Building a XML-based Corporate Memory[1]

Auguste Rabarijaona, Rose Dieng , Olivier Corby
INRIA, ACACIA Project, 2004 Route des Lucioles, BP 93,
06902 Sophia-Antipolis Cedex, France
E-mail:{Rose.Dieng, Olivier.Corby}@sophia.inria.fr,
Tel: 33 - 4 92 38 48 10 or 33 - 4 92 38 78 71, Fax: 33 - 4 92 38 77 83

Abstract. This paper emphasizes the interest of XML meta-language for corporate knowledge management. Taking into account the advantages of the World Wide Web and of ontologies for knowledge management, we present OSIRIX, a tool enabling enterprise-ontology- guided search in XML documents that may consitute a part of a corporate memory.

Keywords: XML, World Wide Web, knowledge management, document-based corporate memory, enterprise ontology, information retrieval.

1 Introduction

Extending the definitions proposed by [28] [20], we consider a corporate memory as an *«explicit, disembodied, persistent representation of knowledge and information in an organization, in order to facilitate its access and reuse by members of the organization, for their tasks»*. We consider its building as relying on the following steps [11]: (1) Detection of needs in corporate memory, (2) Construction of the corporate memory, (3) Diffusion of the corporate memory, (4) Use of the corporate memory, (5) Evaluation of the corporate memory, (6) Maintenance and evolution of the corporate memory. As emphasized in [11], several techniques can be adopted for the building of the corporate memory: it may be non computational, database-based, document-based, knowledge-based, case-based, Web-based...

Several kinds of documents can be exploited in a document-based corporate memory [2]:

- documents linked to projects: specifications of the product to be designed or manufactured, design documents, test documents, contractual technical reports,
- reference bibles in a given profession,
- visual documents such as photos, scanned plans, iconographic documents,
- technical reports, scientific or technical articles,
- books, theses, norms, archive documents, guides, dossiers of technological intelligence,
- on-line documentation, user manuals, reference manuals, business dossiers, etc.

Our work is situated in the context of knowledge management for an enterprise (an actual company or a virtual enterprise made of distributed organizations and teams of people that meet and work together online). We focus on a *document-based corporate memory, distributed through the Web.* The main problems to be solved are then: how to describe the documents that

1.This paper is an extension of a paper to be published at EKAW'99 [22].

will be part of this corporate memory ? How to improve their retrieval on the Web, in particular thanks to knowledge models?

After showing the interest of the Web and of ontologies for knowledge management, we will present the *interest of XML meta-language for corporate knowledge management*. Then we will describe OSIRIX, a system offering *enterprise ontology- guided search in XML documents*.

2 Exploitation of Internet/Intranet for Knowledge Management

2.1 Corporate knowledge diffusion

Individuals and organizations can take advantage of the remarkable possibilities of access to data, to information and to knowledge provided by Internet. Knowledge diffusion in a company can for example exploit the possible access to Internet or to an Intranet inside the enterprise.

Internet and the Web (HTTP, HTML) technologies have dramatically changed information management in the enterprise. The Web can serve as a basis for information distribution in a uniform way, independent of the way the information is stored. Diffusion can rely on a knowledge server on the Web or on publication on the Web [12] [8]. Different kinds of elements can be accessed through Internet/Intranet: documents (classic electronic documents, HTML documents...), databases, ontologies, knowledge bases, case bases, articles of digital journals, etc. Therefore several kinds of knowledge servers can be thought out: document servers, ontology servers, knowledge base servers, database servers, journal servers or digital libraries. The main problems to be solved are (1) retrieval of elements of the corporate memory in answer to a request and (2) adaptation of the answer to the user.

A corporate memory may not be restricted to the sole enterprise: an *internal corporate memory* can rely on an internal competence map inside the company while an *external corporate memory* rather includes information and knowledge stemming from the external world but useful for the enterprise work. Therefore, the retrieval and integration of information explicitly put on the Web by other companies working in the same area may be interesting for an external corporate memory. The Intranet of the enterprise can be exploited for construction and diffusion of an internal corporate memory, while an external memory can rely on (a) either an Extranet connecting the company and some privileged partners such as customers, suppliers, subcontractors, etc, or on (b) Internet and the Web in the case of «technological intelligence» purposes. As noticed by [23] different kinds of «intelligence» are interesting for a company: (a) technological intelligence in order to follow an existing or an emerging technology, (b) competitive intelligence in order to know about activities, products or services of the enterprise competitors or of other actors of the enterprise market (distributors, suppliers, customers...).

2.2 Information retrieval on the Web

The corporate memory is supposed to be used by adequate members of the enterprise: in all cases (database-based, document-based, knowledge-based, case-based, or Web-based corporate memory), we must notice the importance of information search, if possible adapted to the users' needs, their activities and their work environment. The problems to be tackled are: How can the user express his/her requests? How to improve hypertext navigation by the user? How to retrieve elements of the corporate memory in answer to a request? What additional meta-information (such as enterprise models, knowledge models, user models) could help to filter the information to be retrieved? Are inference capabilities needed in this purpose?

For information search on the Web, in addition to traditional search engines such as Altavista, other tools were developed: meta-engines and intelligent agents [23] [24] [20]. Several types of agents for the Web can be distinguished, according to their functions:

- *search meta-tools*, functioning by meta-index and launching in parallel search on several search engines to compile their results. But if such tools obtain results quickly, they suffer from the impossibility to parameterize their functions completely and from the poverty of their answers.
- *«intelligent agents» for the Web*: such software can be parameterized by the type of request, the number of consulted search engines, the depth of the links, the form of the results, the number of backed up pages, the backup of links, the automatic generation of bookmarks... Their operation rely on the following stages: (1) Parameter setting of the request, (2) Choice of the search engines, (3) Launching of search, (4) Automatic recording of the results, (5) Elimination of the identical pages, (6) Generalization of a summary of the results, and updating of possible search to detect the new pages. The advantages of these agents are the parameterization, the backup of the found pages, the link conservation and the search history. The disadvantages are their cost in time and in resources.
- *complex intelligent agents*, that, in addition to the previous functions, carry out information filtering, learning and intelligent management of the results. The advantage is that they obtain more relevant answers, the disadvantage is their cost in resources.

[24] compares several Web search tools according to several criteria (search, indexing, filtering, presentation, distribution and decision-making support).

Even though all such tools are generally not explicitly presented as aiming at knowledge management, they may be useful in the framework of information retrieval in a Web-based corporate memory (either an internal corporate memory or an external corporate memory). Moreover, guiding the searching on the Web by thesaurus [17] or ontologies [13] or expertise models [8] should have promising applications in corporate knowledge management.

3 Ontologies for Knowledge Management

3.1 Ontologies useful for knowledge management

Examples of ontologies that could be useful for knowledge management are *domain ontologies* (for example, a Vehicle ontology in automobile industry) and *enterprise ontologies* allowing to describe the model of a given company. The corporate memory of a company can for example include both a domain ontology and an explicit model of the enterprise.

Examples of enterprise ontologies are: (a) the Enterprise Ontology proposed by AIAI and consisting of five topics: activity, organization, strategy, time, marketing [27]; (b) the TOVE (Toronto Virtual Enterprise) ontologies proposed in the framework of the TOVE project [26] and relying on: activities, state, time, organization, resource, product, service.

3.2 Ontologies guiding information search on the Web

Another interest of ontologies is their exploitation for guiding search on the Web. Among the research projects aimed at *intelligently guiding information search on the Web thanks to ontologies*, let us quote:

- Ontobroker [13] offers an extension of HTML language to annotate documents with information relating to a particular ontology, in order to guide search by this ontology. Ontobroker offers, via the Web, information search guided by a particular ontology [13]. In the current version [3], the authors took as example an ontology on knowledge acquisition community. Ontobroker offers formalisms for the formulation of requests, and enables representation of ontologies and ontological annotation of HTML documents. It consists of three elements: a request interface, a search engine (Ontocrawler), an inference engine.
- SHOE (Simple HTML Ontology Extension) [18] is a tool of search for information on the Web, relying on an extension of HTML grammar so as to allow creation and use of ontologies. SHOE aims at the semantic organization of the documents in addition to their simple textual arrangement. It has an inference engine.
- Marri [29] is a multi-agents system exploiting ontologies and natural language analysis for information search over the Web.
- WebCokace [8] is a server of CommonKADS models of expertise. Starting from a user's request, it delivers models of expertise and associated documents, over the Web. WebCokace offers hypertext navigation within a model of expertise, information search within an expertise model through a query language, navigation between the expertise models and the documents: the entities of an expertise model are annotated by links towards documents, and the documents are themselves annotated by links towards the expertise model entities. Thus the search for information on the level of the documents is guided by a knowledge model.

Let us now focus on the exploitation of information search on the Web in the framework of a document-based corporate memory distributed through the Web or through an Intranet.

4 XML and Knowledge Management

For the Web documents, it would seem natural to rely on HTML as it is the most popular language for Web documents. But HTML has some drawbacks and a new standard, XML, was designed by World Wide Web Consortium (W3C) in order to obviate such drawbacks. Therefore, we rather suggest to rely on XML for a document-based, Web-based corporate memory.
After analysing the drawbacks of HTML and introducing XML, we will detail the features of XML interesting for knowledge management.

4.1 From HTML to XML

HTML suffers from a major drawback: information looses its structure when translated into HTML. It is very difficult, when not impossible, to process information embodied in HTML. This is because *HTML is used as a presentation-oriented markup language*.

Information and knowledge servers are overloaded because they have to search information and to perform format processing. Furthermore, the servers often answer many times to the same request if several views on the same data are requested.
[5] summarizes several weak points of HTML: (a) HTML does not allow the users to create their

own tags or attributes in order to parameterize or semantically qualify their data (lack of extensibility of HTML); (b) HTML does not support the specification of deep structures needed to represent database schemas or object-oriented hierarchies (lack of structure of HTML); (c) HTML does not support language specification that allows applications to check the structural validity of imported data (lack of validation of HTML).

In order to obviate these drawbacks, a working group of W3C created XML (eXtensible Markup Language) intended to be a standard for creation of markup languages [14]. XML has been designed for *distributing structured documents over the Web*. It is a kind of light SGML, simplified to meet Web requirements.

The W3C working group is also developing a common standard for linking in the markup languages developed with XML. Later on, they will develop a common standard for specifying the layout of documents encoded in these languages [14].

XML metalanguage derives from SGML (Standard General Markup Language). The specification of XML can be found in [6]. The notion of well-formed documents is defined. A DTD (Document Type Definition) is a syntactic specification being used as model for XML documents. A document is considered as valid if it respects the DTD with which it is associated. In the DTD, elements and their attached attributes are defined.

Contrarily to HTML, XML allows the users [5]: (a) to define their own tags and attributes; (b) to define data structures, and to nest document structures to any level of complexity; (c) to make applications allowing to test the structural validity of a document: any XML document can contain an optional description of its grammar for use by applications that need to perform structural validation; (d) to extract data from a XML document.

4.2 Advantages of XML for a corporate memory

As such, the new standard XML has some major advantages for corporate memory management, mixing SGML and Web advantages. Let us summarize such advantages:

Many Views on the Same Data. XML enables to manage information and knowledge in a unique structured way and enables several different processings. Knowledge servers retrieve information while clients are in charge of presenting it to users through adapted interfaces. It is then possible to take users and context into account and to present different views of the same data: it may be possible to generate graphic views, table of contents or to show the data themselves. Furthermore, data are loaded into the client (the browser) and can be processed locally: e.g. XML data can be processed by Java applets [5]. Hence, XML may represent for data what Java represents for programs: transparent portability through machines and operating systems.

Documents Built from Heterogenous Data. XML enables to manage structured documents and structured data in a uniform way. The XML format has been designed to enable document description as well as arbitrary data description. It is hence possible to mix data and documents in order to build *virtual documents issued from several sources*. Data may come from a technical data base while text may come from a document management environment. Furthermore, it is possible to annotate documents with modeled knowledge, so-called ontologies.

Standard for Information Exchange. In order to facilitate communication and information exchange, a community (i.e. a department, a company, a group of companies of the same domain, a company and its related providers and clients, etc) may define a standard domainoriented or application-oriented vocabulary by means of a DTD (Document Type Definition). A DTD is a syntactic specification being used as model for XML documents. A document is considered as valid if it respects the DTD with which it is associated. Documents or data can then be expressed with the defined XML markups and then be exchanged using these markups [5].

Document Formatting . XML has a companion formalism called XSL (Extensible Stylesheet Language), to define document-oriented presentation format. XSL may present a document in HTML, PDF, etc. It may also generate a generic format, that may be postprocessed to generate a standard output format. XSL also enables elementary document processing such as sorting, generating table of contents, tables, reorganizing the document structure. Using XSL, it is hence possible to define several output formats for the same document structure: XSL is a document transformation and formatting language. It is possible to write once and to publish many times, from the same source, to different media: digital and paper-based ones. This is very interesting for corporate memory management.

Hypertext. XML will also offer tools to build powerful hypertext documents by means of XLL (XML Linking Language), and XPointer, the language that enables navigation in documents according to their structure. XLL will implement the major hypertext functionality that can be found in dedicated tools: links between more than two documents, external links, links with semantics, etc. With external links, documents can be annotated from the outside, without modifying the source.

Information Search. XML facilitates information search because documents are structured and, hence, can be considered as a database. It is possible to rely on standardized markups to search information in a structured way. Moreover, the database community is currently integrating XML with database technology and search languages (cf. XML-QL [10]).

Annotation with metadata. In the spirit of RDF [16], Shoe [18] and OntoBroker [13], it is possible to annotate a document with metadata which describe in an abstract and synthetic way what the document is about, according to predefined standard ontologies.

Semantic markup. It is also possible to markup parts of the text document with tags the semantics of which is to establish a relationship with ontologies and knowledge models. In some cases, the markup can be interpreted as a hypertext link to ontologies. It can also be exploited by markup-driven search engines.

XML and Memory Management. XML as a structured document open standard may be a good candidate to facilitate migration to new systems or software through long time period. With XML, documents exist by themselves, independently of the processing tools. There are no more X or Y proprietary format documents but just (XML) documents.

Thanks to all these features, XML is an interesting meta-language for building a document-based corporate memory aimed at being accessible to the Intranet of a company.

5 Enterprise model - guided search in XML documents

Taking into account the interesting features of XML, we developed techniques for *enterprise-model-guided search in XML documents*.

5.1 Information search guided by knowledge models

5.1.1 Objectives

Our main objective is to perform information search in documents on the Web, guided by knowledge models. The result of search should include only relevant answers i.e. Web documents which «correspond semantically» to the request.
Instead of developing a specific extension of HTML to support a given ontology (as for example, Ontobroker or Shoe), we choose to handle XML documents, formatted with XML standard. The knowledge models that will guide the search will be CommonKADS expertise models, represented in standard CommonKADS Modelling Language (CML)[1].

The system OSIRIX (*Ontology-guided Search for Information Retrieval In XML-documents*) is integrated in the following global architecture:

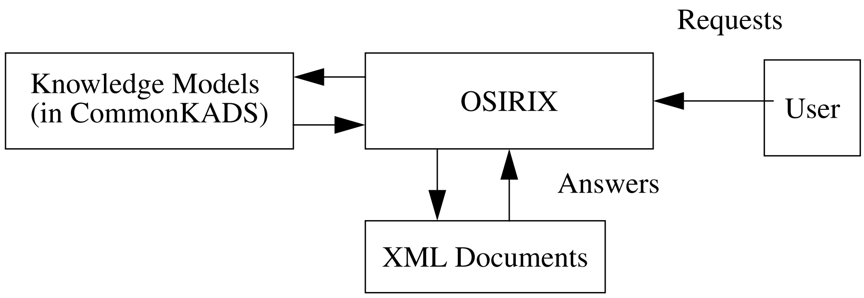

Fig. 1 : Global architecture.

5.1.2 Our approach

Two main phases are necessary: (1) the *creation of XML documents* which contain adequate structured, semantic information, so that these documents can be found later on as answers to requests. This document creation is performed by the author of the document or by the corporate memory builder; (2) and *search for information in the XML documents*. It is carried out by the OSIRIX system, after a request of the corporate memory user.

Creation of Documents. The documents must be annotated by ontological information in order to have a «semantical value» enabling their retrieval.

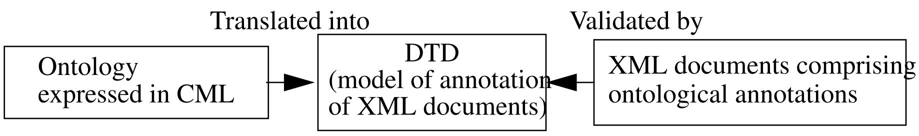

Fig. 1: Link between CML and XML

This ontological information can stem from an ontology developed by the company or impor-

1.CML is the language offered by the CommonKADS method in order to represent CommonKADS expertise models [25].

ted from external world, and upon which the company members agree. From this ontology, a DTD will be generated by our system OSIRIX: then the documents of the CM must respect this DTD that indicates the (optional) elements that can be used as ontological annotations in the documents. We could also require the company members to agree directly on a given DTD. But, as a DTD is rather difficult to read, we prefer to require the company members to agree on the ontology supposed represented in CML. Then, in order to enable to annotate semantically the documents by this ontology, the OSIRIX system generates automatically a DTD based on this ontology.

Remark: The ontologies can be built in the company, independently of their later use for XML document annotation: they are useful per se as components of a knowledge-based corporate memory.

Search for Information. In order to answer the user's requests, the system seeks in the ontological part of the documents, if an ontological answer is present there or not. The ontological filtering engine finds the documents answers among the candidate documents. If the system does not find exact answers, it can exploit the ontology to seek approximate answers: it can exploit the concept hierarchy to find an answer corresponding to subconcepts of the concept appearing in the initial request.

A scenario of information retrieval. As shown in figure 2, here is a possible scenario of information retrieval:

1. Initially, the user expresses his/her request in a browser such as Netscape or Explorer. There are two types of basic requests: (a) the basic request of general type, where the user constrains no attribute value: e.g. « Find the pages of all the projects of the company»; (b) the basic request of specific type, where the user constrains some attribute values: e.g. «Find the pages of the project named GENIE». In general, a hybrid query is made up of several basic requests: e.g. «Find all the reports written by any company for the project named GENIE»

2. Then, the request is transmitted to a classic search engine, accepting a logical AND among the keywords of the request. To focus search on only a few sites, we suppose that the user can specify a URL considered as the root of the URLs of the sites to be browsed. In the context of a company Intranet, it is quite realistic. In the case of a virtual enterprise, such a root URL will probably not exist. So, the user needs to indicate explicitly the sites of the organizations taking part in the virtual enterprise. Then, the search engine visits the specified sites, identifies the documents which follow a XML format, seeks the occurrences of the keywords used in the user's request. In the previous example, it will search for the keywords «project», «company» and «report» and find XML documents where all these keywords occur.

3. The request is decomposed into basic requests. The documents sent back by the search tool are processed by the ontological filtering engine of OSIRIX: it identifies the only documents which have the ontological information answering the request and returns them in the form of URLs in result. In the example, OSIRIX will give as answers the URLs of documents having ontological information such as:

```
<project>
  <name> GENIE</name>
```

```
    <report>
      <title>Description de modèles de coopération et gestion de conflits
           Rapport final du projet Genie, Thème 3, Lot L3.3.2.1 </title>
      <authors> Nada Matta, Olivier Corby</authors>
      <date>Juin 1996</date>
      <company>INRIA</company>
    </report>
  </project>
```

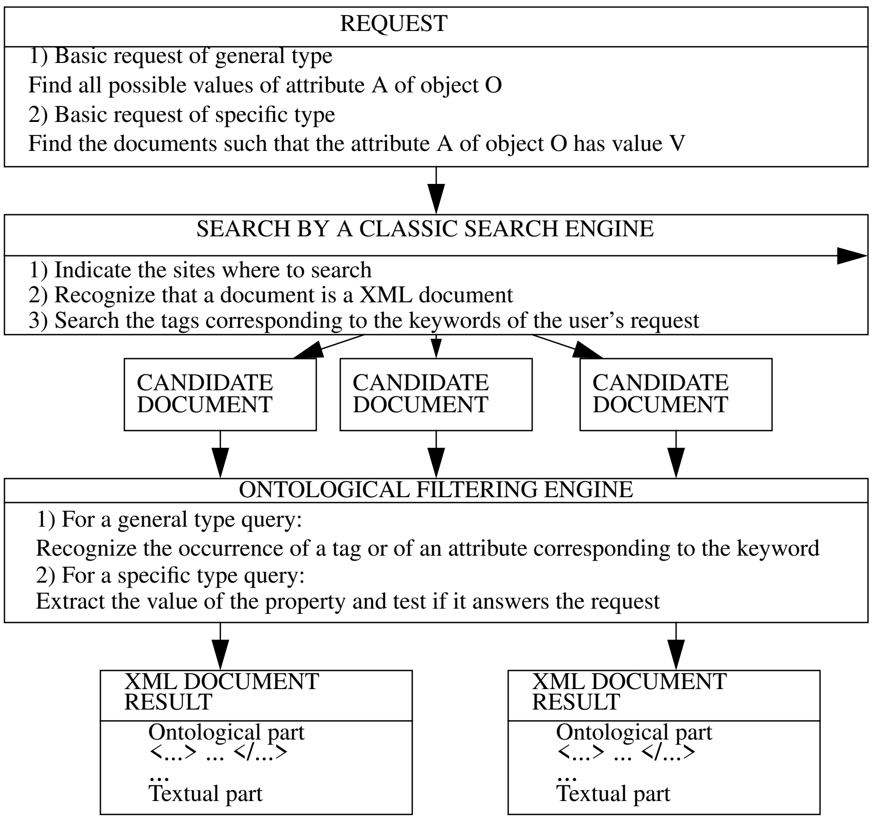

Fig. 2: Scenario of the information search in XML documents

Example: for the request «Find all the reports written by any company for the project named GENIE», OSIRIX will find documents having ontological information as in the previous example.

5.2 Implementation of the OSIRIX System

5.2.1 Translation Engine from CML into a DTD

COKACE [7] is an environment dedicated to CommonKADS Modelling Language (CML) [24]. COKACE is built upon the programming environment generator CENTAUR. When given the specification of the syntax and of the semantics of a programming language, CENTAUR generates a specific programming environment dedicated to this language. COKACE was generated by specifying the syntax and the semantics of the CML language in CENTAUR. COKACE is made of a set of tools enabling to build and verify CML programs, i.e. CommonKADS expertise

models. CENTAUR builds an abstract syntax tree for each parsed program of CML language. The parsed program is then manipulated by means of its abstract syntax tree. OSIRIX is built above COKACE. OSIRIX translator of a CML expertise model to a DTD is detailed in [21]. It is implemented using the tool PPML (Pretty Printer Minicomputer-Language) of CENTAUR generator [9], and a manager of object inheritance mechanism. In order to pretty-print abstract syntax trees in a readable format, the PPML formalism enables the programmer to specify pretty-printers. It allows to generate a textual representation starting from the abstract syntax tree associated to the CML program.
So, OSIRIX exploits PPML in order to generate, from the abstract syntax tree of a CML program, adequate XML elements constituting a DTD.

Here is a part of the translator of a CML concept into a part of DTD:

```
concept(*name, con_body(*descr, *super, *prop_list, *axioms)) ->
   [<v>
   [<h 1> «<!ELEMENT» *name «(» inhslotvrg(*name) *prop_list «)>»]
   [<h 1> «<!ATTLIST» *name «name_id» «ID #IMPLIED>»]
   def_child::*prop_list];
```

A CML concept and its properties are mapped into XML elements, as well as a CML relation. The translator can translate concepts, n-ary relations, the cardinality of a concept property, attributes and expressions.

For example, from the following concept of an ontology in CML:

```
concept report
   properties: title: universal
               authors: universal
               date: universal
               company: universal
end concept report
```

the following DTD will be generated automatically:

```
<!ELEMENT report (title?, authors?, date?, company?)>
<!ATTLIST report name_id ID #IMPLIED>
<!ELEMENT title(#PCDATA)>
<!ELEMENT authors (#PCDATA)>
<!ELEMENT date(#PCDATA)>
<!ELEMENT company (#PCDATA)>
```

Once the DTD obtained, the authors create their XML documents, by respecting the specifications of the DTD.

Remark: For the translation of the properties of CML concepts, we had the choice between XML attributes (ATTLIST) and XML elements (ELEMENT). We preferred to represent such properties of CML entities by XML elements, as an ATTLIST in XML is generally used to represent characteristics such as form, font, etc of the described elements but not to represent structured data.

5.2.2 Validation engine

The purpose of the validation engine is to check if the syntax specified in the DTD is well followed by the documents of the company [6].
Several « validation parsers» are offered on the Web. We chose the parser «XML for JAVA» of IBM [15], which is currently one of the most developed ones. This parser includes the API DOM (Document Object Model) of W3C which describes the interface of XML basic objects [1], and also SAX (Simple API for XML) interface [19].
The validation of a document allows the company to make sure that this document can constitute later on an answer. For OSIRIX filtering engine such a validation facilitates the work of data extraction.

5.2.3 Ontological filtering engine

The objective of the ontological filtering engine is to determine all the XML elements that are present in a given XML document, and to test the semantic presence of a concept (or any other entity) in the XML document.

Definition. We call « semantic presence» of an attribute (resp. concept) in a XML document, the fact that this attribute (resp. concept) appears in the XML document as a tag in the ontological part. This notation replaces the keywords or the <meta> tag that may occur in the HEAD part of a HTML document.
Ontological information can be regarded as meta-information and need not be visible through a browser.

The test on the two kinds of basic requests relies on this « semantic presence»:
- basic request of general type: the test consists of controlling whether the keyword (attribute, etc.) corresponding to the request is in the document or not, without taking into account its value.
- basic request of specific type: The test consists of checking whether the keyword corresponding to the request and its value indicated in the request are present « semantically» in the document or not.

Tools used. We used the SAX interface published on the Web [19]. SAX is an event-based application programming interface. Indeed, SAX sends back events to the application, each time that it meets an element, an attribute, a document, etc. The type of event depends on the type of the data encountered. As SAX requires to work with a parser, we chose the parser of IBM.

The kernel of the ontological filtering engine and examples are detailed in [21]. This engine can distinguish the documents which are really the answers among the candidate documents.

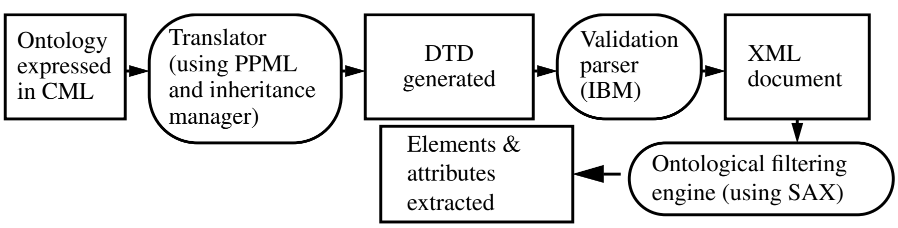

Fig. 3: Internal architecture of OSIRIX

Example :

For the query «Find all the reports written by any company in the project named GENIE», let us suppose that a XML document contains:

```
<project>
     <name> GENIE </name>
     <report>
        <title>Description de modèles de coopération et gestion de conflits
             Rapport final du projet Genie, Thème 3, Lot L3.3.2.1 </title>
        <authors> Nada Matta, Olivier Corby</authors>
        <date>Juin 1996</date>
        <company>INRIA</company>
     </report>
</project>
```

The functioning of the ontological fitering engine will be the following:

1.The engine exploits the events emitted by the SAX interface and tests if, among the elements contained in the XML document, there is <name> GENIE </name>: the answer is yes.

2.Then it tests the presence of the tags <report>, <company>: the answer is yes.

3.Then the URL of the XML document is sent back as answer.

Implementation. We used Cokace [8] to implement in CML an extension of the AIAI's enterprise ontology [27]. This ontology was then translated into a DTD (using PPML). We exploited the IBM validation parser «XML for JAVA», in order to validate the XML documents w.r.t. the DTD. We implemented the ontological filtering engine and a query interface. It remains to integrate the ontological filtering engine in a browser (once browsers for XML will be available).

6 Conclusions

This paper stressed the advantages of XML for corporate knowledge management. We discussed about the interest of the Web and of ontologies for knowledge management. We emphasized some interesting features of XML for knowledge management. Then we presented OSIRIX that offers enterprise ontology-guided search in XML documents.

Comparison with related work. Some aspects of this experiment can be compared with research on ontology-guided information search on the Web.

As WebCokace [8], OSIRIX relies on CommonKADS method and CML language: the extension of AIAI Enterprise Ontology was implemented in Cokace and the translator of CML to DTD was implemented using PPML. The exploitation of XML instead of HTML and a lack of exploitation of axioms are the main differences between OSIRIX and WebCokace, Ontobroker [13] or SHOE [10]. Compared to classic search engines, information search is still keyword-based in OSIRIX, but there, the keywords have a semantics. As a further work, we will exploit the CML axioms, and once XML browsers will be available, we will integrate OSIRIX in them.

Once the final version of RDF [16] officially recommended by the W3C, we will exploit the PPML formalism to build an automatic translator of a CML ontology into a RDF specification: we will use the same principle as for OSIRIX translator of CML into DTD.

Advantages of our approach. In spite of its drawbacks such as the lack of use of CML axioms, our approach seems interesting for the following reasons:

- *genericity:* a company can use OSIRIX with its own ontology, provided that it is described in CML. This ontology may be an enterprise ontology describing the organization of the enterprise, or a domain ontology describing the consensual knowledge handled in a given profession.
- *reliability:* the ontological filtering engine, based on the «semantic value» of the documents found as answers, allows to find more relevant documents than classic search engines.
- *flexibility:* the fact to have an automatic translator increases flexibility: as soon as the company ontology changes, the translator will be used again in order to obtain the new corresponding DTD. However, it will remain to ensure the consistency of previously annotated documents, with the new DTD generated from the modified ontology.
- *standardization:* the approach relies on the XML standard for the Web. And CommonKADS may be considered as one of the standards for ontology engineering.

Acknowledgements. We thank Rajae Ouaddari for completing the implementation of OSIRIX, started by Auguste Rabarijaona.

References

1. V. Apparao, S. Byrne, M. Champion, S. Isaacs, G. Nicol, J. Sorensen, R. Sutor, C. Wilson, L. Wood. Document Object Model Specification, World Wide Web Journal (http://www.w3j.com/)
2. J.-F. Ballay. Capitaliser et transmettre les savoir-faire de l'entreprise, Eyrolles, 1997.
3. V. R. Benjamins, D. Fensel and A. Gomez Perez, Knowledge Management through Ontologies. Proceedings of the Second International Conference on Practical Aspects of Knowledge Management, 29-30 October, 1998, Basel, Switzerland
3. P. Borras, D. Clément , T. Despeyroux, J. Incerpi, G. Kahn, B. Lang, and V. Pascual. CENTAUR: the system. In Proc. of the 3rd Symp. on Software Development Environment, ACM SIGSOFT'88, pp. 14-24, Boston, MA, November 1988.
5. J. Bosak. XML, Java, and the Future of the Web. March 1997. http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm
6. T. Bray, J. Paoli, C. M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0 W3C Recommendation. http://www.w3.org/TR/REC-xml
7. O. Corby, R. Dieng. Cokace: A Centaur-based environment for CommonKADS Conceptual Modelling Language Proc of the 12th European Conference on AI (ECAI'96), Budapest, Hungary, August 12-16, 1996, p. 418-422.
8. O. Corby, R. Dieng. A CommonKADS Expertise Model Web Server, Proc. of the 5th Int. Symposium on the Management of Industrial and Corporate Knowledge (ISMICK'97), Compiègne, (1997).
9. Projet Croap INRIA. The PPML Manual.. Manuel de référence du Pretty Printer Mini-language I et II.

10. A. Deutsch, M. Fernandez, D. Florescu, A. Levy, D. Suciu. XML-QL: A Query Language for XML. Submission to the World Wide Web Consortium, (1998).
11. R. Dieng, O. Corby, A. Giboin, M. Ribière. Methods and Tools for Corporate Knowledge Management. In B. Gaines, M. Musen eds, Proc of the 11th Workshop on Knowledge Acquisition, Modeling and Management (KAW'98), Banff, Alberta, Canada, (1998). To appear in International Journal of Human-Computer Studies, Special Issue on Organizational Memory and Knowledge Management, 1999.
12. J. Euzenat. Corporate Memory through Cooperative Creation of Knowledge Based Systems and Hyper-Documents. In Proc. of KAW'96, p. 36-1 – 36-20, Banff, Canada, November 1996.
13. D. Fensel, S. Decker, M. Erdmann, and R. Studer. Ontobroker: Or How to Enable Intelligent Access to the WWW. In B. Gaines, M. Musen eds, Proc of KAW'98, Banff, Canada, (1998).
14. L.M. Garshol. Introduction to XML. http://www.stud.ifi.uio.no/~larsga/download/xml/xml_eng.html
15. M. Hiroshi, T. Kent. Parser IBM XML for JAVA. http ://www.alphaworks.ibm.com/formula/xml. World Wide Web Journal
16. O. Lassila, R. R. Swick. Ressource Description Framework (RDF) Model and Syntax.http://www.w3.org/TR/1998/WD-rdf-syntax-19980216
17. C. Leloup. Moteurs d'indexation et de recherche. Environnement client-serveur, Internet et Intranet. Eyrolles, 1998.
18. S. Luke, L. Spector, D. Rager, J. Hendler. Ontology-based Web Agents. In Proc. of the First Int. Conference on Autonomous Agents, (1997).
19. Megginson Technologies Ltd. SAX 1.0 The Simple API for XML. http://www.megginson.com/SAX/
20. D. E. O'Leary. Enterprise Knowledge Management. Computer, 31(3):54-61, March 1998.
21. A. Rabarijaona. Aide à la recherche d'informations sur le Web guidée par des modèles de connaissances. DEA Report, INRIA-Sophia-Antipolis, (1998).
22. A. Rabarijaona, R. Dieng, O. Corby. XML and Corporate Knowledge Management. To appear in Proc. of the 11th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW'99), Dagstuhl, Allemagne, Springer Verlag, LNAI 1621, May 1999.
23. C. Revelli. Intelligence stratégique sur Internet. Dunod, 1998.
24. H. Samier, V. Sandoval . La recherche intelligente sur l'internet. Hermès, 1998.
25. Schreiber, G., Wielinga, B., Akkermans, H., van de Velde, W., Anjewierden, A. CML: The CommonKADS Conceptual Modelling Language. In L. Steels & al, eds, A Future for Knowl. Acqu.: Proc. of EKAW'94,Hoegaarden, Belgium, (1994) 1–25. Springer-Verlag, LNAI n. 867.
26. University of Toronto Canada. Enterprise Integration Laboratory - TOVE Ontologies. http://www.ie.utoronto.ca/EIL/tove/toveont.html
27. Uschold, M., King, M., Moralee, S., Zorgios, Y. The Enterprise Ontology. The Knowledge Engineering Review, Vol. 13, Special Issue on Putting Ontologies to Use (1996).
28. Van Heijst, G, Van der Spek, R., and Kruizinga, E. Organizing Corporate Memories. In B. Gaines, M. Musen eds, Proc. of KAW'96, Banff, Canada, (1996) 42-1 42-17.
29. F.-Y. Villemin (1998). Ontologies-based relevant information retrieval. In Wache et al eds, Proc. of ECAI'98 Workshop on Intelligent Information Integration, p. 55-66.