Cross-lingual Information Management from the Web

Vangelis Karkaletsis, Constantine D. Spyropoulos

Software and Knowledge Engineering Laboratory
Institute of Informatics and Telecommunications NCSR "Demokritos"
15310 Athens, Greece
Email: {vangelis,costass}@iit.demokritos.gr
http://www.iit.demokritos.gr/skel/

Abstract. This paper presents a methodology for cross-lingual information management from the Web. The methodology covers all the way from the identification of Web sites of interest (i.e. that contain Web pages relevant to a specific domain) in various languages, to the location of the domain-specific Web pages, to the extraction of specific information from the Web pages and its presentation to the end-user. The methodology has been implemented and evaluated in the context of the IST project CROSSMARC[1].

1 Introduction

The extraction of information from Web sites is a complex task. Most of the information on the Web today is in the form of HTML documents, which are designed for presentation purposes and not for automatic extraction systems. The identification of interesting web pages and the extraction of information from them becomes even harder in multi-lingual societies, where descriptions in web pages are typically written in different languages.

A number of systems have been developed to extract structured data from web pages. Such systems include a set of wrappers that extract the relevant information from multiple web sources and a mediator that presents the extracted information in response to the users' requests. Most of these systems use delimiter-based approaches. Texts processed by them are assumed to convey information in a rigidly structured manner, with entities and features mentioned in a fixed order (e.g. product name always followed by price, then availability), and fixed strings or mark-up acting as delimiters. Though the techniques of delimiter-based approaches have proven to be very efficient with rigidly structured pages, they are not applicable to descriptions written in freer linguistic form.

We present in this paper a methodology for cross-lingual information management from the Web, which covers all the way from the identification of Web sites of interest (i.e. that contain Web pages relevant to a specific domain) in various languages, to the location of the domain-specific Web pages, to the extraction of specific information from the Web pages and its presentation to the end-user. The methodology has been implemented and evaluated in the context of the IST project CROSSMARC.

[1] http://www.iit.demokritos.gr/skel/crossmarc/

The paper presents first the main principles of the methodology. It then describes its implementation in the context of the CROSSMARC project. Finally, it outlines related works and concludes summarizing the current status of our work and presenting our future plans.

2 Methodology

Our goal was to devise a methodology for information extraction from web pages, which would facilitate the coverage of different domains and languages. To achieve this the proposed methodology involves the following basic functions:

- *Domain-specific Web crawling* to discover Web sites containing information about a specific domain.
- *Domain-specific spidering* to identify domain-specific Web pages grouped under the sites discovered by the Crawling function.
- *Information Extraction* to process Web pages collected by the Spidering function, and extract domain facts from them.
- *Information Storage and Retrieval* for maintaining a database of facts for each domain, adding new facts, updating already stored facts and performing queries on the database.
- *Information Presentation* to present the information to the end user according to his/her preferences.

To realize these functions the methodology exploits:

- *language technology* methods to process the content of the web pages in the languages supported (different methods may be used for each language).
- *machine learning* methods in order to facilitate the customisation to new domains,
- domain-specific *ontologies* and the corresponding language-specific lexica in order to facilitate the customisation to new languages and domains,
- *localisation* and *user modelling* techniques in order to provide the results of web pages extraction taking into account the user's personal preferences and constraints.

Additionally, the methodology enforces the use of

- an *open architecture* to enable the experimentation with new methods and techniques either for the existing domains and languages or for new ones,
- a *multi-agent architecture* to ensure clear separation of responsibilities.

The specification of a methodology with the general principles mentioned above is not enough when applied in practice. The implementation of this methodology requires more specific and practical guidelines as well as tools to support them. That's why we started developing a framework to support the task of cross-lingual information management over the web according to the proposed methodology. This framework provides procedures and tools to support each of the basic functions of the methodology. This framework is implemented and evaluated in the context of the IST project CROSSMARC.

3 CROSSMARC Implementation

CROSSMARC can be perceived as an elaborate meta-search engine, which identifies domain-specific information from the Web. The main functions implemented by the system, according to the methodology, are the following:

- Domain-specific Web crawling which is managed by the Crawling Agent. The Crawling Agent consults Web information sources such as search engines and Web directories to discover Web sites containing information about a specific domain (laptops, job offers are the two domains covered in CROSSMARC).
- Domain-specific spidering, which is managed by the Spidering Agent. The Spidering Agent identifies domain-specific Web pages grouped under the sites discovered by the Crawling Agent and feeds them to the Information Extraction Agent.
- Information Extraction, which is managed by the Information Extraction Agent. The Information Extraction Agent manages communication with remote information extraction systems. These systems process Web pages collected by the Spidering Agent and extract domain facts from them. The facts are normalized using a common domain ontology and stored in the system's database.
- Information Storage and Retrieval, which is managed by the Data Storage Agent. Its tasks consist of maintaining a database of facts for each domain, adding new facts, updating already stored facts and performing queries on the database.
- Information Presentation. The information presented to the end user can be adapted to his/her preferences. This user management is taken over by the Personalization Agent.

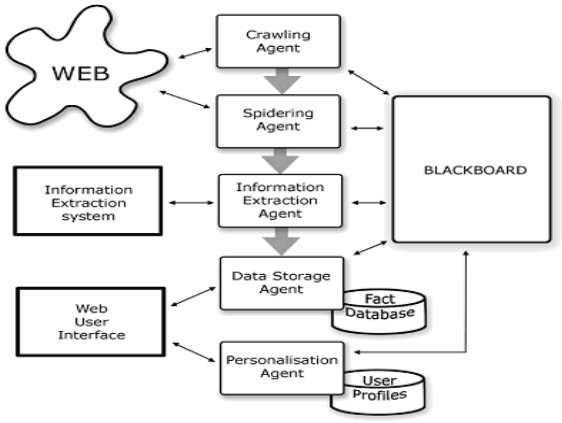

Fig.1.CROSSMARC's agent based architecture

CROSSMARC multi-agent architecture is depicted in Fig. 1. It involves agents for web pages collection (crawling agent, spidering agent), information extraction, data storage and data presentation which communicate through the blackboard.

CROSSMARC, by definition, is a cross-lingual multi-domain system. Our goal was to cover a wide area of possible knowledge domains and a wide range of conceivable facts in each domain for the four languages of the project. Hence we imple-

mented a shallow representation of domain knowledge called the ontology of the domain, which essentially consists of attributes, values and their cardinality inside an attribute [8]. Cross-linguality is achieved with the ontology's lexica, which provide language specific synonyms for all ontology's entries. In the overall processing flow, the ontology plays several key roles:

- During Crawling & Spidering, it comes in use as a "bag of words", that is a rough terminological description of the domain that helps CROSSMARC crawlers and spiders to identify the interesting web pages.
- During Information Extraction, it drives the identification and classification of relevant entities in textual descriptions. Also, ontology is used during fact extraction for the normalization and matching of named entities.
- During Information Storage & Presentation, the lexical layer of the ontology makes possible an easy rendering of a product description from one language to another. User stereotypes maintained by the personalization agent include ontology attributes in order to represent stereotype preferences according to the ontology. Thus, results can be adapted to the preferences of the end user who can also compare uniform summaries of offers descriptions in different languages.

3.1 Crawling

CROSSMARC implementation involves three different crawler versions. The first one exploits the topic-based website hierarchies used by various search engines to return web sites under given points in these hierarchies. The second one takes a given set of queries, exploiting CROSSMARC domain ontology and lexicons, submits them to a search engine, and then returns those sites that correspond to the pages returned. The third one takes a set of 'seed' fit pages and then conducts a 'similar pages' search (available from advanced search engines such as Google) to find pages deemed similar to the seed pages. It then returns the sites corresponding to these pages.

The list of web sites output from the crawler (output of the best combination of the 3 versions based on the evaluation results – see below) is filtered using a light version of the site-specific spidering tool (NEAC) implemented in CROSSMARC (see section 3.2). The light version of NEAC navigates the site until it finds an interesting web page. If it finds one, it considers the site as fit and stops navigating. If no such page is found, the whole site is navigated (this is accelerated through a link scoring mechanism) and if no fit page is found, the site is characterized as unfit.

It is important that the focused crawler should return as many interesting sites as possible. This initial set of sites may later be "reduced" by the spidering process. For reasons of efficiency, however, it is also important that it does not return too high a proportion of uninteresting sites since this would require the site-specific spidering component to perform a great deal of unnecessary processing. A balance between these competing requirements can be obtained by finding the optimal start points for each version of the crawler as well as the optimal combination of versions. In order to find these optimal settings, we performed an evaluation aiming to discover how to maximize the overall effectiveness of the crawler.

The measures for evaluation were the standard measures of *recall, precision, and f-measure*. The recall of the crawler is the ratio of fit sites retrieved by the crawler to all fit sites on the Web. However, since it is not possible to count all fit sites on the Web recall cannot be directly measured. The precision of the crawler is the ratio of fit sites returned by the crawler to all sites returned by the crawler. While it is in principle possible to measure precision by manually inspecting all the sites returned by the crawler, this is impractical given the large number of sites returned. Finally, f-measure is a measure combining the measures of recall and precision defined as *2*Recall* Precision/(Recall+Precision)*. Although we cannot obtain exact figures for precision and recall, it is possible to estimate these measures. For this we needed to estimate how many of the sites returned by the crawler were fit and the number of fit sites on the Web. For the former we performed a manual inspection of a subset of the crawler output. For the latter we had to make the assumption that all fit sites would be found within the combined output of all versions and start points of the crawler. Based on these assumptions (for more details, see [12]), we resulted in the evaluation figures presented in Table 1.

The experience from the application of the specific crawling approach to both domains of the projects led us to certain conclusions about the customization to new domains:

- For any new domain, it seems to be impossible to tell what are the good start points without going through the stages of hypothesis and testing.
- Moreover, the experience of the experiments also seems to show that this cycle needs to be performed per language since results do not appear to translate reliably.
- It is nevertheless possible to minimise the amount of experimenting by putting increased effort into the initial stage of forming hypotheses.

Language	1st Domain: Laptops offers (f-measure)	2nd Domain: Job offers (f-measure)
English	60,6	89,7
Italian	40,0	57,3
Greek	40,5	84,0
French	70,5	-

Table 1. Evaluation results from the application of CROSSMARC crawler in the 2 domains of the project

For the experiments in new domains, the expert will just have to change the settings within the crawler configuration files, i.e. to provide new domain-specific website hierarchies used by various search engines (version 1), give a new set of queries to be submitted to the search engines (version 2), provide a new a set of 'seed' fit pages to be used for a 'similar pages' search by some of the search engines (version 3). The expert will then have to perform the evaluation, following the assumptions presented above, in order to find the optimal start points for each version of the crawler as well as the optimal combination of versions. Customisation into a new domain requires also the customization of the site-specific spidering tool (see section 3.2), a light version of which is used to filter the results of crawler.

3.2 Spidering

CROSSMARC spidering tool comprises of three components:

- Site navigation. It traverses a Web site, collecting information from each page visited, and forwarding part of the collected information to the "Page Filtering" module and another part to the "Link Scoring" module.
- Page filtering. It is responsible for deciding whether a page is an interesting one (e.g. contains laptops offers) and therefore should be stored or not.
- Link scoring. It validates the links to be followed, in order to accelerate site navigation (only links with score above a certain threshold are followed).

As mentioned above, the "Page Filtering" module is responsible for deciding whether a page that has been retrieved is interesting and therefore should be stored or not. A variety of machine learning methods have been evaluated for the page filtering task. In addition to the learning methods, domain-specific heuristic classifiers have been constructed and used for the task. Before feeding the Web page to a classifier, either a machine-learning or a heuristic one, the page is preprocessed, in order to:

- remove all HTML tags, which may affect the classification process, and
- break the text down to a sequence of individual tokens, according to a certain set of delimiting characters, e.g. space, punctuation marks, etc.

For the machine learning based version of the page filtering module, we have developed the *WebPageClassifer* tool which reads a corpus of positive (interesting) and negative Web pages, translates it into the feature vector format required by machine learning algorithms and uses learning algorithms from the WEKA data mining toolkit[2] to construct the Web page classifier. It also exploits the domain ontology and one or more domain lexicon files (more than one lexicons should be given if the Web sites contain pages in various languages).

The heuristic classifier accepts as input the Web page, in the form of a token sequence, and compares each token to a list of regular expressions from the domain lexicon in use. These regular expressions correspond to domain entities that are considered relevant.

In Table 2, we present the evaluation results from the application of the machine learning based version of page filtering in both domains of the project.

Language	1st Domain: Laptops offers (f-measure)	2nd Domain: Job offers (f-measure)
English	97,0	83,0
Italian	94,0	74,0
Greek	93,0	88,0
French	97,0	82,0

Table 2. Evaluation results from the application of CROSSMARC page filtering (machine learning based version) in the 2 domains of the project

The Link scoring module is responsible for the assignment of a "score of interestingness" to the links that are collected by the site navigation module. The goal of this

[2] http://www.cs.waikato.ac.nz/ml/weka/

scoring process is to make the navigation process more efficient. The score is used to order the URLs in the queue of URLs "to be visited". We examined two approaches for link scoring: a machine learning based and a rule-based.

The process of constructing a link scoring function (per language) with the use of machine learning is implemented by a system that takes as input: a collection of domain-specific web sites (these are downloaded locally), the positive web pages within these web sites (identified automatically using two machine learning based classifiers), the domain ontology and one or more domain lexicon files (more than one lexicons should be given if the Web sites contain pages in various languages). Taking this input, the system creates a dataset appropriate for learning a link scoring function. The output of the training, the learned scoring function constitutes the "link scoring" module of the CROSSMARC spidering tool.

In addition to the link scoring function constructed by machine learning, a rule-based link scoring module has also been developed. This takes as input the link's text content as well as its context (left and right). It parses the three strings looking for domain relevant information. The parsing is based on a score-table containing domain relevant regular expressions and their scores. Some of the regular expressions are taken from lexicons and some are hard-coded by the domain expert. The module combines the scores of the three strings using a weighted function.

Both link scoring approaches were evaluated using a common methodology for the 2 domains and the 4 languages of the project. However, the results are rather unsatisfactory, especially in the 1st domain. The fact that both approaches perform rather poorly is a strong indication of the difficulty of the task. Nevertheless, link scoring is an interesting open research issue, which we plan to examine further after the end of the project.

Concerning the spidering customization into new domains:

- the navigation mechanism remains the same;
- the use of the machine learning approach is suggested for page filtering. This requires: the domain ontology and lexicons, the creation of a representative training corpus of positive and negative web pages (CROSSMARC provides a corpus formation tool for this purpose);
- the use of the rule-based approach is suggested for link scoring. This requires the specification of new settings in the configuration file of the link scoring module (specification of groups of terms in different levels according to their significance in the domain) and experimenting with each specification until the optimal setting is found.

3.3 Information Extraction

Information Extraction from the domain-specific web pages collected by the crawling & spidering agents, involves two main sub-stages:

- *named entity recognition (NERC)* to identify named entities (e.g. product manufacturer name, company name) in descriptions inside the web page written in any of the project's four languages [5].

– *fact extraction (FE)* to identify those named entities that fill the slots of the template specifying the information to be extracted from each web page. For this we combine wrapper-induction approaches for fact extraction with language-based information extraction in order to develop site-independent wrappers for the domain.

The architecture of the integrated multi-lingual IE system is a distributed one where the individual monolingual components are autonomous processors, which need not all be installed on the same machine. Figure 2 shows the overall architecture of the Multilingual IE system.

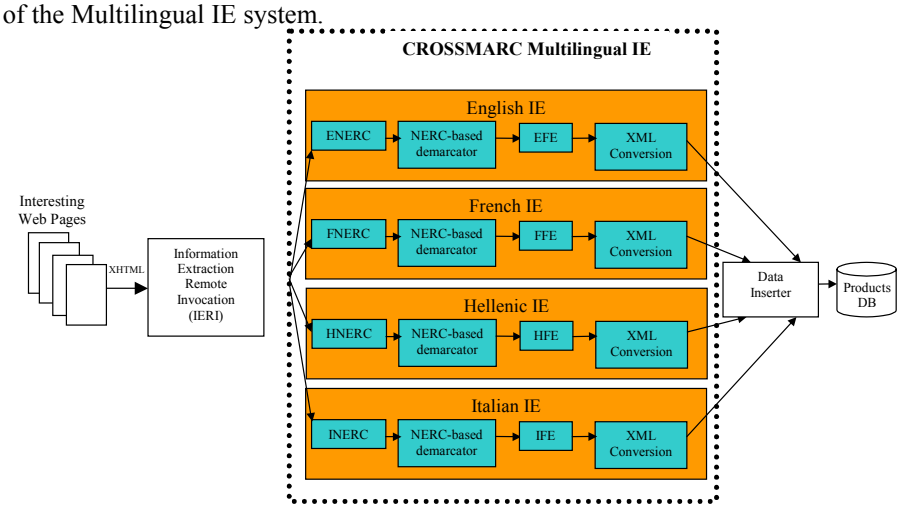

Fig. 2 Architecture of the CROSSMARC IE system

The IE systems are not offered as Web services, therefore a proxy mechanism was required, utilising established remote access mechanisms (e.g. HTTP) to act as a front-end for every IE system in the project. In effect, this proxy mechanism turns every IE system to a Web service. For this purpose, we developed a module named Information Extraction Remote Invocation (IERI) which takes the XHTML pages as input and routes them to the corresponding monolingual IE system according to the language they are written in.

Language	**1st Domain: Laptops offers (f-measure)**	**2nd Domain: Job offers (f-measure)**
English	73,43	51,00
Italian	81,61	-
Greek	86,07	48,00
French	78,49	-

Table 3. Evaluation results from the application of CROSSMARC named entity recognition components in the 2 domains of the project

Although the individual NERC components differ in platforms and annotation methods, they have to produce a common output. According to CROSSMARC specifications, NERC components should add to each XHTML page they process, annota-

tions for the named entities (NE), numeric expressions (NUMEX), time expressions (TIMEX) and TERMs they recognise. For the implementation of the monolingual NERC components, CROSSMARC partners examined two approaches: rule-based for the 1st domain and machine learning based for the 2nd domain. The results are presented in Table 3 (for the 2nd domain the results are preliminary and only for Greek and English).

The rule-based approach has a better performance but it takes a lot of resources to create the rules for a new domain. On the other hand, the machine learning based approach has low results (hopefully, these will be improved in the coming versions) but it enables quick customisation to new domains. We plan to examine the combination of the two approaches in order to develop a semi-automatic approach, which involves:

- use of machine learning to produce a first version of human readable rules;
- editing of the induced rules by a human expert in order to produce the final set of rules for the domain.

The NERC component identifies domain-specific named entities in pages from different sites. The FE component identifies domain-specific facts, i.e. assigns domain-specific roles to some of the entities identified by the NERC module. The FE component is based on wrapper induction (WI) algorithms that capitalize on the page-independent named entity information, rather than relying solely on the HTML tags, which vary among pages from multiple sites. The FE components add an extra attribute to these NEs, NUMEXs, TIMEXs, TERMs that fill a specific fact slot inside a product description in an XHTML page. We wished to experiment with a number of different techniques and have implemented three distinct machine-learning-based Fact Extraction (FE) modules, each of which is applied to all four languages. Two of the FE modules re-implement well-established wrapper induction techniques: the STALKER system [7] and the WHISK approach [11]. The third module treats FE as a classification task and uses a Naive Bayes implementation to perform the classification—this approach bears some similarity to Boosted Wrapper Induction [4]. The evaluation results from the application of the STALKER-based FE technique are presented in Table 4.

Language	**1st Domain: Laptops offers (f-measure)**	**2nd Domain: Job offers (f-measure)**
English	93,80	68,77
Italian	92,73	70,14
Greek	97,59	84,42
French	95,75	72,55

Table 4. Evaluation results from the application of CROSSMARC fact extraction components (STALKER-based version) in the 2 domains of the project

The results in the 2nd domain are lower, due to the fact that the 2nd domain is much richer in textual content compared to the 1st one. The integration of more linguistic knowledge (apart from the named entities) to be exploited by the WI algorithm seems to be the solution to this problem.

In order to store the extracted facts in the products' database, we have to normalise them first according to the ontology. This is necessary in order to be able to present them to the end-user according to the user's preferred locale. For this purpose, the FE components include also normalisation modules. After normalisation is performed, the final processing stage concerns the conversion of the extracted information into a common XML representation, which is used by the Data Storage module in order to feed the products database.

The customization of NERC and FE components to new domains requires the construction of a representative training and testing corpus for each new domain and for all the languages supported. For this purpose, we specified a corpus collection and a corpus annotation methodology which is used in the four languages of the project.

The corpus collection methodology aims at finding a set of characteristics for a certain domain per language, determining how each characteristic must be represented in the training and testing corpora and establishing a set of rules to be followed for the formation of training and testing corpora in the project languages for a given domain. The desired results of its application are the construction of comparable corpora for all languages that reflect the tendencies and possible cultural preferences per domain and per language.

The aim of the annotation process has been the creation of good quality corpora annotated with the named entities and facts of a given domain. Good quality annotated corpora are corpora annotated consistently and according to specific Annotation Guidelines. To ensure the quality of the annotated corpora a standard procedure has been followed and the results are reported for the corpora of each language. Annotation is performed by a user-friendly annotation tool (web annotator) [9].

3.4 Data Storage

The Data Storage (DS) Agent features content storage and content retrieval. It groups facts under datasets, which are identified by the data source. The DS Agent retrieves facts from the blackboard, which were previously stored there by the IE Agent. These facts are used to enrich the fact database of a specific domain. Based on the website they were retrieved from, the DS agents creates new datasets or delete old ones. Also, the DS Agent performs cleaning tasks by periodically scanning its database for facts that have not been updated for a long time which is a likely indicator for a website that previously used to publish such facts to have stopped doing so.

3.5 Data Presentation

The User Interface design is based on a web server application which obtains access to a data source (the Data Storage output) and provides the end user with a web interface for querying datasets. This query interface is customised according to the user's profile. User profiles and user stereotypes are maintained by the personalization server (Pserver). The query interface is depicted in Fig. 3, whereas the results interface in Fig. 4.

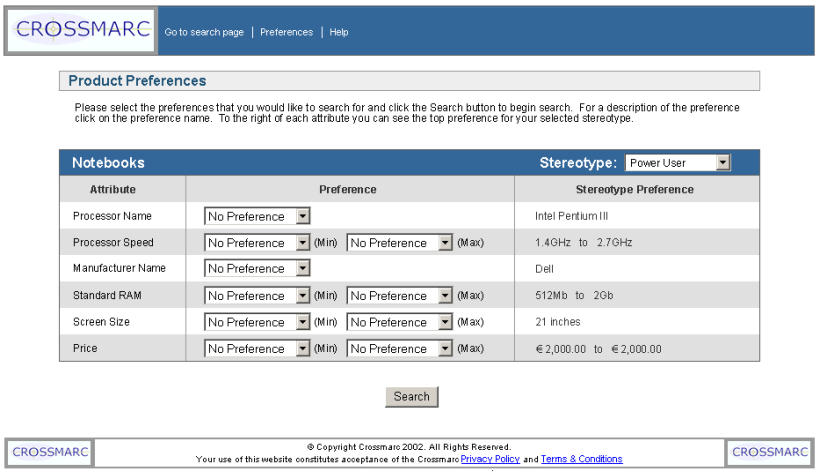

Fig. 3. Query Interface

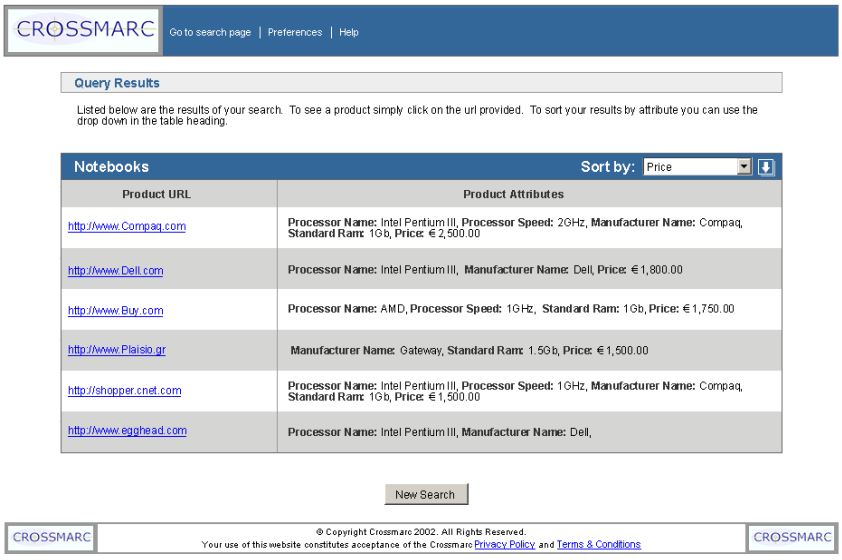

Fig. 4. Results Interface

The adaptation of the personalization subsystem of CROSSMARC to new domains requires the existence of the ontology and lexicons for these domains (these are created using the Protégé-based ontology editor). Once the ontology of the new domain is available, the following steps should be taken:

– The administrator configures PServer so that it can accept requests for the new domain in a new, specially allocated port.

- The User Interface is configured to use the new PServer port whenever the user selects the new domain.
- The administrator uses the Stereotype Editor to create a number of stereotype definitions for the new domain. The Stereotype Editor is a specialized tool that reads the ontology and allows to interactively define stereotypes. It then saves the definitions in a suitable XML format.
- The administrator uses a couple of special applications to remotely upload the stereotype definition XML files to PServer, which is then initialized for the new domain.

From this point on, CROSSMARC system can use PServer for the new domain, supporting all the personalization features.

4 Related Work

The identification and retrieval of Web pages that are relevant to a particular domain or task is a complex process that has been studied by researchers in Artificial Intelligence, Web technologies and databases (e.g. [1], [3]).

The term 'focused crawling' was introduced in [1]. The system described there, starts with a set of representative pages and a topic hierarchy and tries to find more instances of interesting topics in the hierarchy by following the links in the seed pages. Pages are classified into topics, using a probabilistic text classifier. Most text classification studies that deal with HTML documents do not treat the whole document in the same manner. For instance, Craven et al. in [3] use three separate classifiers: one for the words in the URL and the title, one for the words in the anchor text of hyperlinks and one for the body of the text. A more complex approach is to use the DOM tree of the HTML document, in order to separate parts of the text that may refer to different topics, as done in [2].

Apart from the process of identification and retrieval of Web pages that are relevant to a particular domain or task, the information management over the web requires also techniques for extracting information from the retrieved web pages. Kushmerick in [6] first introduced the technique of wrapper induction for Information Extraction from HTML pages. The technique works extremely well for highly structured document collections so long as the structure is similar across all documents, however it is less successful for more heterogeneous collections where structure based clues do not hold for all documents [10].

5 Conclusions

In this paper, we presented a methodology for cross-lingual information management and its implementation in the context of the IST CROSSMARC project. Our methodology covers all the way from the identification of Web sites of interest (i.e. that contain Web pages relevant to a specific domain) in various languages, to the location of the domain-specific Web pages, to the extraction of specific information from the

Web pages and its presentation to the end-user. Concerning the retrieval of domain specific web sites (focused crawling) our approach combines a meta-search engine which can be configured manually for a new domain with a light version of the CROSSMARC spidering module which filters the results of the meta-search engine. Concerning the retrieval of web pages of interest (spidering) within domain-specific web sites (those identified by the focused crawling module), our spider involves modules for web site navigation, page filtering and link scoring. Customisation of the spider into a new domain requires the training of the page filtering and link scoring modules (navigation module remains the same). Training is performed using either machine learning or rule based techniques. Concerning extraction from web pages, CROSSMARC implementation combines language-based IE and wrapper induction based IE in order to develop a site-independent IE technique thus handling the major problem of wrapper induction techniques. Finally, concerning the presentation of the information to the end-user, user modeling techniques are exploited.

The implementation of our methodology required the collaboration of two unpredictable and demanding external entities: the Web and the end user. While the former changes frequently as far as the location of information and the way it is expressed is concerned, the latter assumes the role of a supreme goal: everything, in the end, happens for the user's sake. Thus we need a system that can on one hand adapt to ever changing information sources, extracting as much as possible of accurate information and on the other hand find and present information the user seeks, in a way he or she chooses. In order to realize the 2nd requirement, we adopted an approach that exploits user modeling techniques. User profiles and user stereotypes are maintained in order to deliver the information to the end-user according to his/her preferences as well as taking into account the preferences of other users belonging in the same stereotype with him/her.

Last but not least, the system should be available all the time, bridging operation failures of either external entities or system components. That's why we decided to implement a multi-agent architecture in the context of the CROSSMARC project. A first version of this architecture has already been realized in the 2nd CROSSMARC prototype and has been evaluated. By the end of the project, the final prototype will implement the complete methodology as this is described in the present paper.

References

1. Chakrabarti S., van den Berg M.H., Dom B.E.: Focused Crawling: a New Approach to Topic-specific Web Resource Discovery. *Proceedings 8th International World Wide Web Conference*, Toronto, Canada, (1999)
2. Chakrabarti S., Joshi M., Tawde V.: Enhanced Topic Distillation Using Text, Markup Tags, and Hyperlinks. *Proceedings 24th Annual International ACM SIGIR Conference on Research and development in Information Retrieval*, New Orleans, LO, (2001) 208-216
3. Craven M., DiPasquo D., Freitag D., McCallum A., Mitchell T., Nigam K., Slattery S.: Learning to Construct Knowledge Bases from the World Wide Web. *Artificial Intelligence* 118:69–113 (2000)
4. Freitag D., Kushmerick N.: Boosted Wrapper Induction. *Proceedings 17th Conference on Artificial Intelligence (AAAI),* (2000) 577–583

5. Grover C., McDonald S., Karkaletsis V., Farmakiotou D., Samaritakis G., Petasis G., Pazienza, M.Vindigni M.T., Vichot F., Wolinski F.: Multilingual XML-Based Named Entity Recognition. *Proceedings International Conference on Language Resources and Evaluation (LREC),* Las Palmas, Spain, May (2002)
6. Kushmerick N.: Wrapper Induction for Information Extraction. Ph.D. thesis, University of Washington, (1997)
7. Muslea I., Minton S., Knoblock C.: Stalker: Learning Extraction Rules for Semistructured, Web-based Information Sources. *Proceedings AAAI Workshop on AI and Information Integration,* (1998)
8. Pazienza M.T., Stellato A., Vindigni M., Valarakos A., Karkaletsis V.: Ontology Integration in a Multilingual e-Retail System. *Proceedings Human Computer Interaction International (HCII), Special Session on "Ontologies and Multilinguality in User Interfaces",* Heraklion, Greece, (2003)
9. Sigletos G., Farmakiotou D., Stamatakis K., Paliouras G., Karkaletsis V.: Annotating Web Pages for the Needs of Web Information Extraction Applications. *Proceedings Poster Session 12th International WWW Conference*, Budapest, Hungary, (2003)
10. Soderland S.: Learning to Extract Text-based Information from the World Wide Web. *Proceedings 3rd International Conference in Knowledge Discovery and Data Mining (KDD),* (1997) 251–254.
11. Soderland S.: Learning Information Extraction Rules for Semi-structured and Free Text. *Machine Learning*, 34:233–272 (1999)
12. Stamatakis K., Karkaletsis V., Paliouras G., Horlock J., Grover C., Curran J.R., Dingare S.: Domain-Specific Web Site Identification: the CROSSMARC Focused Web Crawler. *Proceedings 2nd International Workshop on Web Document Analysis (WDA),* Edinburgh, UK, (2003)