Exploitation of XML for Corporate Knowledge Management

Auguste Rabarijoana, 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 and presents an experiment of enterprise-ontology-guided search in XML documents constituting a part of a corporate memory.

1 Introduction

Extending the definition proposed by [14], we define a corporate memory (CM) as an *«explicit, disembodied, persistent representation of knowledge and information in an organization, in order to facilitate their access and reuse by members of the organization, for their tasks»*. Several techniques can be adopted for building the CM [6]: it may be non computational, database-based, document-based, knowledge-based, case-based, Web-based... The Web can serve as a basis for information and knowledge distribution in a uniform way. Ontologies can be exploited for guiding information search on the Web, as in Ontobroker [7], SHOE [10], and WebCokace [3].

Our work is situated in the context of a *document-based corporate memory, distributed through the Web.* After showing the interest of XML meta-language for corporate knowledge management, we will describe an experiment of enterprise ontology-guided search in XML documents.

2 XML and Knowledge Management

HTML, the most popular language for Web documents, has some drawbacks: lack of extensibility, of structure and of validation [1]. As HTML is used as a presentation-oriented markup language, it is very difficult to process information embodied in 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 [8]. XML has been designed for *distributing structured documents on the Web.* It is a kind of light SGML (Standard General Markup Language), simplified to meet Web requirements.

The specification of XML can be found in [2]. Contrarily to HTML, XML allows the users [1]: (a) to define their own tags and attributes; (b) to define data structures, and to nest document structures at any level of complexity; (c) to make applications allowing to test the structural validity of a document; (d) to extract data from a XML document. As such, the new standard XML has some major advantages for CM management, mixing SGML and Web advantages.

Many Views on the Same Data. XML enables to manage information and knowledge

D. Fensel and R. Studer (Eds.): EKAW'99, LNAI 1621, pp. 373-378, 1999.

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 [1]. 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 domain-oriented 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 [1].

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 CM 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 [5]).

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: XML documents exist by themselves, independently of processing tools.

3 Enterprise model - guided search in XML documents

Taking into account those interesting features of XML, we developed the system OSIRIX (*Ontology-guided Search for Information Retrieval In XML-documents*), based on techniques of *enterprise-model-guided search in XML documents*.

3.1 Information Search Guided by Knowledge Models

Our main objective is to perform information search in documents on the Web, guided by knowledge models. The result should include only relevant answers i.e. Web documents which «correspond semantically» to the request. Instead of developing a specific extension of HTML, we choose to handle XML documents. The knowledge models that will guide the search will be CommonKADS expertise models, represented in standard CML language [13]. Two main phases are necessary: (a) the *creation of XML documents* containing structured, semantic information, so that they can be found later on as answers to requests. This document creation is performed by the document author or by the CM builder; (b) and *search for information in the XML documents*. It is carried out by the OSIRIX system, after a request of the CM user.

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

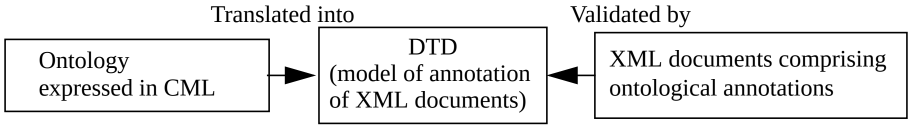
Translated into
Validated by
Ontology expressed in CML
DTD (model of annotation of XML documents)
XML documents comprising ontological annotations

Fig. 1: Link between CML and XML

This ontological information can stem from an ontology developed by the company or imported 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.

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 is shown in figure 2.

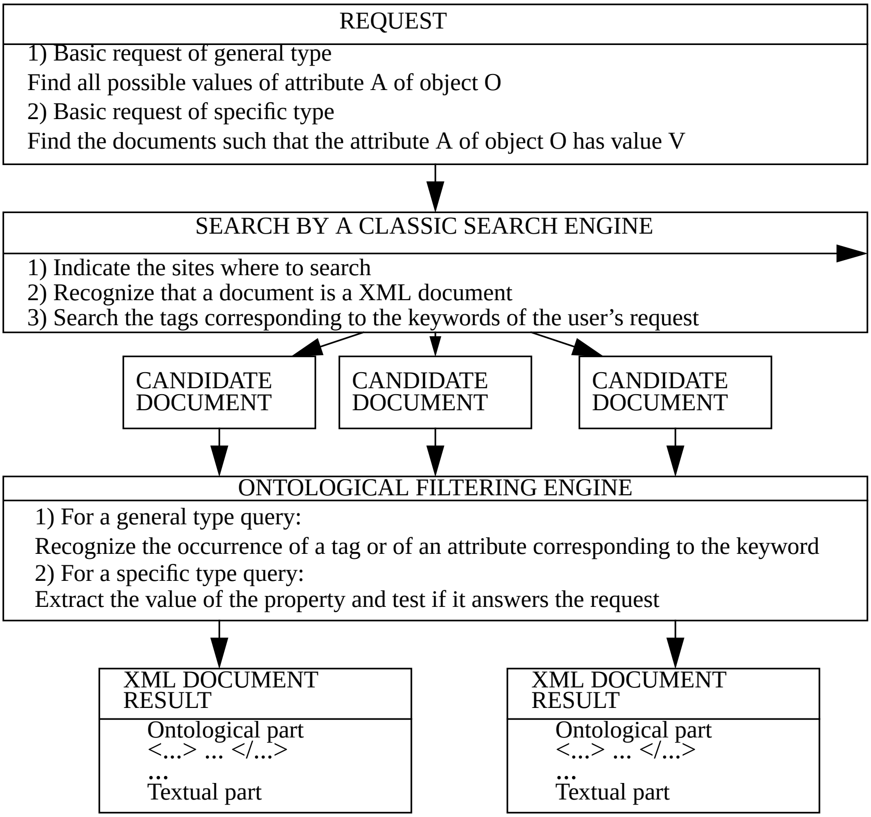
REQUEST
1) Basic request of general type
Find all possible values of attribute A of object O
2) Basic request of specific type
Find the documents such that the attribute A of object O has value V
SEARCH BY A CLASSIC SEARCH ENGINE
1) Indicate the sites where to search
2) Recognize that a document is a XML document
3) Search the tags corresponding to the keywords of the user's request
CANDIDATE DOCUMENT
CANDIDATE DOCUMENT
CANDIDATE DOCUMENT
ONTOLOGICAL FILTERING ENGINE
1) For a general type query:
Recognize the occurrence of a tag or of an attribute corresponding to the keyword
2) For a specific type query:
Extract the value of the property and test if it answers the request
XML DOCUMENT RESULT
Ontological part
<...> ... </...>
...
Textual part
XML DOCUMENT RESULT
Ontological part
<...> ... </...>
...
Textual part

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 such as :

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

3.2 Implementation of the OSIRIX System

Translation Engine from CML into a DTD. The translator of CML to DTD, detailed in [12], is implemented using the tool PPML (Pretty Printer Minicomputer-Language) of CENTAUR generator [4], and a manager of object inheritance mechanism. PPML allows to generate a textual representation starting from a tree of objects. Here is a part of the translator of the concepts in CML into 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];
```

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.

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 [2]. We chose the parser «XML for JAVA» of IBM [9]. The validation of a document allows the company to make sure that this document can later constitute an answer.

Ontological filtering engine. The ontological filtering engine [12] aims at determining 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. 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. 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».

We used SAX, an event-based application programming interface [11]: it 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 data encountered.

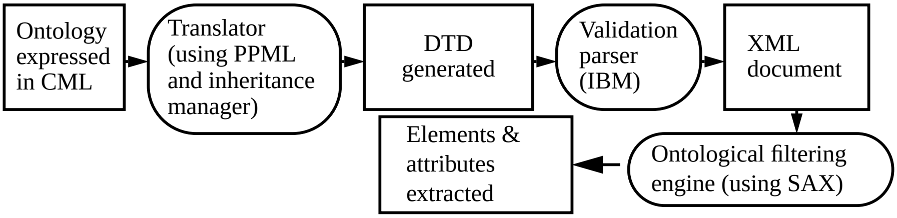
Ontology expressed in CML
Translator (using PPML and inheritance manager)
DTD generated
Validation parser (IBM)
XML document
Elements & attributes extracted
Ontological filtering engine (using SAX)

Fig. 3: Internal architecture of OSIRIX

Implementation. We used WebCokace [3] to implement in CML an extension of the AIAI›s enterprise ontology [14] that was 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. It remains to implement the query interface and to integrate the ontological filtering engine in a browser (once browsers for XML will be available).

4 Conclusions

The paper stressed the advantages of XML for corporate knowledge management and presented OSIRIX that offers enterprise ontology-guided search in XML documents. As WebCokace [3], it relies on CommonKADS method and CML language: the extension of AIAI Enterprise ontology was implemented in WebCokace and the translator 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 [7] 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, we will implement the request interface, and once XML browsers will be available, we will integrate OSIRIX in them.

References

1. Bosak, J. XML, Java, and the Future of the Web. March 1997. http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm
2. Bray, T. , Paoli, J., Sperberg-McQueen, C. M. Extensible Markup Language (XML) 1.0 W3C Recommendation. http://www.w3.org/TR/REC-xml
3. Corby, O., Dieng, R. A CommonKADS Expertise Model Web Server, Proc. of ISMICK'97, Compiègne, (1997).
4. Projet Croap INRIA. The PPML Manual.. Manuel de référence du Pretty Printer Mini-language I et II.
5. Deutsch, A., Fernandez, M., Florescu, D., Levy, A. Suciu, D. XML-QL: A Query Language for XML. Submission to the World Wide Web Consortium, (1998).
6. Dieng, R., Corby, O., Giboin, A., Ribière, M. Methods and Tools for Corporate Knowledge Management. Proc. of KAW'98, Banff, Alberta, Canada, (1998).
7. Fensel, D., Decker, S., Erdmann, M. and Studer, R. Ontobroker: Or How to Enable Intelligent Access to the WWW. In B. Gaines, M. Musen eds, Proc of KAW'98, Banff, Canada, (1998).
8. Garshol, L. M. Introduction to XML. http://www.stud.ifi.uio.no/~larsga/download/xml/xml_eng.html
9. Hiroshi, M., Kent, T. Parser IBM XML for JAVA. http ://www.alphaworks.ibm.com/formula/xml. World Wide Web Journal
10. Luke, S. , Spector, L., Rager, D., Hendler, J. Ontology-based Web Agents. In Proc. of the First Int. Conference on Autonomous Agents, (1997).
11. Megginson Technologies Ltd. SAX 1.0 The Simple API for XML. http://www.megginson.com/SAX/
12. Rabarijoana, A. Aide à la recherche d'informations sur le Web guidée par des modèles de connaissances. DEA Report, INRIA-Sophia-Antipolis, (1998).
13. 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.
14. 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).
15. 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.