Use of Ontologies for Cross-lingual Information Management in the Web

Ben Hachey[*], **Claire Grover**[*], **Vangelis Karkaletsis**[†], **Alexandros Valarakos**[†], **Maria Teresa Pazienza**[◇], **Michele Vindigni**[◇], **Emmanuel Cartier**[‡], **José Coch**[‡]

[*]Division of Informatics, University of Edinburgh
{bhachey, grover}@ed.ac.uk

[†]Institute for Informatics and Telecommunications, NCSR "Demokritos"
{vangelis, alexv}@iit.demokritos.gr

[◇]D.I.S.P., Universita di Roma Tor Vergata
{pazienza, vindigni}@info.uniroma2.it

[‡]Lingway
{emmanuel.cartier,Jose.Coch}@lingway.com

Abstract

We present the ontology-based approach for cross-lingual information management of web content that has been developed by the EC-funded project CROSSMARC. CROSSMARC can be perceived as a meta-search engine, which identifies domain-specific information from the Web. To achieve this, it employs agents for web crawling, spidering, information extraction from web pages, data storage, and data presentation to the user. Domain ontologies are exploited by each of these agents in different ways. The paper presents the ontology structure and maintenance before describing how domain ontologies are exploited by CROSSMARC agents.

1 Introduction

The EC-funded R&D project CROSSMARC[1] proposes a methodology for management of information from web pages across languages. It is a full-scale approach starting with the identification of web sites in various languages that contain pages in a specific domain. Next, the system locates domain-specific web pages within the relevant sites and extracts specific product information from these pages. Finally, the end user interacts with the system through a search interface allowing them to select and view products according to the characteristics they deem important. A unique ontology structure is exploited throughout this process in different ways.

The CROSSMARC architecture is characterised by the following design points:

- machine learning methods to facilitate rapid tailoring of linguistic resources to new domains with minimal human intervention,
- multi-agent architecture ensuring clear separation of module responsibilities, providing the system with a clear interface formalism, and providing robust and intelligent information processing capabilities.
- user modelling and localisation to adapt the information retrieved to the users' preferences and locale.
- domain-specific ontologies and the corresponding language-specific instances

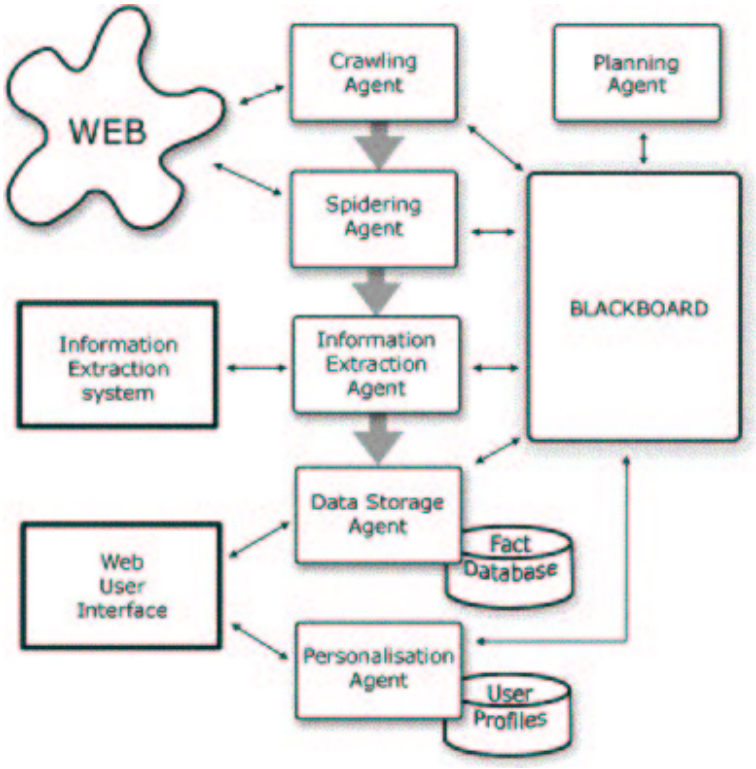
The focus of this paper is the ontology exploitation at various processing stages of the CROSSMARC project.

The main functionality of the system is implemented in the agent modules, which appear in the centre column of Figure 1. Namely, these consist of domain-specific Web crawling, domain-specific spidering, information extraction, information storage and retrieval, and information presentation. We briefly describe the primary functionality of these agent modules below.

Domain-specific Web crawling 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 domains.[2]

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

[2] Two domains are being implemented during the term of the project: laptops and job offers.



Domain-specific spidering 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.

The Information Extraction Agent manages communication with remote information extraction systems (four such systems are employed for the four languages of the project). These systems process Web pages collected by the Spidering Agent and extract domain facts from them (Grover et al., 2002). The facts are stored in the system's database.

Information storage and retrieval 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. Finally, information presentation is managed by the Personalisation Agent, which allows the presentation to be adapted to user preferences and locale.

CROSSMARC is a cross-lingual multi-domain system for product comparison. The goal is to cover a wide area of possible knowledge domains and a wide range of conceivable facts in each domain, hence the CROSSMARC model implements a shallow representation of knowledge for each domain in an ontology (Pazienza et al., 2003). A domain ontology reflects a degree of expert knowledge for that domain. Cross-linguality is achieved through the lexical layer of the ontology, which provides language specific synonyms for all ontology entries. In the overall processing flow, the ontology plays several key roles:

- During Crawling & Spidering, it comes in to 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. It is also used during fact extraction for the normalisation and matching of named entities.

- During Data 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 Personalisation Agent include ontology attributes in order to represent stereotype preferences according to the ontology. Thus, results can be adapted to the pref-

```
<ontology xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance" id="RTVD1-R1.2"
        xsi:noNamespaceSchemaLocation="../XSD/New_Ontology.xsd">
    <description>Laptops</description>
    <features>
        <feature id="OF-d0e5">
            <description>Brand</description>
            <attribute type="basic" id="OA-d0e7">
                <description>Manufacturer Name</description>
                <discrete_set type="open">
                    <value id="OV-d0e3283">
                        <description>Fujitsu-Siemens</description>
                    </value>
                    ...
                </discrete_set>
            </attribute>
            <attribute type="basic" id="OA-d0e349">
                <description>Model Name</description>
                <discrete_set type="open">
                    <value id="EOV-d0e351"><description>Unknown Model</description></value>
                </discrete_set>
            </attribute>
        </feature>
        .....
    </features>
</ontology>
```

Figure 2: Excerpt from XML export of concept instances for the laptop domain

erences of the end user who can also compare uniform summaries of offers descriptions from Web sites written in different languages.

This paper first presents the CROSSMARC ontology and discusses ontology management issues. It then details the manner in which CROSSMARC agents exploit domain-specific ontologies at various processing stages of the multi-agent architecture. It next presents related work before concluding with a summary of the current status of the project and future plans.

2 The CROSSMARC Ontology

2.1 Ontology Structure

The structure of the CROSSMARC ontology has been designed, first, to be flexible enough to be applied to different domains and languages without changing the overall structure and, second, to be easily maintainable by modifying only the appropriate features. For this reason, we have constructed a three-layered structure. The ontology consists of a meta-conceptual layer, a conceptual layer, and an instances layer. The instances layer can be further divided into concept instances and lexical instances, which provide support for multilingual product information. For use by CROSSMARC agents, the concept instances and lexical instances are exported into XML (Figures 2 and 3).

The *meta-conceptual layer* defines the top-level commitments of the CROSSMARC ontology architecture defining the language used in the conceptual layer. It denotes three meta-elements (features, attributes, and values), which are used in the conceptual level to assign computational semantics to elements of the ontology. Also, this layer defines the structure of the templates that will be used in the information extraction phase. In essence, the meta-conceptual layer specifies the top-level semantics of CROSSMARC across domains.

The *conceptual layer* is composed of the concepts that populate the specific domain of interest. These concepts follow the structure defined in the meta-conceptual layer for their internal representation and the relationship amongst them. Each concept element is discriminated by the use of a unique identity (ID) number, which is called onto-reference. This conceptual layer defines the semantics of a given domain. An important aspect of this is the domain-specific information extraction template.

Finally, the *instances layer* represents domain specific individuals. It consists of two types of instances: (1) concept instances that act as the normalised values of each individual, and (2) lexical

```
<lexicon xmlns:xsi="http://www.w3.org/2000/10/XM
LSchema-instance" idref="RTVD1-R1.2" lang="en" x
si:noNamespaceSchemaLocation="../XSD/New_Lexicon
.xsd">
    <node idref="OV-d0e3283">
        <!--Lexical entries for
                            Fujitsu-Siemens-->
        <synonym>Fujitsu</synonym>
        <synonym>FUJITSU</synonym>
        <synonym>Siemens</synonym>
        <synonym>SIEMENS</synonym>
        <synonym>Fujitsu Siemens</synonym>
        <synonym>Fujitsu-Siemens</synonym>
        <synonym>FUJITSU-SIEMENS</synonym>
        <synonym>FUJISTU SIEMENS</synonym>
    </node>
    ...
</lexicon>
```

Figure 3: Excerpt from the XML export of English lexical instances for the laptop domain

instances that denote linguistic relationships between concepts or instances for each natural language. Concepts are instantiated in this layer by populating their attribute(s) with appropriate values. Every instance is unique and a unique identity number, named onto-value, is attributed to it.

As previously mentioned, lexical instances support multi-lingual information. They are instantiated in a domain specific lexicon for each natural language supported (currently English, Greek, French, and Italian). Here, possible instantiations of ontology concepts for each language are listed as synonyms. The "idref" attribute on synonym list nodes associates lexical items with the ontology concept or instances that it corresponds to. Also, regular expressions can be provided for each node of a lexicon for a broader coverage of synonyms.

We can illustrate the overall ontology structure with an example concept instantiation from the laptop domain. Again, the way we describe the structure of the domain is constrained by the *meta-conceptual* layer. The *conceptual layer* defines the items of interest in the domain, for laptops, these include information about the brand (e.g. manufacturer name, model), about the processor (e.g. brand, speed), about preinstalled software (e.g. OS, applications), and etcetera. Finally, in the instances layer, we declare instances of concepts and provide a list of possible lexical realisations. For example, the exported domain ontology in Figure 2 lists 'Fujitsu-Siemens' as an instance of the `manufacturer name` concept and the exported English lexicon in Figure 3 lists alternative lexical instantiations of 'Fujitsu-Siemens'.

Though it is common knowledge that conceptual clustering is different from one language to the next, the ontology structure described is sufficient to deal with product comparison. Firstly, because commercial products are fairly international, cross-cultural concepts. Secondly, the ontology design phase of adding a new domain provides a forum for discussing and addressing linguistic cultural differences.

2.2 Ontology Maintenance

After a survey of existing ontology editors and tools, we decided to use Protégé-2000[3] as the tool for ontology development and maintenance in CROSSMARC. We modified and improved the Protégé model of representation and the user-interface in order to fit CROSSMARC's user needs and to facilitate the process of editing CROSSMARC ontologies. This work has led to the development and release of a range of tab plug-ins dedicated to the editing of sections of the ontology related to specific steps in the Ontology Maintenance Process.

The default Protégé editing Tabs are divided into Class, Slots and Instances. Although this organisation is quite logical, it was impractical for the purposes of CROSSMARC, as the Class view of the knowledge base puts together the Domain Model, the Lexicons, and the Meta layers. For this reasons we developed several plug-in Tabs (described in Table 1) that focus the attention on each different aspect of the knowledge base, allowing for more functional inspection and editing of the specific component under analysis. For more information on ontology maintenance in CROSSMARC, refer to (Pazienza et al., 2003).

3 Ontology Use in CROSSMARC

3.1 Crawling & Spidering

The CROSSMARC implementation of crawling exploits the topic-based website hierarchies used by various search engines to return web sites under given points in these hierarchies. It also takes a given set of queries, exploiting CROSSMARC domain ontologies and lexicons, submits them to a search engine, and then returns those sites that correspond to the pages returned. The list of web sites output from the crawler is filtered using a light version of the site-specific spidering tool (NEAC) im-

[3] `http://protege.stanford.edu/`

Protégé Tab	Maintenance Task
Domain Model Editor	World modelling
Template Editor	Creation of a task-oriented model to be used as template for purposes of fact-extraction
Lexicon Editor	Upgrade of the lexicon for the ontology
Import/Export	Import and Export of the Ontology and Lexicons in XML according to the Schema adopted in CROSSMARC

Table 1: CROSSMARC Protégé Tabs with description of associated maintenance tasks.

plemented in CROSSMARC, which also exploits the ontology.

The CROSSMARC web spidering tool explores each site's hierarchy starting at the top page of the site, scoring the links in the page and following "useful" links. Each visited page is evaluated and if it describes one or more offers, it is classified as positive and is stored in order to be processed by the information extraction agent. Thus, the CROSSMARC web spidering tool integrates decision functions for page classification (filtering) and link scoring.

Supervised machine learning methods are used to create the page classification and link scoring tools. The development of these classifiers requires the construction of a representative training set that will allow the identification of important distinguishing characteristics for the various classes. This is not always a trivial task, particularly so for Web page classification. We devised a simple approach which is based on an interactive process between the user (person responsible for corpus formation) and a simple nearest-neighbour classifier. The resulting Corpus Formation Tool presents difficult pages to the user for manual classification in order to build an accurate domain corpus with positive and negative examples.

For the feature vector representation of the web pages, which is required both by the corpus formation tool and the supervised learning methods, we use the domain ontology and lexicons. A specialised vectorisation module has been developed that translates the ontology and the lexicons into patterns to be matched in web pages. These patterns vary from simple key phrases and their synonyms to complex regular expressions that describe numerical ranges and associated text. The vectorisation module generates such a pattern file (the feature definition file) which is then used by an efficient pattern-matcher to translate a web page into a feature vector. In the resulting binary feature vector, each bit represents the existence of a specific pattern in the corresponding web page. A detailed discussion and evaluation of the CROSSMARC crawling and spidering agents can be found in (Stamatakis et al., 2003).

3.2 Information Extraction

Information Extraction from the domain-specific web pages collected by the crawling & spidering agents, involves two main sub-stages. First, an entity recognition stage identifies named entities (e.g. product manufacturer name, company name) in descriptions inside the web page written in any of the project's four languages (Grover et al., 2002). After this, a fact extraction stage identifies 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.

Although each monolingual information extraction system (four such systems are currently under development) employs different methodologies and tools, the ontology is exploited in about the same way. During the named-entity recognition stage, all the monolingual IE systems employ a gazetteer look up process in order to annotate in the web page those words/phrases that belong to its gazetteers. These gazetteers are produced from the ontology and the corresponding language-specific lexicon through an automatic or semi-automatic process.

During the fact extraction stage, most of the IE systems employ a normalisation module. This runs after the identification of the named entities or expressions that fill a fact slot according to the information extraction template (i.e. the entities representing the product information that will eventually be presented to the end-user). The on-

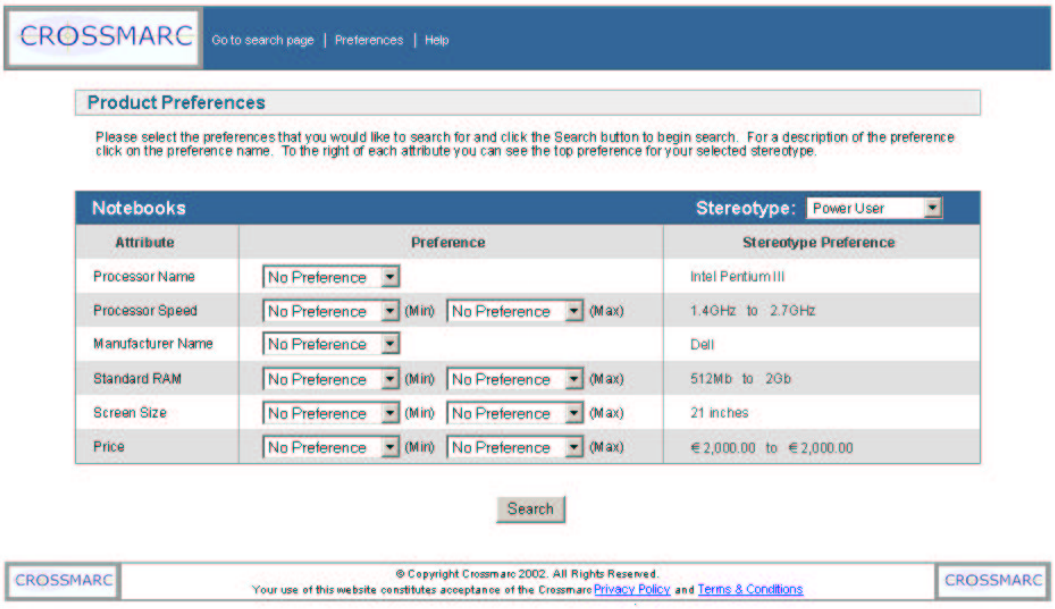


Figure 4: Screen shot of CROSSMARC search form.

tology and the language dependent lexicons are used for the normalisation of the recognised names and expressions that fill fact slots. As a first step, names and expressions are matched against entries in the ontology. If a match is not found, names and expressions are matched against all synonyms in the four lexicons. Whenever a match is found the text is annotated with the characteristic "ontoval" that takes as value the ID of the corresponding node from the domain ontology. If no match is found for a name or expression belonging to a closed class, their "ontoval" characteristic takes the value of the ID of the corresponding "unknown" node. If the name or expression belongs to an open set the ID of the category is returned. In the cases of annotated numeric expressions, the module returns not only the corresponding ID of the ontology node but also the normalised value and unit.

3.3 Information Storage & Presentation

The information extracted and normalised by the monolingual IE systems is stored into the CROSSMARC database by the Data Storage Agent. A separate database is constructed for each domain covered. The structure of the database is determined by the fact extraction schema, which is generated by the Template Editor Tab implemented in Protégé.

The ontology is also exploited for the presentation of information in the CROSSMARC end-user interface. The User Interface design (see Figures 4 and 5) is based on a web server application which accesses the data source (i.e. the Data Storage output) and provides the end user with a web interface for querying data sets. This interface is customised according to a user profile or stereotype maintained by the personalisation agent and defined with respect to the domain ontology. Each query is forwarded to the Data Storage component and query results are presented to the user after subsequent XSL transformation stages. These transformations select the relevant features according to the user's profile and apply appropriate lexical information onto them by accessing the normalised lexical representations corresponding to the user's language preferences.

4 Related Work

In the last years, the increasing importance of the Internet has re-oriented the information extraction community somewhat toward tasks involving texts such as e-mails, web-pages, web-logs and newsgroups. The main problems encountered by this generation of IE systems are the high heterogeneity and the sparseness of the data on such domains. Machine learning techniques and ontologies have been employed to overcome those prob-

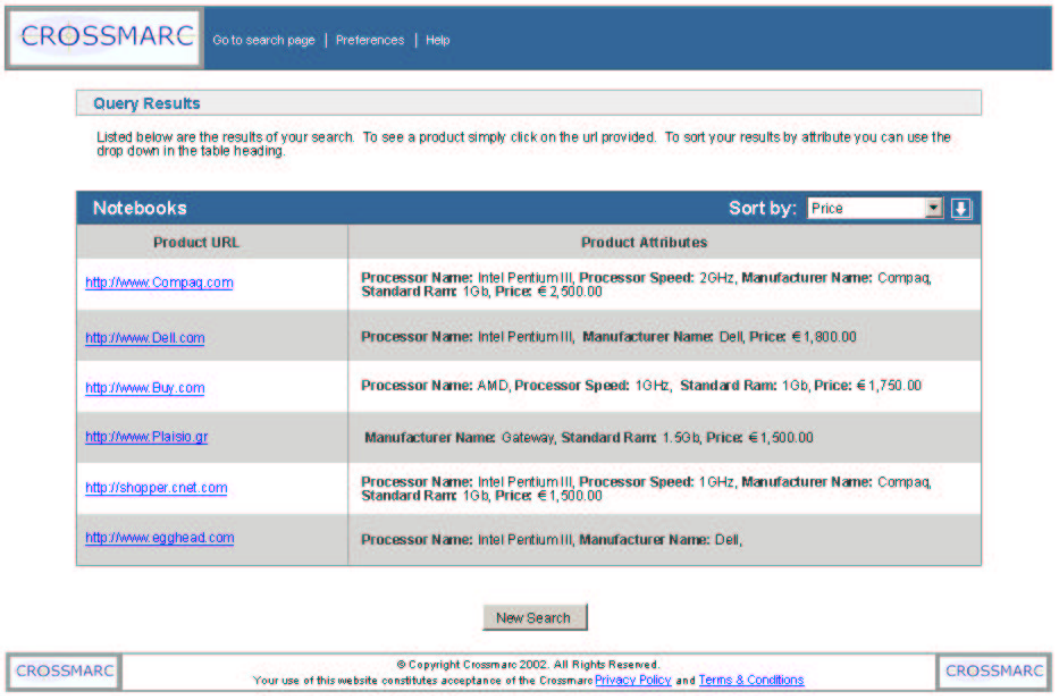


Figure 5: Screen shot of CROSSMARC search results display.

lems and improve system performance. RAPIER (Califf and Mooney, 1997) is such a system that extracts information from computer job postings on USENET newsgroup. It uses a Lexical ontology to exploit the hypernym relationship to generalise over a semantic class of a pre or post filler pattern. Following the same philosophy, CRYSTAL (Soderland et al., 1995) uses a domain ontology to relax the semantic constraints of its concept node definitions by moving up the semantic hierarchy or dropping constraints in order to broaden the coverage. The WAVE (Aseltine, 1999) algorithm exploits a semantic hierarchy restricted to a simple table look-up process to assign a semantic class to each term. And in (Vargas-Vera et al., 2001), an ontology is used to recognise the type of objects and to resolve ambiguities in order to choose the appropriate template for extraction.

IE systems have encountered another limitation as regards the static nature of the background knowledge (i.e. the ontology) they use. For that reason bootstrapping techniques for semantic lexicon and ontology extension during the IE processes have been introduced. (Brewster et al., 2002) uses an ontology to retrieve examples of lexicalisation of relations amongst concepts in a corpus to discover new instances which can be inserted to the ontology after user's validation. In (Maedche et al., 2002) and (Roux et al., 2000), the initial IE model is improved through extension of the ontology's instances or concepts, exploiting syntactic resources.

Ontologies are also used to alleviate the lack of annotated corpora. (Poibeau and Dutoit, 2002) employ an ontology to overcome this limitation for an information extraction task. The use of ontologies in this work is twofold. First, it is used to normalise the corpus by replacing the instances with their corresponding semantic class using a named entity recogniser to specify the instances. Second, it generated patterns exploiting the semantic proximity between two words (where one of them is the word that should be extracted) in order to propose new patterns for extraction. The ontology used in this work is a multilingual net over five languages having more than 100 different kinds of links.

Kavalec (2002) conducted an ontological analysis of web directories and constructed a meta-ontology of directory headings plus a collection of interpretation rules that accompany the meta-ontology. He treats the meta-ontology schema as

a template for IE and uses the ontology's schema and interpretation rules to drive the information extraction process in the sense of filling a template. Another work (Craven et al., 1999) uses an ontology that describes classes and relationships of interest in conjunction with labelled regions of hypertext representing instances of the ontology to create an information extraction method for each desired type of knowledge and construct a knowledge base from the WWW.

5 Current Work and Conclusions

CROSSMARC is a novel, cross-lingual approach to e-retail comparison that is rapidly portable to new domains and languages. The system crawls the web for English, French, Greek, and Italian pages in a particular domain extracting information relevant to product comparison.

We have recently performed a user evaluation of the CROSSMARC system in the first domain. This evaluation consisted of separate user tasks concerning the crawling, spidering, and information extraction agents as well as the end user interface (Figures 4 and 5). We are in the process of analysing the results and are scheduling further user evaluations.

We are also currently porting the system into the domain of job offers. An important result of this will be the formalised customisation strategy. This will detail the engineering process for creating a product comparison system in a new domain, a task that consists broadly of developing a new domain ontology, filling lexicons, and training the crawling, spidering, and information extraction tools.

The CROSSMARC system benefits from a novel, multi-level ontology structure which constrains customisation to new domains. Furthermore, domain ontologies and lexicons provide an important knowledge resource for all component agents.

The resulting system deals automatically with issues that semantic web advocates hope to alleviate. Namely, the web is built for human consumption and thus uses natural language and visual layout to convey content, making it difficult for machines to effectively exploit Web content. CROSSMARC explores an approach to extracting and normalising product information that is adapted to new domains with minimal human effort.

References

J. H. Aseltine. 1999. WAVE: An incremental algorithm for information extraction. In *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI 1999)*.

C. Brewster, F. Ciravegna, and Y. Wilks. 2002. User centered ontology learning for knowledge management. In *Proceedings of 7th International Workshop on Applications of Natural Language to Information Systems*.

M. E. Califf and R. J. Mooney. 1997. Relational Learning of Pattern-Match Rules for Information Extraction. In *Proceedings of the 1st Workshop on Computational Natural Language Learning (CoNLL-97)*.

M. Craven, D. DiPasquo, D. Freitag, A. McCallum, K. Nigam, T. Mitchell, and S. Slattery. Learning to construct knowledge bases from the world wide web *Artificial Intelligence*, 118:69–113.

C. Grover, S. McDonald, D. Nic Gearailt, V. Karkaletsis, D. Farmakiotou, G. Samaritakis, G. Petasis, M. Pazienza, M.Vindigni, F. Vichot and F. Wolinski. 2002. Multilingual XML-Based named entity recognition for e-retail domains. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*.

M. Kavalec and V. Svatek. 2002. Information extraction and ontology learning guided by web directory. In *Proceedings 15th European Conference on Artificial Intelligence*.

A. Maedche, G. Neumann, and S. Staab. 2002. Bootstrapping an ontology-based information extractions system. In P. S. Szczepaniak, J. Segovia, J. Kacprzyk, and L. A. Zadeh (eds), *Intelligent Exploration of the Web*.

M. T. Pazienza, A. Stellato, M. Vindigni, A. Valarakos, and V. Karkaletsis. 2003. Ontology integration in a multilingual e-retail system To appear in *Proceedings of the Human Computer Interaction International (HCII'2003)*.

T. Poibeau and D. Dutoit. 2002. Generating extraction patterns from large semantic networks and an untagged corpus. In *Proceedings of the 19th International Conference on Computational Linguistics*.

C. Roux, D. Proux, F. Rechenmann, and L. Julliard. An ontology enrichment method for a pragmatic information extraction system gathering data on genetic interactions. In *Proceedings of the ECAI 2000 Workshop on Ontology Learning*.

S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. 1995. Issues in inductive learning of domain-specific text extraction rules. In *Proceedings of the Workshop on New Approaches to Learning for Natural Language Processing*.

K. Stamatakis, V. Karkaletsis, G. Paliouras, J. Horlock, C. Grover, J. Curran, and S. Dingare. 2003. Domain-specific web site identification: The CROSSMARC focused web crawler. To appear in *Proceedings of the Second International Workshop on Web Document Analysis*.

M. Vargas-Vera, J. Domingue, Y. Kalfoglou, E. Motta, and S. Shum. 2001. Template-driven information extraction for populating ontologies. In *Proceedings IJCAI 2001 workshop on Ontologies Learning*.