

# **HPS: Manuale d'uso**

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](mailto:cnuce.cnr.it)

*CNUCE C96-016*

Pisa, 5 luglio 1996



## Introduzione

HPS<sup>2</sup> è 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....). È in grado di generare documenti HTML a partire da documenti in formato ASCII e nel fare ciò si occupa di fornire una modalità di presentazione omogenea per documenti semanticamente simili seppur forniti da sorgenti differenti.

## Premessa

Il meccanismo di recupero di traduzione 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 pacchetto che viene distribuito comprende anche un software, HYNIS, che fornisce l'accesso al sistema permettendo di eseguire ricerche su database WAIS da client WWW tramite un'interfaccia

---

<sup>2</sup>N.Aloia, S. Giuliano, G.A.Romano, "HPS: Un Sistema di Presentazione Iperstuale". CNUCE Internal Report C96-015.

grafica costruita dinamicamente. Le ricerche possono essere formulate sulle parole dell'intero documento o su specifici paragrafi per i quali è stato costruito un indice di selezione. HYNIS definisce dominio-indice, o semplicemente dominio di un database, il valore semantico di un sottoinsieme dei punti di accesso al database stesso. Se si vuole effettuare un'interrogazione su più database che hanno uno dominio in comune, viene dinamicamente costruita una form contenente un input-field per ogni punto di accesso appartenente a tale dominio. Altrimenti viene presentata un fill-out form in cui inserire il testo della query. Il sistema è configurabile dal database administrator che, a seconda della tipologia di utenza, può sottoporre liste diverse di database e modalità diverse di interazione. HYNIS è il risultato di una modifica apportata ad un software, YNIS, sviluppato nell'agosto del 1995 con il contributo economico della Regione Toscana, Servizio Beni Librari, nell'ambito del progetto ABITO. Per permettere l'integrazione è stata modificata la parte che si occupa di formulare le richieste ai database WAIS e di costruire le handle dei documenti appartenenti all'insieme risultato, YNISiswww, in modo tale che i riferimenti ai documenti di risposta a una ricerca vengano tradotti come URL di tipo HTTP al sistema HPS. HYNIS è la versione di YNIS che permette di accedere a HPS. Per quanto riguarda la configurazione di HYNIS si rimanda al rapporto *"Una interfaccia grafica per l'accesso a database WAIS da browser WWW"* L. Apis, G. A. Romano (CNUCE C96-11).

## **Come invocare il sistema HPS**

Il sistema può essere acceduto da un qualsiasi browser WWW utilizzando una URL di tipo HTTP. Il passaggio dei parametri è implementato seguendo lo standard di comunicazione descritto dal CGI e utilizzando il metodo "GET". Le variabili di ambiente utilizzate sono:

- PATH\_INFO: contiene la directory di lavoro in cui si trovano i descrittori delle sorgenti, ecc..

- QUERY\_STRING: contiene il nome del descrittore della sorgente da interrogare seguita dal nome del descrittore di ambiente e da una lista di parametri specifici del Client da invocare. Attualmente l'unico Client disponibile è di tipo WAIS. In questo caso il primo parametro è il metodo di operatività del Client. Sono stati definiti quindi i seguenti metodi:

1. "src": richiede l'esecuzione del metodo di ricerca;
2. "get": richiede l'esecuzione del metodo di recupero di un documento.
3. "src-get": richiede l'esecuzione del metodo di ricerca e recupero dei documenti selezionati.

nel caso di metodo get, il successivo parametro è una codifica del documentID, negli altri casi seguono il numero massimo di documenti da recuperare e il filtro di selezione espresso secondo il formalismo specifico della sorgente<sup>3</sup>.

- REQUEST\_METHOD: contiene il metodo con cui sono inoltrati i parametri. Il modulo può essere invocato solo utilizzando il metodo "GET".

Ad esempio per effettuare una ricerca sul Catalogo delle Pubblicazioni dell'Istituto IEI (iei-p.db) utilizzando il sistema (HPS) installato sulla macchina il cui nome è "demi.cnuce.cnr.it", si può utilizzare la seguente URL:

```
http:\\demi.cnuce.cnr.it\\cgi-bin\\HPS\\home\\HPS-  
System\\conf?  
iei-p.db;env1;src;40;au=carlesi%20and%20au=venerosi
```

in questo caso le variabili di ambiente assumono i seguenti valori:

---

<sup>3</sup> si ricorda che nelle URL gli spazi devono essere sostituiti con il suo valore esadecimale preceduto da un simbolo di percentuale (%20).

```
REQUEST_METHOD="GET";  
PATH_INFO = "home\HPS-System\conf";  
QUERY_STRING =  
    "iei-p1.db;env1;src;40;au=carlesi%20and%20au=venerosi"
```

in particolare, per quanto riguarda l'ultima variabile, "iei-p.db" è il nome del file descrittore della sorgente che si vuole interrogare; "env1" è il nome del file descrittore dell'ambiente nel quale opera l'utente che ha formulato la richiesta. Il resto della stringa contenuta nella variabile QUERY\_STRING \_ costituita da parametri specifici del Client da mandare in esecuzione. I file devono essere ricercati nella directory "home\HPS-System\conf".

## **Installare il software HPS**

Il software di HPS è disponibile nel file HPS.tar.gz presso l'anonymous ftp server di ftp.nis.garr.it con i seguenti comandi:

```
binary  
get pub/WAIS/HTTPtoWAIS/HPS.tar.gz
```

Per installarlo occorre decomprimerlo e suddividerlo nelle sue componenti utilizzando i seguenti comandi:

```
gunzip  
tar -xvf HPS.tar
```

A questo punto nella directory <HPS-System> si troveranno le seguenti directory:

- 1) src: contiene le procedure di compilazione (Makefile) e i file sorgenti;
- 2) bin: contiene gli eseguibili per le piattaforme LINUX e SUNOS-4.1.3 e IBMAIX-3.2.5;

3) html: contiene dei file in formato HTML di utilità;

4) conf: contiene i file di configurazione che contengono le informazioni necessarie per accedere alle sorgenti e per interpretare i documenti provenienti da esse. Si hanno i seguenti file:

- descrittori di ambiente: contengono le associazioni fra schema sorgente e schema canonico;

- descrittori di sorgente: contengono le informazioni per l'accesso alla sorgente e lo schema di origine dei documenti che producono. Il nome di un file descrittore di sorgente può coincidere con il nome del database che identifica la sorgente;

- descrittori canonici: contengono le definizioni degli schema canonici. I nomi dei descrittori canonici devono avere una estensione <.canonical>. Ad esempio, nel caso di bibliografie si potrà utilizzare il nome <bibliografia.canonical>;

- descrittori di preferenze: contengono le preferenze di presentazione dei documenti per ogni specifico schema canonico. I nomi di questi descrittori devono avere la stessa radice del descrittore canonico a cui si riferiscono seguito dall'estensione <.preferences>. Nel caso dell'esempio sopra citato, il nome del descrittore di preferenze dovrà essere <bibliografia.preferences>;

- i file per la configurazione di HYNIS.

## **L'ambiente**

Il file descrittore di ambiente contiene le associazioni fra file descrittori di sorgente e file canonici.

La sintassi per definire un file descrittore di ambiente è la seguente:

<descrittore di ambiente> := <lista\_associazioni>;

<lista\_associazioni> := <associazione> ||  
                          <associazione>  
                          <lista\_associazioni>;

<associazione> :=  
          <nome\_compl\_descrittore\_sorgente> <nome\_compl\_canonico>;

<nome\_compl\_descrittore\_sorgente> := nome completo di path  
          del descrittore di una sorgente;

<nome\_completo\_canonico> := nome completo di path del  
          descrittore di schema canonico associato alla sorgente;

In figura 1 è presentato il descrittore dell'ambiente comprendente un unico schema canonico bibliografia al quale sono associate le sorgenti iei-p.db e cnuce-i-r.db.

```
/home/HYsystem/conf/iei-p.db /home/HYsystem/conf/bibliografia.canonical
/home/HYsystem/conf/cnuce-i-r.db /home/HYsystem/conf/bibliografia.canonical
```

Figura 1. Il file descrittore di ambiente "env1".

## Il descrittore canonico

La sintassi di un descrittore canonico è la seguente:

<descrittore\_canonico> := <lista\_sorgenti>;

```
<lista_sorgenti> :=      <sorgente> ||
                          <sorgente>
                          <lista_sorgenti>;
```

-- deve esserci un <sorgente> per ogni schema sorgente associato allo schema canonico;

```
<sorgente> :=
  <nome_completo_descrittore_sorgente>
  n
  <corrispondenza_nome_etichetta1>
  ...
  <corrispondenza_nome_etichettan>;
```

-- n >= 0, indica il numero delle <corrispondenze\_nome\_etichetta>;

<nome\_completo\_descrittore\_sorgente> := nome completo di path del descrittore di una sorgente associata allo schema canonico;

<corrispondenza\_nome\_etichetta<sub>i</sub>> := <nome\_interno> <etichetta>;

<nome\_interno> := nome interno di paragrafo dichiarato nel descrittore di preferenze (6 caratteri);

<etichetta> := etichetta dichiarata nel descrittore di sorgente per individuare il paragrafo all'interno dei documenti.

Per indicare un commento si utilizza, all'inizio della riga, COMMENT\_CHR, come definito nel file HPS.c.

**Esempio:** per lo schema canonico bibliografia si può avere il<bibliografia.canonical> mostrato in figura 2.

```

# iei-p.db è il nome del database contenente il Catalogo delle
# Pubblicazioni dell'IEI.
/home/HYsystem/conf/iei-p.db
6
HIEI **
autore AU:
titolo TI:
note NO:
a_num1 ND:
p_year PY:
# cnuce-i-r.db è il nome del database contenente il Catalogo
# delle Pubblicazioni del CNUCE
/home/HYsystem/conf/cnuce-i-r.db
5
HCNUCE **
autore AU:
titolo TI:
a_num1 DT:
p_year PY:
#

```

Figura 2. Il file bibliografia.canonical

## Il descrittore di preferenze

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

Le preferenze per ogni singolo paragrafo possono essere espresse rispettando la seguente sintassi:

```

<descrittore di preferenze> :=      n
                                   <preferenze_campo|>

```

...  
<preferenze\_campo<sub>n</sub>>;

-- MAX\_FIELD\_NUMBER >= n >= 0, è il numero dei campi previsti per lo schema canonico a cui sono associate le preferenze, MAX\_FIELD\_NUMBER è definito nel file HPS.h.

<preferenze\_campo<sub>i</sub>> :=  
    <nome\_interno>    <opzionalità>    <uniformità>  
    <stile\_nome\_interno>  <interlinee\_dopo\_nome\_interno>  
    <stile\_valore>  <interlinee\_dopo\_valore>  <stile\_paragrafo>  
    <stile\_lista>  <allineamento\_commento>  <gruppo>  <regola>  
    <nome\_esterno>;

<nome\_interno> := nome interno di paragrafo (6 caratteri);

<opzionalità> := 0 || 1 || 2 || 3 ;

<uniformità> := per futuri sviluppi (1 carattere);

<stile\_nome\_interno> := qualsiasi stile supportato da HTML (8 caratteri);

<interlinee\_dopo\_nome\_interno> := 0 || 1 || ... || 9;

<stile\_valore> := qualsiasi stile supportato da HTML (8 caratteri);

<interlinee\_dopo\_valore> := 0 || 1 || ... || 9;

<stile\_paragrafo> := per futuri sviluppi (1 carattere);

<stile\_lista> := qualsiasi stile di lista supportato da HTML (4 caratteri);

<allineamento\_commento> := per futuri sviluppi nel caso di introduzione di tipo immagine in linea (12 caratteri);

<gruppo> := per futuri sviluppi (1 carattere);

<regola> := per futuri sviluppi (2 caratteri);

<nome\_esterno> := qualsiasi stringa (massimo  
(MAX\_LINE\_LENGTH - 61) caratteri);

-- MAX\_LINE\_LENGTH è definito nel file HPS.c.

```
7
HIEI  0 0          0          0 0 <OL>          a r l <H1>
Pubblicazioni dell'Istituto IEI <HR>
HCNUCE 0 0          0          0 0 <OL>          a r l
Pubblicazioni dell'Istituto CNUCE
autore 0 0 <STRONG> 0          0 0 <OL>          a r l
titolo  0 0 <STRONG> 0          0 0 <OL>          a r l
note    0 0 <STRONG> 0          0 0 <OL>          a r l
a_num1 0 0 <STRONG> 0          0 0 <OL>          a r l
p_year 0 0 <STRONG> 0          0 0 <OL>          a r l
#
```

Figura 3. Il descrittore di preferenze bibliografia.preferences.

## Il descrittore di sorgente

Un descrittore di sorgente si divide essenzialmente in due parti, nella prima sono contenute informazioni per accedere alla sorgente, nella seconda è contenuta la descrizione dello schema origine. Il descrittore segue la seguente sintassi:

```
<descrittore_di_sorgente> :=
    <informazioni_per_accesso>
    <schema_origine>;
```

```
<informazioni_per_accesso> :=
    <servizio>
    <parametri_dipendenti_dal_servizio>
n
```

```

        <informazioni_dipendenti_dal_servizioi>
        ...
        <informazioni_dipendenti_dal_servizion>;

    -- n >= 0, indica il numero delle
    <informazioni_dipendenti_dal_servizio>;

<servizio> := wais || altri servizi se verranno implementati;

    -- nel caso di servizio WAIS:

<parametri_dipendenti_dal_servizio> := <server>;<porta>;

<server> := il server che distribuisce la sorgente;

<porta> := la porta su cui va effettuata la connessione;

<informazioni_dipendenti_dal_servizioi> := una stringa da
    ricercare all'interno del document_ID di documenti di cui
    non si vuole effettuare il recupero;

    -- parte riguardante lo schema origine:

<schema_origine> := <lista_tipi>;

<lista_tipi> :=    <tipo> ||
                  <tipo>
                  <lista_tipi>;

<tipo> :=        <identificatore_tipo>
                  n
                  etichetta1
                  {<dichiarazione_paragrafo{>}
                  ...
                  etichettan
                  {<dichiarazione_paragrafon>}};

<identificatore_tipo> := sentence || sentence_list || pointer ||
    pointer_list || multiple_pointer ||

```

```

multiple_pointer_list;

-- nel caso di tipo sentence <dichiarazione_paragrafoj> è
assente;

-- nel caso di tipo sentence_list:

<dichiarazione_paragrafoj> :=
    <numero_elementi_lista> <separatore>;

-- nel caso di tipo pointer:

<dichiarazione_paragrafoj> :=
    <uffisso_URL>
    <infisso_URL>
    <flag> <numero_chiavi> <key_finder> n <radice_URL>
    <key_finder_param1>
    ...
    <key_finder_paramn>;

-- n >= 0, indica il numero dei <key_finder_param>;

-- nel caso di tipo pointer_list:

<dichiarazione_paragrafoj> :=
    <numero_elementi_lista> <separatore>
    <uffisso_URL>
    <infisso_URL>
    <flag> <key_finder> n <radice_URL>
    <key_finder_param1>
    ...
    <key_finder_paramn>;

-- n >= 0, indica il numero dei <key_finder_param>;

-- nel caso di tipo multiple_pointer:

<dichiarazione_paragrafoj> :=
    m

```

```

    <handle_HTML1>
    <uffisso_URL1>
    <infisso_URL1>
    <radice_URL1>
    ...
    <handle_HTMLm>
    <uffisso_URLm>
    <infisso_URLm>
    <radice_URLm>
    <flag> <numero_chiavi> <key_finder> n
    <key_finder_param1>
    ...
    <key_finder_paramn>;

```

-- m >= 0, indica la molteplicità del puntatore;

-- n >= 0, indica il numero dei <key\_finder\_param>;

-- caso tipo multiple\_pointer\_list:

```

<dichiarazione_paragrafoi> :=
    <numero_elementi_lista> <separatore>
    m
    <handle_HTML1>
    <uffisso_URL1>
    <infisso_URL1>
    <radice_URL1>
    ...
    <handle_HTMLm>
    <uffisso_URLm>
    <infisso_URLm>
    <radice_URLm>
    <flag> <numero_chiavi> <key_finder> n
    <key_finder_param1>
    ...
    <key_finder_paramn>;

```

-- m >= 0, indica la molteplicità del puntatore;

```

-- n >= 0, indica il numero dei <key_finder_param>;

-- in tutti i casi di tipo lista:

-- <numero_elementi_lista> indica il numero massimo di
elementi della lista;

<numero_elementi_lista> := 1 || 2 || ...;

<separatore> := stringa || SPECHAL_CHR;

-- SPECHAL_CHR è un carattere speciale, definito in HPS.c,
che indica che il separatore è il new line;

--in tutti i casi di tipo puntatore:

<uffisso_URL> := stringa;

<infisso_URL> := stringa;

-- <flag> indica se la stringa contenuta fra due chiavi deve
essere considerata come commento per la prima ( valore 0) o
per la seconda (valore1);

<flag> := 0 || 1;

-- <numero_chiavi> indica il numero massimo di chiavi
presenti all'interno del paragrafo:

<numero_chiavi> := 1 || 2 || 3 || ...;

-- <key_finder> è il nome del programma per la ricerca delle
chiavi all'interno del paragrafo:

<key_finder> := keyfinder || altri nomi (9 caratteri) se verranno
introdotte nuove funzioni;

```

<radice\_URL> := stringa;

-- nel caso in cui si voglia puntare al sistema HPS occorre specificare come "radice\_URL" :

<radice\_URL> :=

http://<host>/<nome\_completo\_HPS>/<path\_info>?  
<query\_string\_parziale>;

<path\_info> := il contenuto della variabile di ambiente PATH\_INFO come specificato nel paragrafo xxx.

-- il prefisso è opzionale.

<handle\_HTML<sub>i</sub>> := stringa da utilizzare come handle HTML;

-- nel caso di keyfinder n = 3 e:

<key\_finder\_param<sub>1</sub>> := le lettere che possono comparire all'interno di una parola chiave;

<key\_finder\_param<sub>2</sub>> := la lunghezza minima di una parola chiave;

<key\_finder\_param<sub>3</sub>> := una maschera di pattern-matching;

**Esempio:** in figura 4 è fornito un esempio di file descrittore di sorgente per la sorgente del catalogo dell'IEI.

```

wais
biblio.area.pi.cnr.it;210;
2
iei-p.db.cat
iei-p.db.src
sentence
8
**
NO:
SU:
PY:
ND:
NO:
TI:
multiple_pointer_list
1
AU:
100 .
2
IEI

%20and%20au=
http://demi.cnuce.cnr.it:80/cgi-bin/HYsystem/HPSYNISClient/
home/HYsystem/conf?/iei-p.db
;env1;src-get;256;au=
CNUCE

%20and%20au=
http://demi.cnuce.cnr.it:80/cgi-bin/HYsystem/HPSYNISClient/
home/HYsystem/conf?/cnuce-i-r.db;env1;src-get;256;au=
1 20 keyfinder 3
-abcdefghijklmnopqrstuvzABCDEFGHIJLMNOPQRSTUVWXYZ
3
1@*

```

Figura 4. Il file <iei-p.db>.

## Come inserire una nuova sorgente.

Seguendo l'esempio portato avanti nei paragrafi precedenti vediamo ora come sia possibile introdurre una nuova sorgente bibliografica, il Catalogo delle Pubblicazioni dell'IFC ad esempio, nell'ambiente env1.

I passi da eseguire sono i seguenti:

1. deve essere fornito nella directory /HYSsystem/conf il file descrittore di sorgente ifc-p.db;

2. nel file descrittore di ambiente env1 deve essere introdotta la seguente linea:

```
/home/HYsystem/conf/cnuce-i-r.db /home/HYsystem/conf/bibliografia.canonical
```

3. nel file canonico bibliografia.canonical deve essere introdotto il .....

4. nel file di preferenze bibliografia.preferences devono eventualmente essere introdotti i nuovi campi presenti nella sorgente.

#### 4.3.1. La funzione *Keyfinder* implementata

Il "prototype" della funzione *KeyFinder* che è stata implementata è approssimativamente il seguente<sup>4</sup>:

```
char** KeyFinder(char *line, char *WordLetters, int MinWordLen,  
                 char *Mask);
```

dove:

- line: la stringa da cui estrarre le chiavi.
- WordLetters: un insieme di lettere che possono appartenere alle chiavi; per default sono assunte tutte le lettere dell'alfabeto inglese sia maiuscole che minuscole;
- MinWordLen: una lunghezza minima di una chiave; per default è assunto il valore 0;
- Mask: un formato ("maschera") che deve essere rispettato dalle chiavi: se non viene specificato questo parametro vengono accettate tutte le parole che rispettano i punti precedenti;

---

<sup>4</sup>Il prototype completo verrà descritto nell'appendice B.

Una maschera inizia con un numero '1' o '0' che indica se devono essere rispettivamente accettate (1) o respinte (0) le parole che corrispondono alla maschera. Tale numero è seguito da una sequenza dei seguenti caratteri:

- @ : indica un carattere alfabetico;
- # : indica un numero;
- & : indica un carattere alfanumerico;
- \$ : indica un carattere ASCII;
- \* : corrisponde a una qualsiasi sequenza di caratteri.

Schematicamente la funzione esegue i seguenti passi:

1. legge una "parola" dalla stringa, se raggiunge la fine della stringa *s* senza trovare una parola termina. Una parola in questo caso inizia con il primo dei caratteri definiti nel parametro *WordLetters* e termina con la prima occorrenza di carattere non contenuto in *WordLetters* oppure con la fine della stringa *s*;
2. controlla che la lunghezza in caratteri della parola sia maggiore di *MinWordLen*;
3. controlla, nel caso sia stata specificata una maschera, che la parola la rispetti e torna al punto 1 fino a che non raggiunge la fine della stringa *s*;

## APPENDICE A

### L'interfaccia di un generico ASClient

Come si è già detto, l'unico ASClient implementato è di tipo WAIS. E' possibile inserire eventuali altri ASClient rispettando le interfacce che i moduli devono avere verso MAAE.

Il modulo MAAE essenzialmente opera nel seguente modo:

1. controlla che la variabile REQUEST\_METHOD indichi che il metodo di passaggio di parametri utilizzato sia "GET": se non lo È il programma termina, altrimenti costruisce il MIME del documento specificando che si tratta di un documento di testo in formato HTML;
2. ricava il nome del descrittore della sorgente da interrogare dalle variabili di ambiente PATH\_INFO e QUERY\_STRING; legge nel descrittore il tipo della sorgente da consultare. Nel caso in cui la sorgente non sia di tipo WAIS termina;
4. se la sorgente è di tipo WAIS richiama l'ASClient WAIS.

Per inserire un nuovo ASClient occorre innanzi tutto modificare il file <HPSClient.c> che contiene appunto il modulo MAAE in maniera tale che riconosca sorgenti del nuovo tipo e che invochi l'appropriato ASClient.

MAAE conosce il nome e la collocazione del descrittore di sorgente per poterci leggere il tipo della sorgente. I nomi completi dei

descrittori di sorgente e di ambiente sono parametri che è necessario passare a qualsiasi ASClient da implementare.

```
sturct _source_ido {
    char pchSourceDescriptorFile[MAX_FILE_NAME_LENGTH];
    char **ppchIdo;
        }source_ido;

struct _parameters {
    char          *pchEnv;
    source_ido    *SourceIdo[MAX_SOURCE_NUMBER+1];
        }parameters;
```

Figura 5. L'interfaccia verso il modulo HPS.

L'interfaccia di uscita è costituita dai parametri di ingresso del modulo HPS. In figura 5 è descritta la struttura utilizzata per il passaggio dei parametri, queste strutture sono definite nel file <HPSido.h>. Come si può vedere si tratta di una coppia:

- una stringa, che contiene il nome del file descrittore di ambiente;
- un array di record in cui il primo campo contiene il nome del descrittore della sorgente e il secondo campo È a sua volta un array di documenti provenienti da quella sorgente e che devono essere tradotti in HTML.

Nella versione attuale MAAE è in grado di interrogare una sorgente alla volta, ma il modulo HPS è in grado di tradurre documenti

provenienti da sorgenti diverse, ammesso che siano dello stesso tipo canonico, uniformandone il modello di presentazione.

## APPENDICE B

### L'interfaccia di un generico KeyFinder

Il prototype completo di una generica funzione per la ricerca delle chiavi all'interno di un paragrafo È il seguente:

```
static char *
GenericKeyFinder(char *line, char *l_end, int pos, pointer
*PointerValues, char *ppchParameters);
```

dove:

- line: è il puntatore all'inizio del segmento di documento da cui si vogliono estrarre le chiavi;
- l\_end: è il puntatore alla fine di tale segmento;
- pos: indica a quale chiave si vuole associare la porzione di stringa compresa fra due di esse;
  - 0: deve essere associato alla chiave di sinistra;
  - 1: deve essere associato alla chiave di destra;
- PointerValues: è la struttura nella quale devono essere inseriti i puntatori all'inizio e alla fine delle chiavi individuate (figura 6);
- ppchParameters: è un array di parametri alfanumerici specifici per la funzione, tali parametri sono ricavati dal source descriptor file.

Il modulo HPS delega l'invocazione di *GenericKeyFinder* a una funzione *key\_resolver* che effettua essenzialmente uno switch sul nome della funzione. Per inserire una nuova funzione di ricerca di

chiavi è necessario modificare la *key\_resolver* in maniera tale che riconosca il nome della nuova funzione.

La *key\_resolver* è contenuta nel file <keyfinder.c>.

```
struct _sentence {
    char      *pchStart;
    char      *pchEnd;
}sentence;

struct _pointer {
    sentence  **BeforeString;
    sentence  **KeyValue;
    sentence  **AfterString;

}parameters;
```

Figura 6. La struttura Pointer come definita nel file <HPS.h>.

