuare dei paragrafi. Si può rappresentare l'algoritmo di traduzione di un testo in ipertesto tramite un grafo. Gli archi di tale grafo rappresentano la stringa che si deve riconoscere nel testo per poter procedere da un nodo ad un altro. In ogni nodo il traduttore esegue un passo di conversione da testo a ipertesto.

Si consideri ad esempio un documento bibliografico in cui siano contenuti i paragrafi titolo, autore ed editore, individuati rispettivamente dalle etichette "TI", "AU" e "ED". In figura 3 è mostrato un possibile grafo di traduzione. Sugli archi sono marcate le espressioni da ricercare nei documenti, ogni volta che viene incontrata una espressione l'algoritmo di traduzione procede di un passo. I cicli più piccoli rappresentano la raccolta delle informazioni contenute in ogni paragrafo.[4]

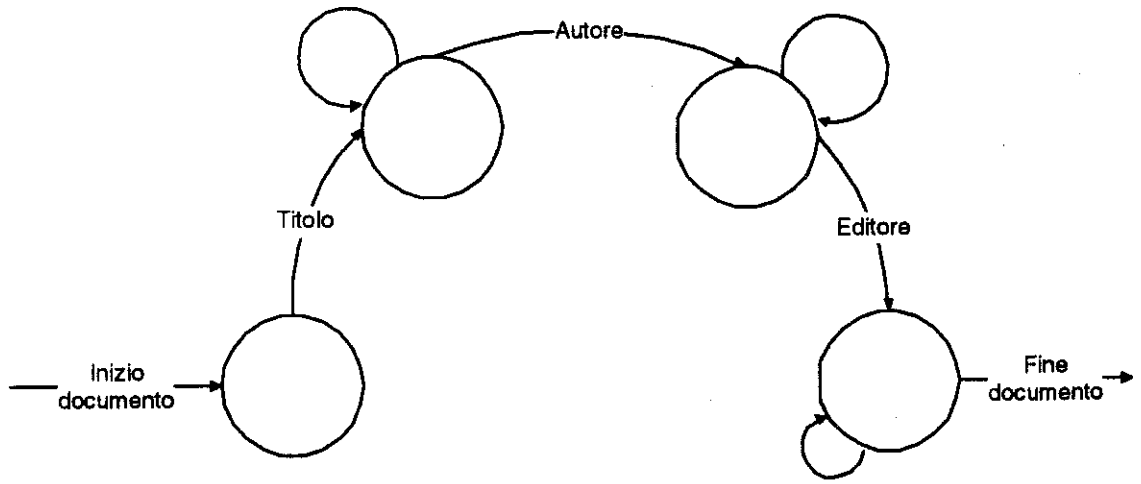

Figura 3. Un possibile grafo di traduzione per documenti bibliografici.

L'architettura del sistema

In figura 4 è illustrata l'architettura del sistema proposto. Il sistema è composto da un insieme di moduli in esecuzione su un computer connesso a Internet ("server del sistema" o "HPS server") che svolge funzioni di gateway fra il protocollo HTTP e

[4] R, Furuta, C. Plaisant, B. Shneiderman, "Automaticlly transforming linear documents into hypertext", *Electronic Publishing: Origination, Dissemination and Design*, 1989, 2, 4, 211-230.

i protocolli usati dai server che possiedono e distribuiscono le sorgenti di documenti. Il Multi Archive Access Sistem (MAAS) formula richieste ai sistemi sorgente. L'Hypertext Presentation Sistem (HPS) interpreta le istanze di documenti risultato della ricerca e genera la descrizione in HTML per la workstation client.

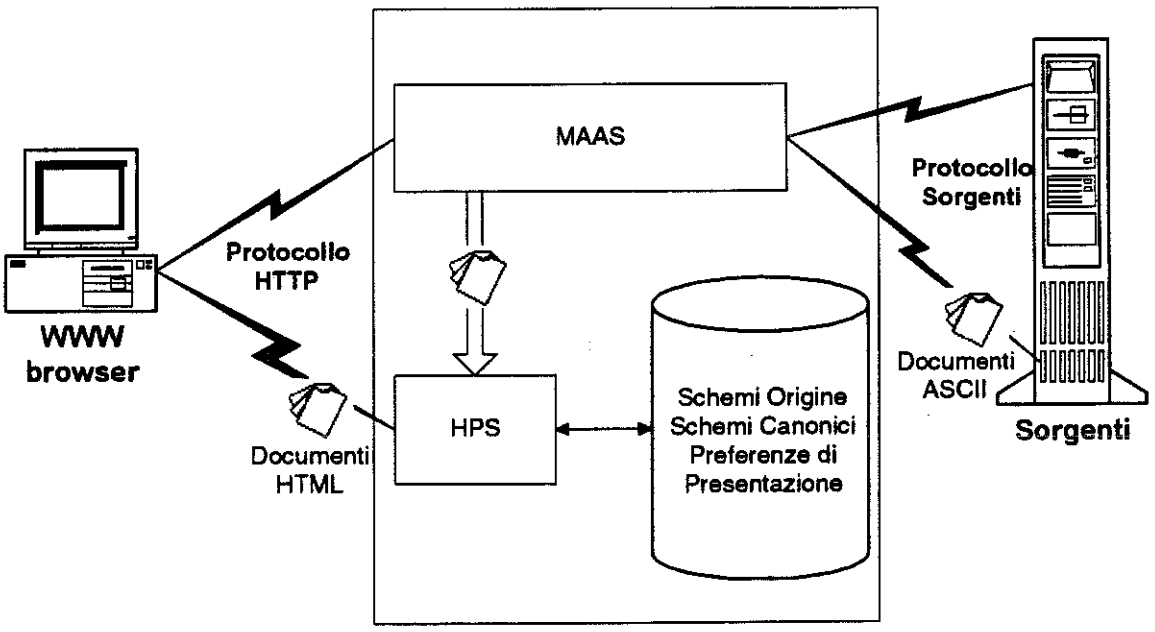

Figura 4. L'architettura del sistema.

Il sottosistema MAAS funziona da server HTTP e da client per i sistemi che possiedono le sorgenti. E' quindi in grado di effettuare delle ricerche utilizzando il protocollo di comunicazione appropriato. Il risultato di tale ricerca è fornito in input al modulo HPS.

Per effettuare la traduzione, il modulo HPS ha bisogno delle seguenti informazioni (figura 5):

- lo schema origine che descrive la struttura dei documenti provenienti da una sorgente. Per ogni sorgente è possibile definire più schemi origine diversi. Ogni

schema origine corrisponde ad una particolare "visione" della sorgente che si vuole fornire ad un determinato gruppo di utenti.

- lo schema canonico che descrive i documenti in maniera indipendente dalle sorgenti di provenienza. Ogni schema canonico viene definito dall'utilizzatore astraendo le proprietà dei vari schema origine, significativi per il suo scopo.

- le eventuali preferenze di presentazione associate allo schema canonico.

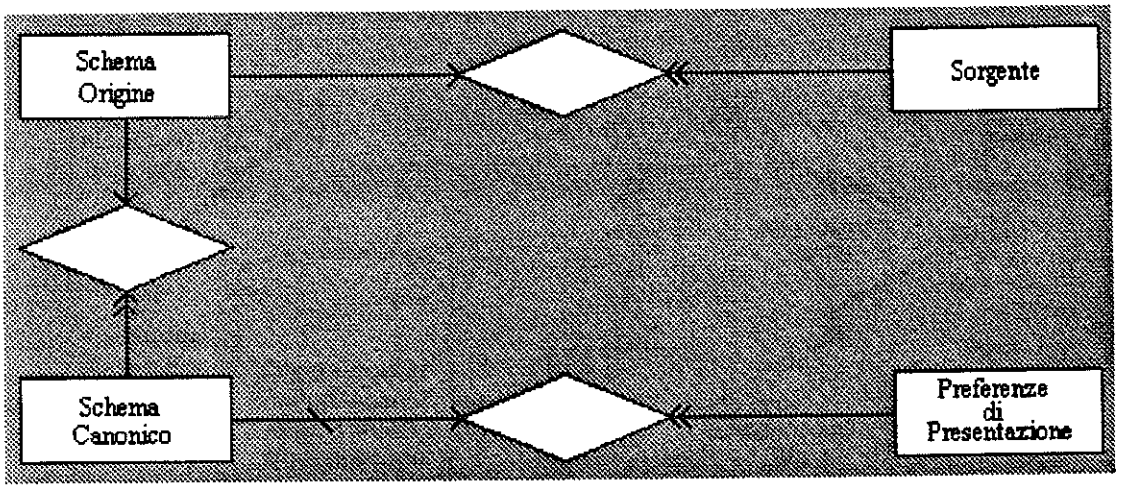

Figura 5. Il modello dei dati necessari al modulo HPS.

Ad ogni sorgente possono corrispondere più schemi canonici diversi a seconda delle esigenze degli utenti come rappresentato in figura 5. Ad esempio per alcuni utenti può essere sufficiente riconoscere documenti bibliografici (corrispondenti a uno schema canonico *Bibliografia*), mentre per altri può essere necessario distinguere le bibliografie scientifiche (con schema canonico *Bibliografie Scientifiche*) da quelle umanistiche (con schema canonico *Bibliografie Umanistiche*). Di volta in volta lo schema canonico al quale associare una certa sorgente viene deciso in base all'ambiente nel quale si deve operare. L'ambiente è costituito da un insieme di associazioni fra sorgenti e schemi canonici come mostrato in figura 6.

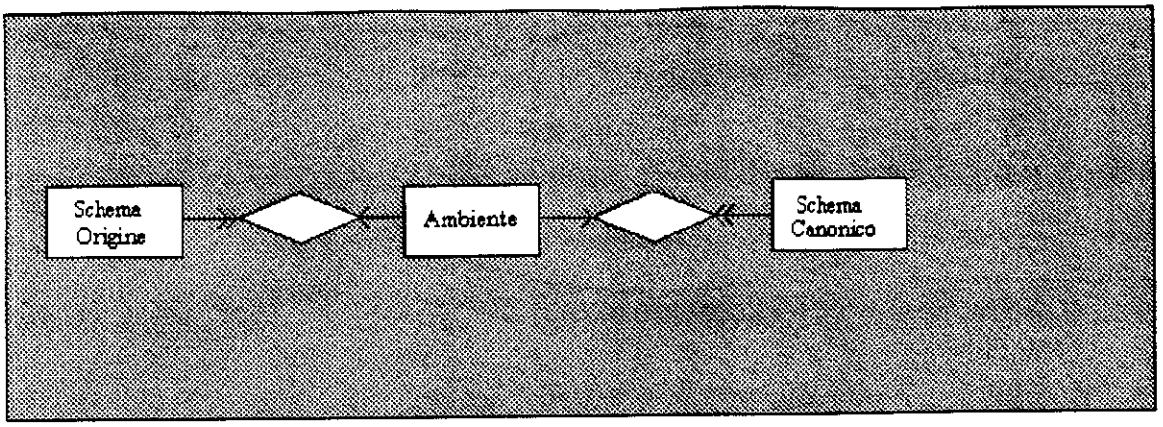

Figura 6. L'ambiente individua l'associazione fra schema origine e schema canonico.

A partire da queste informazioni, espresse secondo un formalismo che verrà illustrato nei prossimi paragrafi, viene generato uno schema di presentazione in base al quale verrà eseguita la traduzione.

Il risultato di questa trasformazione consiste nella generazione di istruzioni HTML per il client HTTP responsabile della presentazione.

Multi Archive Access System - MAAS

Il sottosistema per l'accesso alle sorgenti deve permettere all'utente di interrogare un insieme di sorgenti semanticamente simili utilizzando un client HTTP, anche se tali sorgenti sono accedibili tramite protocolli diversi, come ad esempio WAIS, FTP, o altri. In questo senso MAAS funziona da gateway fra HTTP e i protocolli delle sorgenti. E' possibile interrogare contemporaneamente più sorgenti associabili ad uno stesso schema canonico, anche se fornite con protocolli differenti, in maniera da rendere trasparente all'utente questo meccanismo. Si può dire che il sistema permette di simulare una sorgente con schema origine identico allo schema canonico e che ingloba tutte le sorgenti associabili a tale schema canonico. Questo fa sì che il meccanismo possa diventare ricorsivo, nel senso che un documento tradotto in formato HTML può contenere al suo interno dei riferimenti a questa sorgente simulata in modo tale che i documenti ritrovati percorrendo il link vengano a loro volta tradotti. (figura 7)

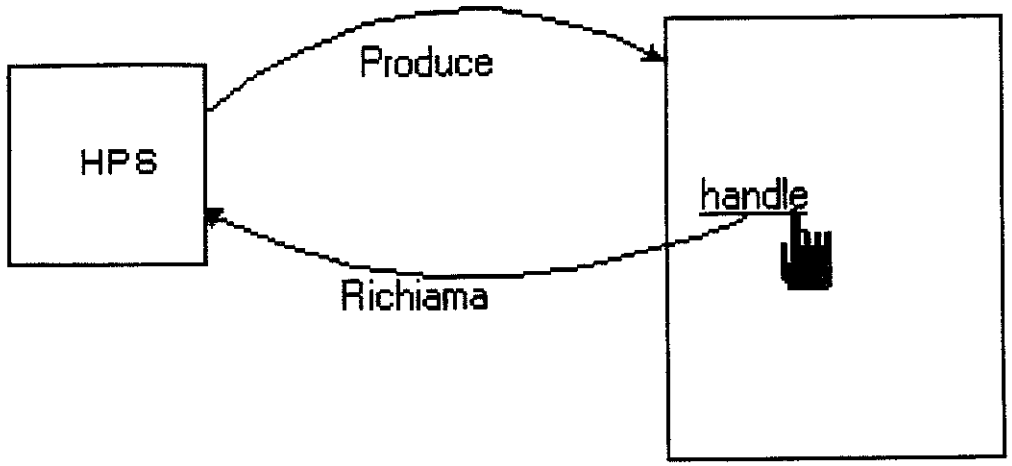

Figura 7. Il sistema può produrre documenti che lo riferiscono stabilendo un processo ricorsivo.

L'architettura del sottosistema MAAS è mostrata in figura 8. Le richieste provenienti dal browser Web sono ricevute dal server HTTP. Tramite il meccanismo del Common Gateway Interface, il server manda in esecuzione MAAS. Il compito principale di questo modulo consiste nel riconoscere il tipo delle sorgenti a cui l'utente vuole accedere e invocare i client opportuni (ASClient). Questi, a loro volta, comunicano con le sorgenti ed eseguono la ricerca e il recupero dei documenti. A questo punto viene invocato il modulo HPS.

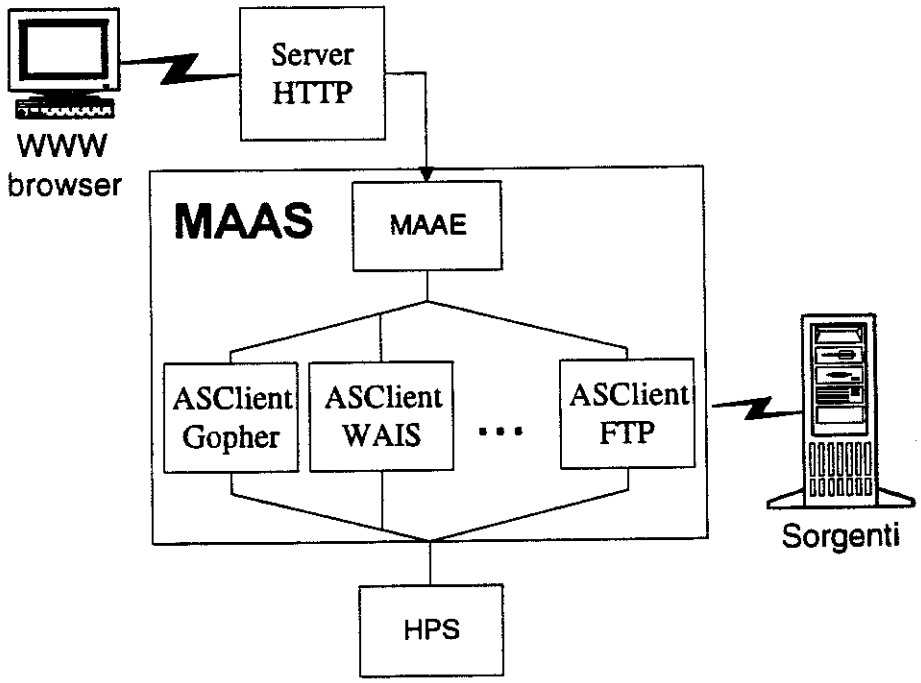

Figura 8. Architettura del sottosistema MAAS.

Sottosistema di presentazione ipertestuale - HPS

L'obiettivo del sottosistema di presentazione ipertestuale è quello di produrre una presentazione uniforme di documenti semanticamente simili. I documenti da presentare possono provenire da più sorgenti, ognuna delle quali può contemplare uno schema origine differente.

Il sistema si compone essenzialmente di due parti, una descrittiva e una operativa. Della prima fanno parte i descrittori di sorgente, che contengono le informazioni sullo schema di origine, le definizioni degli schema canonici e le eventuali preferenze di presentazione dei documenti per ogni specifico schema canonico. La parte operativa è costituita da un modulo che genera documenti HTML e li trasmette al client HTTP. Per effettuare la traduzione il modulo si basa sullo schema origine, sullo schema canonico e sulle preferenze di presentazione associate alle varie istanze di documenti provenienti dalle sorgenti coinvolte dall'interrogazione.

Lo schema canonico

Lo schema canonico incorpora le caratteristiche comuni ad un insieme di schemi origine, ritenute significative per uno specifico ambiente. In generale potrà omettere parti degli schemi di origine e uniformare le intestazioni delle varie parti componenti il documento.

Lo schema canonico è composto da un insieme di dichiarazioni col seguente formato:

c_1 (s_1 e_{11}, ..., s_n e_{n1})
c_2 (s_1 e_{12}, ..., s_n e_{n2})
. . . . .
. . . . .
c_m (s_1 e_{1m}, ..., s_n e_{nm})

- c_i è il nome assegnato nello schema canonico all'i-esimo paragrafo del documento;

- e_{ji} è il nome dichiarato dallo schema origine sj-esimo per tale paragrafo.

Il descrittore di sorgente

Il descrittore di sorgente descrive la sorgente dal punto di vista dell'amministratore del sistema (da qui in avanti HPS Administrator). Alcune informazioni sono fornite dal gestore della sorgente, altre sono aggiunte dall'HPS Administrator in base alle sue conoscenze specifiche e ai bisogni informativi del gruppo di utenza al quale intende fornire l'accesso. Il gestore della sorgente deve fornire l'elenco dei paragrafi che costituiscono lo schema origine con una loro descrizione e la modalità di estrazione dei valori dalle varie istanze di documento. Inoltre il gestore della sorgente può specificare altre caratteristiche quali ad esempio l'elenco dei paragrafi su cui è costruito un indice di accesso, e/o l'elenco degli oggetti che "puntano" documenti di altre sorgenti, etc.

Un descrittore di sorgente contiene la definizione dello schema origine nella seguente forma:

e_1 , t_1;
e_2 , t_2;
.
.
.
e_n , t_n;

- e_i indica l'etichetta, corrispondente al nome del paragrafo nello schema origine, che permette di individuare i valori fornita per una particolare istanza di documento;

- t_i è il tipo associato al paragrafo.

Il tipo di un paragrafo specifica le caratteristiche comuni a tutti i valori che sono forniti per esso. Abbiamo individuato i seguenti tipi:

1. *Sentence*: è un tipo elementare che indica una qualsiasi stringa.

2. *SentenceList*(s): è una lista di *sentence* in cui *s* è il separatore che permette di riconoscere gli elementi di tipo *sentence*. Il separatore può essere una qualsiasi stringa.

3. *Pointer*: un paragrafo è di questo tipo se può contenere delle chiavi con le quali costruire dei riferimenti a ulteriori documenti.

4. *Pointer List*: indica una lista di elementi di tipo *pointer*.

5. *MultiplePointer*: un paragrafo è di questo tipo se dal suo valore si possono estrarre delle chiavi con le quali costruire più URL diverse.

6. *MultiplePointerList*: indica una lista di elementi di tipo *multiple pointer*.

7. *InLineImage*: questo tipo è in realtà una specializzazione del tipo *pointer* per il quale si sappia che l'oggetto puntato è una immagine che si desidera presentare in linea.

Il tipo Pointer

Un paragrafo è di tipo puntatore se nelle sue istanze si possono riconoscere dei valori con i quali si possono eseguire delle ricerche su una qualche sorgente ben definita. Prendiamo, ad esempio, il documento di figura 1 proveniente dal database delle pubblicazioni IEI che era stato trovato formulando una interrogazione sul titolo della pubblicazione. Il paragrafo con etichetta "AU:" contiene il nome dell'autore della pubblicazione. Se l'utente fosse interessato a conoscere ulteriori pubblicazioni dell'autore, dovrebbe formulare, manualmente, una nuova interrogazione, in questo caso, alla stessa sorgente: il Catalogo delle Pubblicazioni dell'Istituto di Elaborazione del CNR. Sarebbe più comodo poter inoltrare questa richiesta in maniera automatica semplicemente selezionando la parola "BERTINO" e aspettandosi che il sistema fosse capace di eseguire di nuovo le richieste sulla stessa sorgente, ma utilizzando la parola chiave "BERTINO" da ricercare nel paragrafo "Autori".

Per poter definire un paragrafo di tipo puntatore occorrono le seguenti informazioni:

- deve essere fornito un meccanismo che permetta di ricercare automaticamente le eventuali chiavi presenti nel paragrafo con cui costruire il riferimento;

- si devono conoscere le informazioni che permettano di costruire la URL che individua la sorgente che si vuole interrogare. In HTML un riferimento a una risorsa è espresso tramite una URL (Universal Resource Locator) che consiste essenzialmente di quattro parti:

  - l'identificatore di un servizio (protocollo);

  - un indirizzo Internet (Internet host name/address);

  - l'identificatore di una porta IP (Internet Protocol port);

  - il nome di una risorsa;

La risorsa può essere un documento in qualsiasi formato (testo, immagine, suono, immagine in movimento) o il nome di un database (nel caso ad esempio di servizio WAIS), il nome di un processo (nel caso di servizio http). Nel nostro contesto il nome della risorsa può essere conosciuto staticamente o può essere costruito a partire da un insieme di parole chiave presenti nell'istanza di paragrafo, in quest'ultimo caso il descrittore di sorgente deve specificare le modalità di costruzione di tale nome.

- In generale, una URL può essere seguita da un insieme di parametri da passare alla risorsa puntata (ad esempio nel caso in cui la risorsa sia di tipo WAIS il parametro può essere il filtro di selezione di documenti), in questa situazione occorre fornire nella definizione del tipo, il nome di una funzione per costruire automaticamente tali parametri.

La definizione del tipo puntatore assume la seguente forma:

Pointer (**KeyFinder, KeyFinderParam, URLParam,**
[QueryConstructor, QueryConstructorParam]);

dove:

- *KeyFinder* e *KeyFinderParam* sono rispettivamente il nome e i parametri di una procedura che permette di riconoscere all'interno del paragrafo insiemi di parole chiave con i quali costruire i riferimenti a documenti attinenti.

- *URLParam* contiene le informazioni per costruire la URL ed assume la seguente forma:

(Service, Host, [Port], [ResourceName | ResourceConstructor,Parameters])

dove *Port* è opzionale così come *ResourceName*. Ove la risorsa non sia definita staticamente, invece, occorre fornire il nome di un *ResourceConstructor*.

- *QueryConstructor* e *QueryConstructorParameters* sono rispettivamente il nome e i parametri di una funzione che costruisce gli eventuali parametri da passare alla sorgente.

Il tipo PointerList

La definizione di un tale tipo assume la seguente forma:

pointer_list(**s, PointerParam**);

dove *s* è il separatore degli elementi di tipo *pointer* e *PointerParam* sono i parametri del tipo pointer (precedentemente descritto). Come esempio per questo tipo si pensi al caso di un documento bibliografico in cui ci sia un paragrafo "Keywords" che contenga un elenco di parole chiave tramite le quali cercare ulteriori documenti relativi allo stesso argomento.

Un possibile risultato è:

Keywords:

- Data Base
- HypertextTrasmission Protocol

Il tipo MultiplePointer

Un paragrafo è di tipo puntatore multiplo se nelle sue istanze è possibile riconoscere dei valori con i quali si possono costruire più interrogazioni diverse. Si consideri ad esempio il caso di un documento bibliografico in cui esista un paragrafo "Autore", per il quale si conosca l'esistenza di una sorgente *biografia* e di una sorgente *bibliografia* relative ad ogni autore. In questo caso, una volta individuate le chiavi contenute nel paragrafo, è possibile costruire due interrogazioni diverse da rivolgere alle due diverse sorgenti.

Per definire un paragrafo di tipo puntatore multiplo non solo deve essere fornito un meccanismo che permetta di ricercare automaticamente le eventuali chiavi presenti nel paragrafo, ma, per ogni sorgente che si vuole interrogare devono essere specificate:

- le informazioni che permettano di costruire la URL che individua la sorgente;

- deve essere fornito un meccanismo che permetta di costruire automaticamente l'interrogazione da rivolgere a tale sorgente;

- deve essere specificata una frase da utilizzare come handle. Tale handle deve spiegare all'utente che tipo di informazioni si possono ricavare percorrendo il riferimento (nell'esempio citato sopra, le handle potrebbero essere le parole "Bibliografia" e "Biografia").

Il tipo MultiplePointerList

La definizione di un tale tipo assume la seguente forma:

MultiplePointer (**KeyFinder, KeyFinderParam, AnchorParamList**)

dove *KeyFinder*, e *KeyFinderParam* hanno lo stesso significato descritto per il tipo *pointer* e *AncorParamList* è una lista di quadruple:

(**URLParam,QueryConstructor, QueryConstructorParameters, Handle**).

I primi 3 elementi della quadrupla hanno lo stesso significato descritto nel caso di paragrafo di tipo pointer e servono quindi a costruire le URL, mentre l'ultimo deve essere una stringa che descriva la risorsa puntata da utilizzare come handle dell'ancora HTML.

La definizione di un tale tipo assume la seguente forma:

MultiplePointerList(s, **MultiplePointerParam**);

dove s è il separatore che permette di distinguere i sottovalori di tipo *MultiplePointer* e *MultiplePointerParam* sono analoghi a quelli descritti per il tipo *MultiplePointer*.

Il tipo InLineImage

La definizione di un tale tipo assume la seguente forma:

InlineImage(**PointerParam**);

dove *PointerParam* sono i parametri descritti per il tipo *pointer*.

Per tutti questi tipi è definito un costruttore di lista. Un paragrafo è di tipo lista se al suo interno è possibile individuare dei sottoparagrafi componenti; utilizzando, ad esempio, un separatore.

I problemi della presentazione uniforme di documenti simili

Nei paragrafi precedenti abbiamo affrontato il problema del riconoscimento di istanze di documenti ed abbiamo a questo proposito introdotto il concetto di tipo associato ad ogni paragrafo. Siamo ora in grado di trattare in maniera più esaustiva l'aspetto, più volte accennato, riguardante le modalità di presentazione di istanze di documenti provenienti da sorgenti diverse. L'obiettivo è quello di individuare delle regole che permettano di arrivare a produrre una presentazione il più possibile uniforme delle varie istanze risultanti da una interrogazione. Questo meccanismo di trasformazione non può avvenire in maniera completamente automatica senza pregiudicare l'efficacia della presentazione o addirittura la perdita di informazione. Il meccanismo di omogeneizzazione delle istanze da presentare si basa essenzialmente su specifiche preferenze espresse dall'utilizzatore.

Supponiamo di avere due documenti, d_1 e d_2, semanticamente simili, ma provenienti da sorgenti diverse. Tali documenti possono differire per uno dei seguenti aspetti:

1. d_1 può avere uno o più paragrafi non contenuti in d_2;

2. d_2 può avere uno o più paragrafi non contenuti in d_1;

3. i due documenti possono utilizzare nomi diversi per paragrafi uguali;

4. i due documenti possono presentare i paragrafi uguali in ordine diverso;

5. gli schemi origine dei due documenti possono dichiarare paragrafi uguali con tipi diversi.

Le differenze descritte ai punti 1 e 2 possono essere presenti anche all'interno di documenti provenienti da una stessa sorgente.

Definire il concetto di uniformità di presentazione significa rispondere alla domanda: "È possibile e necessario eliminare tutte o alcune delle differenze sopra elencate?".

Bisogna innanzi tutto tener presente che nell'effettuare le modifiche si deve preservare il significato originale dei documenti.

Gli aspetti di cui si è parlato ai punti 1 e 2 possono essere risolti utilizzando lo schema canonico.

Una soluzione semplice e immediata al problema consiste nel presentare tutti e soli i paragrafi presenti in entrambi i documenti. Ciò rende visivamente più gradevole la presentazione dei documenti e semplifica la comprenzione dei documenti a scapito però della perdita di una parte delle informazioni.

Una soluzione altrettanto semplice consiste nell'inserimento di tutti i paragrafi per entrambi i documenti, presentandone l'intestazione e mantenendo il valore nullo per le istanze di documenti che non contemplano i paragrafi previsti dallo schema canonico. Questa soluzione preserva il contenuto informativo dei documenti di origine, ma porta alla produzione di documenti grandi, specialmente nel caso in cui i paragrafi diversi fra i due documenti siano molti, che però hanno un piccolo contenuto informativo.

Modificare l'ordine di presentazione dei paragrafi e scegliere un nome uguale per tutti i paragrafi uguali non modifica il significato dei documenti.

L'ultimo punto elencato propone il problema di uniformare i tipi dei paragrafi uguali. Questo comporta delle difficoltà più grosse in quanto in alcuni casi la cosa non è proprio possibile.

La trasformazione è possibile solo rispettando le seguenti regole:

- È sempre possibile presentare un tipo Lista come se fosse del suo sottotipo (ad esempio da *SentenceList* può essere presentata come *Sentence*); in questo caso i vari elementi della lista saranno separati con il separatore utilizzato nel documento di origine.

- È possibile presentare un paragrafo di tipo elementare come Lista di elementi dello stesso tipo. In questo caso il paragrafo sarà costituito da una lista di un solo elemento.

- È possibile presentare ogni tipo come *Sentence*; in un paragrafo di tipo *MultiplePointer*, ad esempio, non saranno presentate le *handle*.

- È possibile trasformare un tipo *MultiplePointer* in *Pointer*, solo se è definito un preciso criterio di scelta di una delle ancore. Ad esempio si può utilizzare sempre la prima ancora definita dallo schema origine.

- Le altre trasformazioni non possono essere effettuate per mancanza di informazioni.

In ogni caso una trasformazione di tipo comporta la perdita di informazioni.

Non è possibile decidere automaticamente fino a che punto uniformare i documenti, a tale scopo il sistema proposto permette all'utente di definire i parametri di uniformità come verrà descritto in dettaglio nel prossimo paragrafo.

Preferenze di presentazione

Lo scopo del modulo di preferenze è quello di permettere agli amministratori di ogni HPSserver, di modificare il più liberamente possibile la presentazione. Occorre tener presente che le preferenze sono associate allo schema canonico e non al modello delle singole sorgenti.

Si possono dare preferenze sulla presentazione di singoli paragrafi, di gruppi di paragrafi o dell'intero documento.
Le preferenze per ogni singolo paragrafo possono essere espresse nella forma:

c, Preferences

dove **c** è il nome canonico del paragrafo e Preferences può comprendere una o più delle seguenti opzioni:

n, fn, in, fv, lt, cp, o, u;

- **n** intestazione da usare per il paragrafo;

- **fn** *font* da usare per l' intestazione;

- **in** numero di interlinee che seguono l'intestazione;

- **fv** *font* per il valore del paragrafo. I *font* ammessi sono tutti quelli supportati dall'HTML;

- **lt** formato di lista da utilizzare per paragrafi con tipo lista. Sono ammessi tutti i formati supportati dall'HTML;

- **cp** posizione del commento per il tipo immagine in linea (apice, pedice o centrale);

- **o** opzionalità del paragrafo;

- **u** grado di uniformità del tipo del paragrafo;

Per quanto riguarda l'uniformazione in base ai tipi dei paragrafi è possibile scegliere fra quelle definite al paragrafo precedente.

Si possono definire sei gradi di opzionalità dei paragrafi:

opzionalità 0: il paragrafo non va presentato mai.

opzionalità 1: il paragrafo va presentato se e solo se il documento di origine lo contiene;

opzionalità 2: il paragrafo va presentato se e solo se lo schema origine del documento lo prevede;

opzionalità 3: il paragrafo va presentato se e solo se almeno uno dei documenti da presentare contemporaneamente lo contiene;

opzionalità 4: il paragrafo va presentato se e solo se almeno uno degli schema origine dei documenti da presentare lo prevede;

opzionalità 5: il paragrafo va presentato sempre.

Per gruppi di paragrafi si può richiedere una disposizione particolare sulla pagina, per esempio, due o più paragrafi potranno essere rappresentati appaiati, etc. Per ottenere questo tipo di presentazione l'HTML fornisce il meccanismo delle tabelle.

Sull'intero documento si può decidere l'interlinea da utilizzare fra un paragrafo e l'altro, i colori da utilizzare e l'ordine di presentazione dei paragrafi.

Esempi

Come primo esempio consideriamo il database WAIS (con identificatore ITRV) relativo al *Catalogo Collettivo Nazionale delle Pubblicazioni Periodiche* a cura dell'Istituto per la Ricerca e la Documentazione Scientifica del CNR o ISRDS/CNR. In figura 9 è mostrato il risultato di una interrogazione effettuata tramite un qualsiasi client WAIS, senza l'utilizzo di HPS. La prima difficoltà per un generico utente, a cui venga proposto questa istanza di documento, è quella di comprendere il significato dei paragrafi in esso contenuti. Un paragrafo di particolare interesse è quello indicato con il nome LC che contiene i riferimenti alle biblioteche italiane presso cui si può reperire il periodico in oggetto. Per avere ulteriori informazioni su tali biblioteche l'utente deve formulare una interrogazione al *Database delle biblioteche italiane*[5]. In Figura 10 è mostrato l'istanza di documento relativa al codice BO010.

In figura 11 è mostrato lo schema origine per il database ITRV, così come recepito dal nostro amministratore HPS. In figura 12 è mostrato lo schema canonico *Bibliografia* associato allo schema origine di ITRV. Le istanze di documento prodotte da HPS a partire dai documenti di figura 9 e 10, sono mostrate in figura 13 e 14. Come si può notare, dallo schema canonico sono stati omessi alcuni paragrafi dello schema di origine in quanto non ritenuti significativi per lo specifico esempio di ambiente di lavoro. Il paragrafo *Location* di Figura 13 è dichiarato di tipo *PointerList*. Selezionando con il mouse la handle BO010 si ottiene il documento presentato in figura 14. Le modalità di presentazione sono quelle assunte per default tranne che per il paragrafo *Location* per cui è stata espressa la preferenza di presentazione come *ordered list* (default *unordered list*).

[5]Anche questo è un database WAIS a cura dell'ISRDS/CNR.

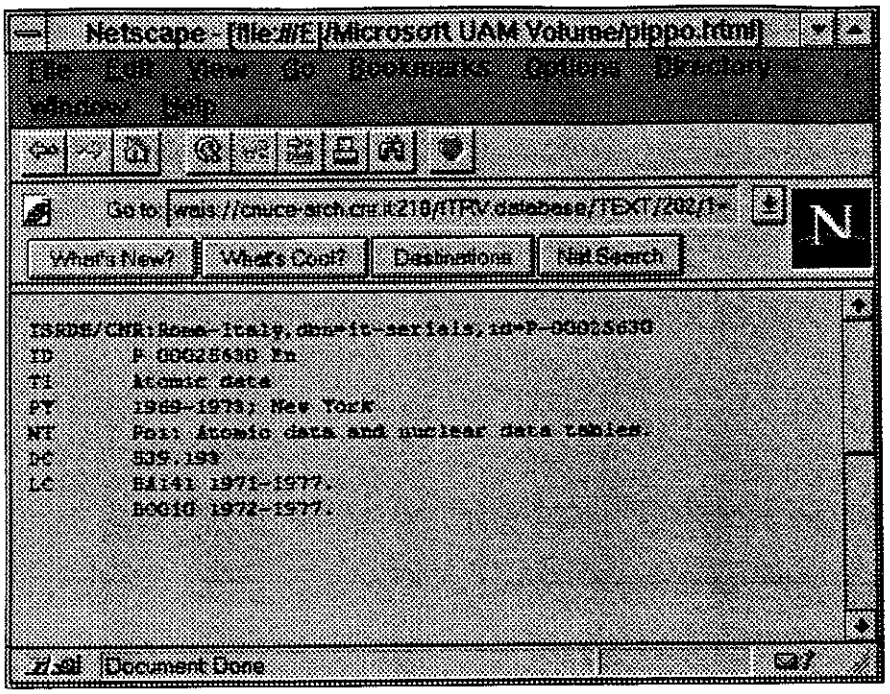

Figura 9. Esempio di interrogazione senza l'utilizzo di HPS

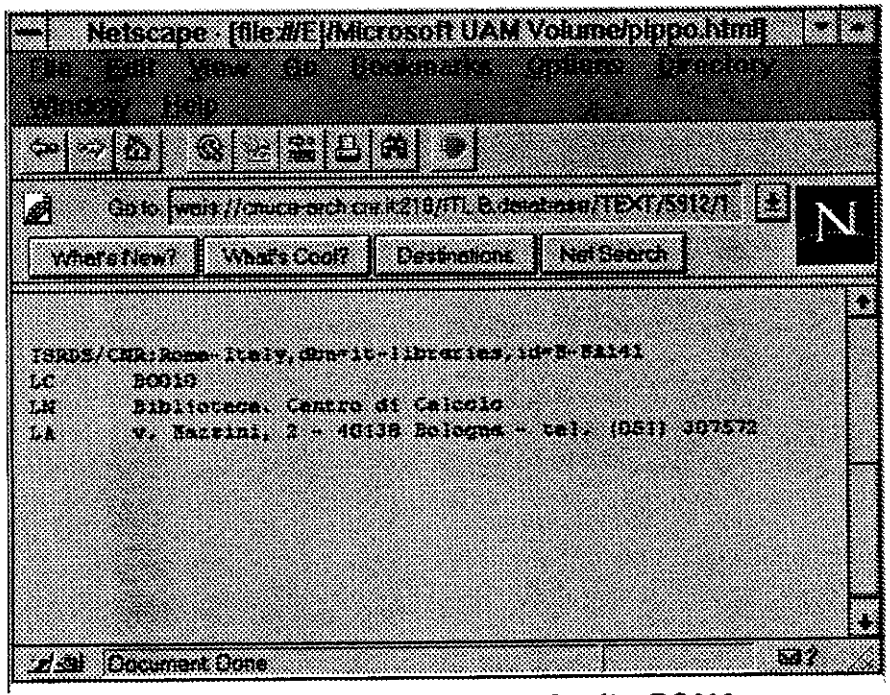

Figura 10. Istanza di documento relativa al codice BO010.

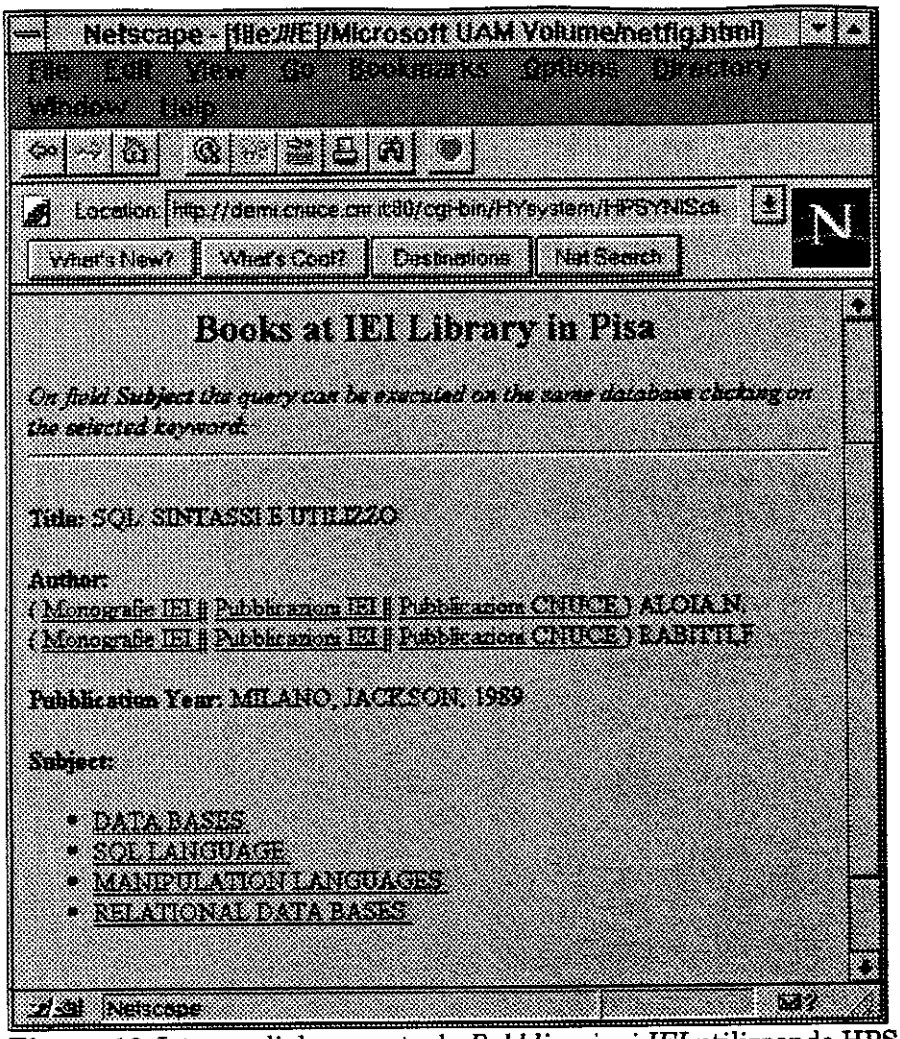

Figura 16. Istanza di documento da *Pubblicazioni IEI* utilizzando HPS

Conclusioni

Il software è stato implementato in ambiente UNIX-LINUX, utilizzando il compilatore gcc, e portato in ambiente SUN/OS ed è disponibile freeware alla seguente URL:

`ftp://ftp.nis.garr.it/pub/WAIS/HTTPtoWAIS/HPS.tar.gz`

Con il software è distribuito anche il file "Install.help" che contiene le informazioni per effettuare l'installazione e la descrizione dettagliata della sintassi per la definizione di schema origine, schema canonico e preferenze di presentazione.

Un esempio di applicazione è attualmente operativa sulla Biblioteca Virtuale del CNR alla seguente URL:

`http://biblio.area.pi.cnr.it/LIB/lib.html`

selezionando la handle "Un sistema integrato di OPACipertestuale".

Bibliografia

T. Berners-Lee, MIT/W3C, D. Connolly, "Hypertext Markup Language -2.0", rfc 1866, novembre 1995.

D. M. Chandler, B. Kirkner, J.Minatel, "Running a Perfect Web Site", Que, ISBN 0-7897-0210-X.

R, Furuta, C. Plaisant, B. Shneiderman, "Automaticlly transforming linear documents into hypertext", *Electronic Publishing: Origination, Dissemination and Design,* 1989, 2, 4, 211-230.

M. Handley, J. Crowcroft, "World Wide Web: Beneath the Surf", UCL Press, 1994, ISBN: 1-85728-435-6, URL:
http://www.cs.ucl.ac.uk/staff/join/book/book.html

T. Verners-Lee, L. Masinter, M. McCahill, "Uniform Resource Locators (URL)", rfc 1738, dicembre 1994.