AN ONTOLOGY-GUIDED ANNOTATION SYSTEM FOR TECHNOLOGY MONITORING

Tuan-Dung CAO[1 2]
[1] *CSTB, 290 route des Lucioles – B.P.209 06904 Sophia Antipolis FRANCE*
[2]*INRIA, ACACIA Team , 2004 route des Lucioles – B.P.93, 06902 Sophia Antipolis FRANCE*
Tuan-Dung.Cao@sophia.inria.fr

Rose DIENG-KUNTZ[2]
INRIA, ACACIA Team , 2004 route des Lucioles – B.P.93, 06902 Sophia Antipolis FRANCE
Rose.Dieng@sophia.inria.fr

Bruno FIÉS
CSTB, 290 route des Lucioles – B.P.209 06904 Sophia Antipolis FRANCE, Bruno.Fies@cstb.fr

ABSTRACT

Currently, in the field of technology monitoring, it is very important to be able to get relevant information from heterogeneous sources, especially on the World Wide Web. The coming of Semantic Web technologies promises intelligent retrieval and access to information through the use of semantic annotations based on ontologies. In a scenario of technology monitoring, agents are useful not only for handling semantic annotations to collect information, but also for automatic generation of these annotations from Web documents. In this article, we describe a new approach based on an ontology for building a multi-agent technology monitoring system.

KEYWORDS

Ontologies, Semantic web, technology monitoring, annotation generation, multi agent system

1. INTRODUCTION

Technological Watch or Technology Monitoring (TM) is now recognized as a crucial activity for achieving and maintaining competitive positions in a rapidly evolving business environment. It serves the purpose of identification and assessment of technological advances critical to the company's competitive position, and of detecting changes and discontinuities in existing technologies. The Web, considered as the hugest online information source, promises to be a mine of gold for TM. TM task is currently carried out by human actors, with the assistance of the traditional search tools, which sometimes do not give the expected results as they do not take into account the context and the semantic of information. The Semantic Web (Berners-Lee et al 2001), aims at defining and linking Web data in a way they can be understood and used by machines; it enhances intelligent information retrieval with semantic search based on semantic annotations and ontologies. In this paper, we present our approach based on semantic web to build an information system supporting TM process, relying on an ontology and a multi agent system (MAS). After explaining the TM process carried out at CSTB, we analyze the possible role of an ontology for the system; then, we present our algorithm using an ontology for searching and annotating documents on the WWW, before concluding.

2. TECHNOLOGY MONITORING PROCESS AT CSTB

Technology monitoring consists of monitoring the environment of an organization in order to discover the most recent technological and scientific knowledge, to collect and process all the relevant information,

likely to make the organisation flourishing, at short or long term. In this section, we propose a description of the TM task at CSTB (Centre scientifique et technique du bâtiment de France) by relying on the generic watch model proposed by Lesca (Lesca, 2002). This analysis will enable to identify the various phases where ontology or agents could intervene for improving the CSTB current TM process. But firstly, let us introduce the possible roles for the human actors involved in the monitoring task:

- *Observer (watcher):* person that carries out a thematic search on the Internet or on Intranet and sends to TM Users information evaluated as relevant.
- *Area referent (expert):* Technical monitoring actor responsible for an expertise area and who has a group of observers to manage.
- *TM User:* any employee from the company could be a TM user, i.e. a consumer of TM (manager, technician, engineer and researcher…).

The TM process at CSTB consists of the following phases:

1) Targeting (Expression of needs)

The objective of targeting phase is to answer the questions: How to collect relevant information? How to avoid to be overwhelmed with useless information? All the difficulty is precisely to know what is the relevant and useful information. Targeting means to express explicitly and clearly "WHAT" can be interesting in common for the various participants of the monitoring process. In this phase, the observers express their needs, define the topics of a given operation of monitoring, they also define criteria of precision on exhaustiveness, the spatial, temporal, technical cover of the search, the nature of expected information (bibliographical references, reports...), the intended use of the obtained information.

2) Information source detection and identification

In this phase, the watcher draws up a list of the sources to be questioned. They can be either sources already known by the watcher, or new sources found by classic keyword-based search engines.

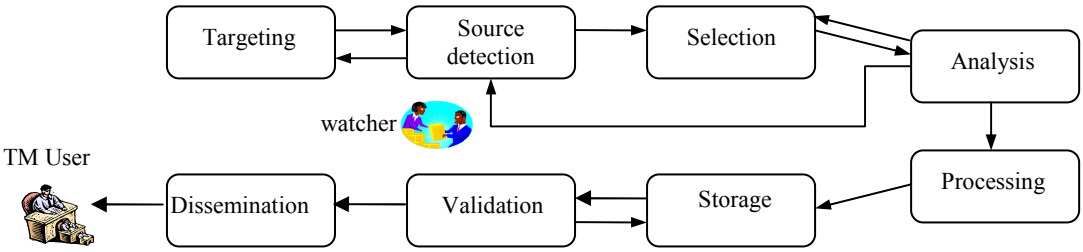


Figure 1: The TM process at CSTB

3) Information collecting and Selection

The documents are downloaded by crawlers or browsers. As the results obtained are likely to be too numerous, therefore the selection of the documents seeming relevant plays an important role to reduce the burden of the next phase: analysis.

4) Analysis

This phase consists of the evaluation of collected information. The actors concerned here are the watcher as well as the area referent (expert). If information is not sufficient to meet the needs, it is necessary to turn back to the second phase: information source detection.

5) Processing and storing

This phase consists of reading, synthesizing information in order to constitute the output of the TM: information cards, files serving as resources. In this stage, either observers or experts can create annotations on the documents. The results of this phase are stored in order to be available in case of demand from users.

6) Validation and Diffusion

It is necessary to have a partial transmission to the TM user who asked for the monitoring service. Only after his/her validation, a general dissemination to all the TM users can take place. For validation purposes, the watchers can send the monitoring results to some domain experts.

3. THE PROPOSED APPROACH AND SYSTEM

3.1 Approach overview

After studying the TM process, which is currently performed at CSTB, we found that the performance of two important sub-tasks could be improved with the support of an information system: searching documents and annotating documents. Based on Semantic Web technologies, our approach aims at building a system, which essentially retrieves information by semantic search as much as possible. To achieve that, Web resources that contain relevant information for the TM task must be described by one or several semantic annotations. So the main idea in our solution is: when a document is needed, the system will first try to make an intelligent search on the resources already annotated. For other Web resources relevant for this question but not yet annotated, it is necessary to discover them on the Web and then annotate them and store their annotations in an annotation base so that these annotations become available for later queries of users.

The figure 2 below shows the global vision of a searching and annotation system supporting the TM task.

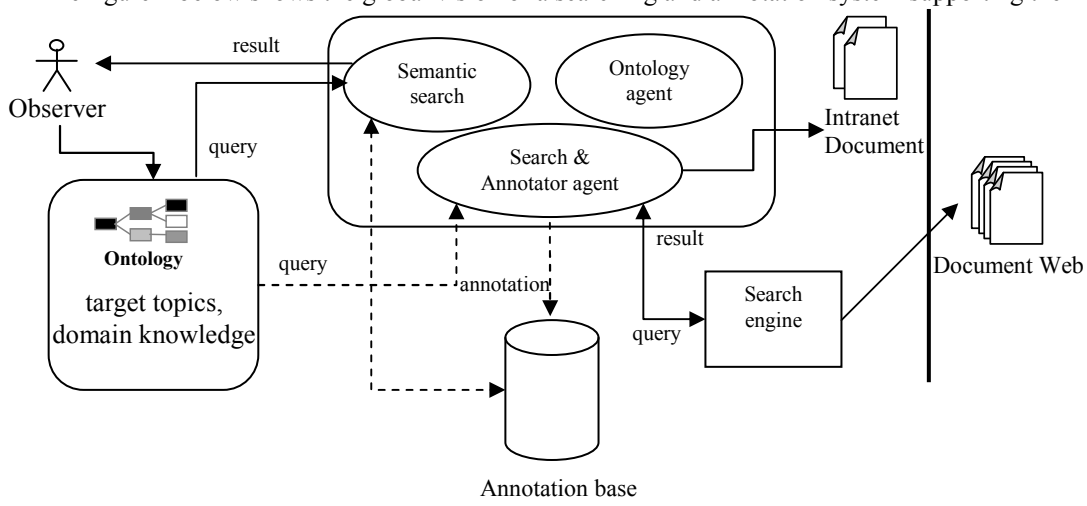


Figure 2: Ontology and MultiAgent System in the TM supporting system

We have chosen RDF/RDFS (Lassila 1999) as representation language for the semantic annotations and the ontology. For the semantic search in our system, we used an RDF-dedicated semantic search engine, CORESE (Corby et al 2002) developed by our team. Two problems remain to solve: search relevant documents concerning a topic and generate annotations about such documents. In the next section we present our approach to annotate a document on the Internet.

3.2 Annotation strategy

In the TM scenario at CSTB, we are interested in discovering new information or knowledge in the field of construction and building. Thus our system needs to be able to generate annotations enabling to know that a given document relates to a given topic and is useful for such user (group). Here are some specific cases where automatic annotation of a textual document is possible:

a) Annotations concern information related to the Dublin-Core like: the title, the authors, the document type and creation date, , etc

b) Annotations relate to concepts absent from the text (for example, for a document speaking about "Knowledge Management", the watcher can introduce an annotation specifying that this document is interesting for the engineers of Service SIA. The semantic search engine CORESE uses inference rules for generating automatically this type of annotations (Corby et al, 2002).

c) For the annotations the contents of which are extracted from texts, it is necessary to go from the simplest case to the most complicated case to recognize:

- significant words: words in bold, italics, between quotation marks,
- particular words which relate to the type of document, its nature…
- terms which correspond to concepts of the ontology,

The way to include them in the annotation will depend on the point of view of the annotator and will be predefined for automatic annotation generation.

If information to be extracted for the annotation is of structured or semi-structured type, the use of XSLT rules (by using the expression power of the access path language XPATH) for semi-automatic annotation generation (Cao et al, 2003) is effective and unexpensive.

4. ROLE OF ONTOLOGY FOR TECHNOLOGY MONITORING TASK

In our TM system, the role of ontology is significant since it forms the basis for syntactic and semantic metadata, which can be used for annotating about the resource content. These annotations can then be used by software agents to make intelligent search on the (external or internal) Web. This ontology must be built carefully so that the system can provide convincing results.

After analysing the CSTB TM process, we found the use of an ontology could be useful in several phases:
(a) In the targeting phase: using the concepts of the ontology can help to get rid of the ambiguity of the search context, enabling to specify the formulation of the query. Such an ambiguity can be related to linguistic phenomena like synonymy or homonymy. With an ontology describing the various sources of information, the system could launch specific agents dedicated for each source, to search for information.
(b) In the selection phase: Ontologies can be useful for associating the documents to specific topics.
(c) In the processing phase: Besides the main work where machines cannot replace humans (e.g. summary, synthesis of the document content, etc,), the potential role of ontologies in the document annotating task is undeniable. We focused on this direction.
(d) In the dissemination phase: According to the user's centre of interest, described by using the ontology, the system can automatically send to a given user suggestions for reading some documents which seem to be relevant to him/her.

4.1 Use of the ontology for document searching

To day's search engines do not typically know the exact context of searched entities. The solution for this problem is semantic search. But even in the case where the use of classic keyword-based search engine is inevitable, an ontology could be used to get better search results. In our system, to improve the results when the agents go out searching for new documents to annotate and integrate them into the annotated world managed by the system, the use of ontology is essential. Instead of keywords, concepts of the ontology will be selected to form a query. The advantage is the fact that we can use the child concepts of the initial concepts in the ontology to constitute the request, so it helps to reduce the relevant lacks in the search results obtained from a classic search engine. For example, when we search for documents which speak about "car accident", it is necessary to be able to find documents evoking not only "car accident" but also "truck accident", "bus explosion" "car pile-up", thus the query must be able to include all these alternatives more precise than "car accident". The problem is how to form the query and send it to search engine.

4.2 Extension of the O'CoMMA ontology for monitoring task

We reused the O'CoMMA ontology built in the CoMMA project (Gandon et al, 2002) since a part of this ontology relates to the field of building and construction. After analysing the CSTB's thesaurus, we noticed that O' CoMMA included most of the terms related to the building domain main fields. It thus remained to add the concepts more specialized in the field interesting for CSTB monitoring and concepts and relations dedicated for TM task: TM actors, monitoring phases, the information sources and some document types.

The ontology for modelling information sources is useful not only for adapting the agents to the source they will work on, but also for annotating the sources already known by the system. The annotations of an information source give additional information useful when annotating the documents which belong to these sources. Figure 3 shows the O'CoMMA ontology extension for TM task.

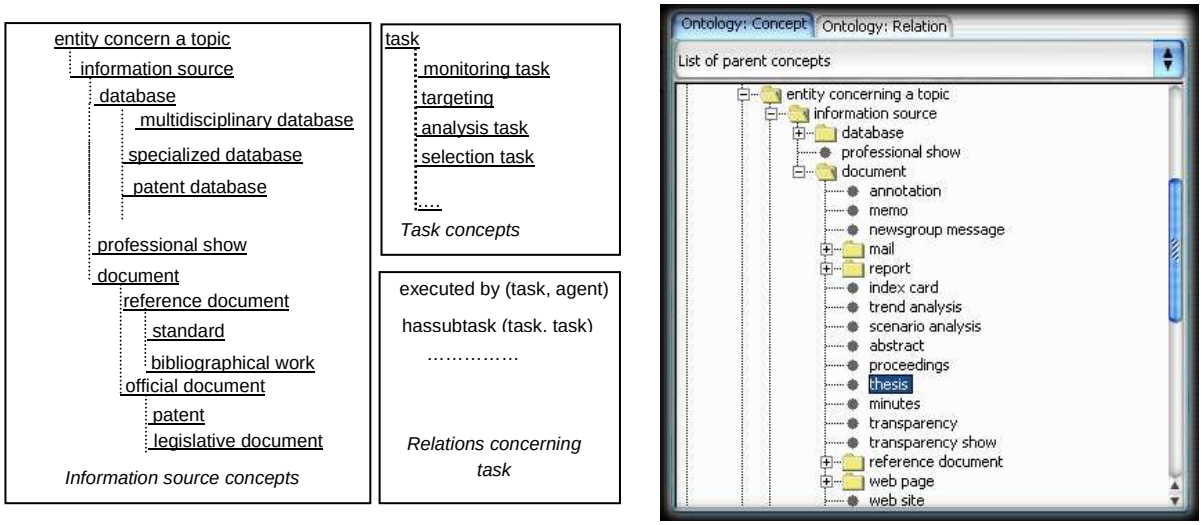


Figure 3: Extension of O'CoMMA for TM task

5. ALGORITHM USING AN ONTOLOGY FOR DOCUMENT SEARCH AND ANNOTATION

In this section, we present our first effort in the system development: an algorithm which allows to send to Google search engine a query formed by concepts of the ontology, and then automatically annotates documents found in the results returned. Google is currently considered as the most efficient search engine with more than two billion web pages indexed. Developers can use Google searching service in their application via a Google Web API. For each request, the result obtained from Google is a list of specific elements representing each document found. From such elements, we can get: the URL of document, the title, a text extracted from the document and showing the keywords of the query in the context where they appear in the document, and other information. So we can process this text to extract keywords found in the contents of the document.

Algorithm Description: Considering the user's query $Request_U$ sent to Google as a set of n concepts of the ontology, $Request_S$ is the query that the system will send to Google instead of $Request_U$. A branch of the ontology having C_i as root concept is a set of all successive concepts appearing in a path starting from the root concept C_i to any leaf concept descendant of C_i

Function *BasicSearchAnnotate* constitutes the query from a set of concepts generated by the main algorithm to send to Google. In each document in the results, the list of keywords found in the document is compared with the ontology in order to extract the deepest concepts in the ontology. These concepts and each corresponding original concept (in the user's request) are used to annotate the document in RDF format.

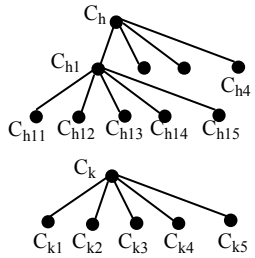
```
Algorithm BasicSearchAnnotate(Request, RequestU)
begin
      R= Combination of all labels of each concept in Request, with OR operator.
      Send R to Google
      for each element D in Google's results do
            K = GetAllKeyWords(D); Ann = ∅.
            for each concept w  in K do
                  if  w ∈ RequestU then add w to Ann , K= K \ {w}
                        else
                                Add to Ann all concepts w'∈ Descendants(w) ∩ K such that
                                not ∃  w'' ∈ Descendants(w') ∩ K
                        endif
            endfor
        Annotate the corresponding document D with all concepts in Ann and D.URL
        Return D and its annotations
        endfor
```

```
end
```

The main idea of the algorithm is the following: instead of using the concepts of the user's initial request, the system will generate a query $Request_S$ consisting of all concepts descendants of each initial concept and send this query to Google. But Google limits the number of keywords admitted in a request. Thus, if the number of concepts in the $Request_S$ is lower than this limit, the module BasicSearchAnnotate is sufficient for performing the search and generating annotations from the results of Google. In the opposite case, we studied three solutions to solve the problem of searching in Google with all the concepts in $Request_S$. To simplify the problem, we suppose that the depth of each branch having as root a concept belonging to $Request_U$ is lower than the Google limit. In the first solution, the algorithm uses all branches of the ontology starting from a concept in the user's request in order to form several queries sent to the Google search engine. After receiving all the annotations for each branch, the remaining work is to eliminate the redundancies, and aggregate all annotations of the same document. In the second and third solution, we make only the search for the first branch, then, for each document in the results obtained from Google, we search in the other branches of the ontology the concepts which are relevant to the document content. But, instead of browsing all the concepts remaining in ontology so as to compare them with the content of the document, in the third solution, we make an additional search in the web site containing the document to see if the same document is also found with another query (corresponding to another branch of ontology).

```
Algorithm OntologySearchAnnotation (RequestU, option)
input
RequestU : Set of concepts Ci in the user's request. Option: parameter for selecting among various
solutions.
begin
for each concept Ci in RequestU do
       Add all the concepts descendants of Ci to RequestS
endfor
if Cardinality(RequestS) < Google_keyword_limit then
       BasicSearchAnnotate(RequestS, RequestU)
else
    if option=1 then
       for each concept Ci in RequestU do
              Get S = set of all the branches having Ci as root
              for each branch Br in S do Call BasicSearchAnnotate(Br, RequestU)
       endfor
       Call RedundantScan(Annotation Base)
    else
       get Br = a branch the root of which is a concept chosen randomly C in RequestU
       Search Google with the Request = set of the concepts of Br
       for each element D in Google's results do
              Ann = {C}
              Get K = list of concepts found in document D.
              Add to Ann the deepest concept of Br belonging to K
              if option = 2 then
                     Let Words(D) = the list of the successive words appearing in the document D
                     for each Br'≠ Br in the set of remaining branches the root of which is a concept in
                     RequestU do
                            Add to Ann the deepest concept of Br' relevant to Words(D)
                     endfor
                     Annotate D with all concepts belonging to Ann. Return D with its annotations
              else
                     Get URLSite = URL address of the website containing D
                     for each Br'≠ Br in the set of remaining branches of which the root is a concept in
                     RequestU do
                            Search Google with the Request = Br' with the parameter website= URLSite
                            for each element D' in Google's results do
                                   if D.URL = D'.URL then
                                          Get K'= list of concepts found in document D'.
                                          Add to Ann the deepest concept of Br' belonging to K'.
                            endfor
                     endfor
                     Annotate D with all concepts belonging to Ann and return D with its annotations
              endif
       endfor
    endif
endif
end
```

Let us illustrate the algorithm by an example. Suppose that the user asks documents concerning two concepts of O'CoMMA: C_h: "fire detecting system" and C_k:"emergency lighting". All their descendants concepts are shown in figure 4:



C_{h1} : automatic fire detecting system
C_{h11}: flame-proof system
C_{h12} : parameterizable temperature detector
C_{h13} : ionization-type smoke detector
C_{h14} : line-type smoke detector
C_{h15} : optical smoke detector
C_{h2}: leak detecting system
C_{h3}: smoke detecting system
C_{h4}: indicating panel

C_{k1} : emergency lighting autonomous block
C_{k2}: autonomous feed box
C_{k3} : background light
C_{k4} : beacon light
C_{k5} : circulating light

Figure 4: Concepts in the user's query and their descendants that will be used by the algorithm.

The number of concepts descendants of C_h and C_k is 14 and exceeds the limit of Google (10). In the first solution, our algorithm will send to Google requests from all branches of ontology starting from C_h or C_k: (C_h, C_{h1}, C_{h11}), (C_h, C_{h1}, C_{h12}), …(C_h, C_{h1}, C_{h15}),….., (C_h, C_{h4}), (C_k, C_{k1})... (C_k, C_{k5}). In the second solution, it sends to Google only one branch for example (C_h,C_{h1},C_{h13}) and then for each document found D, all other branches are examined to see whether their concepts could be relevant to the document content. All these concepts found will be used to annotate automatically the document. For example if C_{k2} and C_{h15} are found relevant to D, so D will be annotated with (C_h,C_{h13},C_k,C_{k2},C_{h15}). In the third solution, for each document D found with a request corresponding to (C_h,C_{h1},C_{h13}) for example, the algorithm will search in the web site containing D, with queries corresponding to other branches: (C_h, C_{h1}, C_{h11}), (C_h, C_{h1}, C_{h12}), (C_h, C_{h1}, C_{h14}),….., (C_k, C_{k5}),... If with a branch like (C_k, C_{k4}), document D is once again in the search's result, D will be annotated with (C_h,C_{h13}, C_k,C_{k4}).

6. TOWARDS A MULTIAGENT SYSTEM SUPPORTING THE TM TASK

In the TM scenario, *the tasks to be performed on the distributed and heterogeneous information sources (in the Internet and Intranet of CSTB), the population of system users are distributed. The Multi-Agent System paradigm appears very well suited for the deployment of a software architecture above the distributed information* (Gandon et al, 2002). Moreover, the distribution of work between agents to implement our algorithm for searching and generating document annotations seems interesting, especially when each agent works on one branch of ontology. We identified four dedicated sub-societies of agents (cf. figure 5).

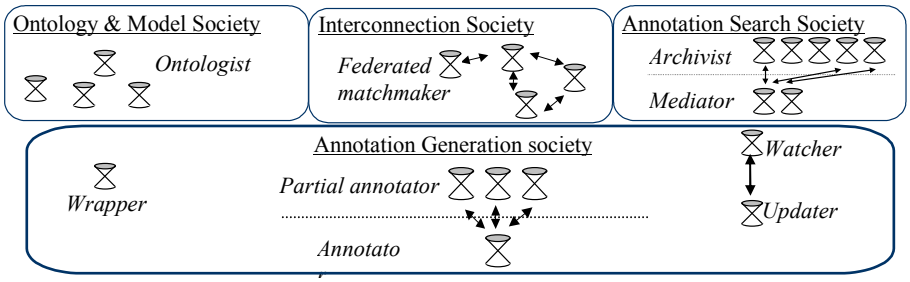


Figure 5: Agent Sub-Societies

- The agents of the *ontology-dedicated sub-society* are responsible for maintaining the ontology and for delivering information about the ontology to agents needing it.
- The agents of the *interconnection-dedicated sub-society* are in charge of the matchmaking of the other agents based upon their respective needs.
- The agents of the *annotation search-dedicated sub-society* are concerned with the use base of annotations of all documents known by the system, they use semantic search engine to retrieve reference matching user's request.
- Finally, the agents of *annotation generation-dedicated sub-society* are in charge of feeding the system with annotations of new resources, watching and updating annotations when an annotated resource changes. In this society, several agent roles are identified: *agent wrappers* that extract information from semi-structured sources, *agent annotators* that encapsulate the algorithm presented before, to search and annotate automatically documents from results of search engine.

CONCLUSIONS

This paper proposed a new approach to build information system supporting TM task, based on the use of an ontology and multi-agents technology. We discussed the role of ontology for intelligent search of information, especially in a TM scenario. An ontology dedicated to TM task has been built, and an algorithm using this ontology for searching and automatic annotating external documents was introduced. Currently, the first solution in the algorithm has been implemented.

The problem studied in this paper is concerned with automatically annotating free text documents. Currently, the most well known systems in this field are: MnM (Vargas et al 2002), Ontomat (Handschuh et al 2003), KIM Platform (Popov et al 2003), Pia-Core (Collier et al 2002), offering integration of ontologies and linguistic tools for adaptive information extraction. But their annotations rely only on named entities in the text, and belong to specific semantic types: person, organisation, address, date, money, number, etc. In the TM scenario, these types of annotations do not seem to be very useful. We rather try to annotate information such as: the themes and technical topics evoked in the document, Dublin Core metadata, information source in which the document was found, users for whom it is potentially relevant. Many projects share our idea to choose a multiagent information systems, especially in the domain of Semantic Web: CONACYT (Pérez-Coutiño et al 2003) developed a system that automates the authoring of document, CASMIR (Berney and Ferneley 1999), Ricochet (Bothorel and Thomas 1999), FRODO (Van Elst and Abecker 2002).

Currently we focus on improving our algorithm, implementing all the three solutions, and evaluating them on a real TM scenario of CSTB. Then, we plan to study the co-operation between agents in the annotation generation work and to implement annotator agents encapsulating our algorithm.

REFERENCES

Berners-Lee,T, Hendler, J and Lassila, O, 2001.The Semantic Web. *Scientific American*

Bothorel, C., Thomas, H., 1999. A Distributed Agent Based Platform for Internet User Communities, *PAAM'99.* p. 23-40.

Berney, B., Ferneley, E., 1999. CASMIR: Information Retrieval Based on Collaborative User Profiling. *PAAM'99.*

Cao, D. and Gandon, F., 2003. Integrating external sources in a corporate semantic web managed by a multi-agent system. *Proc. of Agent-Mediated Knowledge Management 2003.* USA, pp. 262-275.

Collier, T., Takeuchi, K., 2002. PIA-Core: Semantic Annotation through Example-based Learning. *Proceedings of the Third International Conference on Language Resources and Evaluation*. Las Palmas, Spain, pp. 1611-1614

Corby, O., Faron-Zucker, C., 2002. Corese: A Corporate Semantic Web Engine. *Workshop on Real World RDF and Semantic Web Applications 11th International World Wide Web Conference.* Hawaii

Van Elst, L., Abecker, A., 2002. Domain Ontology Agents in Distributed Organizational Memories. *Knowledge Management and Organizational Memories,* Dieng-Kuntz, R., Matta, N., (eds), Kluwer Ac. Publishers. p. 145-158.

Gandon, F., Dieng-Kuntz, R., Corby, O., Giboin, A., 2002, Semantic Web and Multi-Agents Approach to Corporate Memory Management, 17th IFIP World Computer Congress p. 103-115, August 25-30, 2002, Montréal, Canada

Handschuh, S., Staab, S., 2003. CREAM CREAting Metadata for the Semantic Web. *Int. J.. of Comp. & Tel. Networking*

Lassila, O., 1999, Resource Description Framework (RDF) Model and Syntax Specification. W3C Recomm. 22 Feb.1999

Lesca, H., 2002. Veille Stratégique - Concepts et méthode de mise en place dans l'entreprise. *Third International Conference on Language Resource and Evaluation.* Spain

Pérez-Coutiño, M., López-López, A. et al, 2003. A Multi-Agent System for Web Document Authoring, *Lectures Notes in Artificial Intelligence, Vol. 2663, Springer Verlag, 2003, ISSN: 0302-9743*, pp. 189-198

Popov, B., Kiryakov, A. et al, 2003. KIM – Semantic Annotation Platform. *ISWC' 2003.* Florida, USA, pp. 834-849.

Vargas, M., Motta, E. et al, 2002. MnM Ontology driven tool for semantic markup. *Proceedings of the workshop Semantic Authoring Annotation & Knowledge Markup (SAAKM 2002)*. Lyon, France.