Integrating external sources in a corporate semantic web managed by a multi-agent system

Tuan-Dung Cao, Fabien Gandon

INRIA, ACACIA Team
2004, route des Lucioles, B.P. 93,
06902 Sophia Antipolis, FRANCE
{Tuan-Dung.Cao | Fabien.Gandon}@sophia.inria.fr

Abstract

We first describe a multi-agent system managing a corporate memory in the form of a corporate semantic web. We then focus on a newly introduced society of agents in charge of wrapping external HTML documents that are relevant to the activities of the organization, by extracting semantic Web annotations using tailored XSLT templates.

Agents and corporate semantic webs

Organizations are entities living in a world with a past, a culture and inhabited by other actors; the pool of knowledge they mobilize for their activities is not bounded neither by their walls nor by their organizational structures: organizational memories may include or refer to resources external to the company (catalogs of norms, stock-markets quotations, digital libraries, *etc.*).

Our research team currently studies the materialization of a corporate memory as a corporate semantic web; this follows the general trend to deploy organizational information systems using internet and web technologies to build intranets and intrawebs (internal webs, corporate webs). The semantic intraweb that we shall consider here, comprises an ontology (O'CoMMA [Gandon, 2001]) encoded in RDFS, descriptions of the organizational reality encoded as RDF annotations about the groups (corporate model) and the persons (user profiles), and RDF annotations about the documentary resources.

The result of this approach is a heterogeneous and distributed information landscape, semantically annotated using the conceptual primitives provided by the ontology. To manage this corporate knowledge, it is interesting to rely on a software architecture that is itself heterogeneous and distributed, and the adequacy of multi-agent systems have been acknowledged in a range of projects proposing multi-agent systems addressing different aspects of knowledge management inside organizations. CASMIR [Berney and Ferneley, 1999] and Ricochet [Bothorel and Thomas, 1999] focus on the gathering of information and the adaptation of interactions to the user's preferences, learning interest to build communities and collaborative filtering inside an organization. KnowWeb [Dzbor *et al.*, 2000] relies on mobile agents to support dynamically changing networked environment and exploits a domain model to extract concepts describing documents and use them to answer queries. RICA [Aguirre *et al.*, 2000] maintains a shared taxonomy in which nodes are attached to documents and uses it to push suggestions to interface agents according to user profiles. FRODO [Van Elst and Abecker, 2002] is dedicated to building and maintaining distributed organizational memories with an emphasis on the management of domain ontologies. Finally, our team participated to the European project CoMMA [CoMMA, 2000], which aimed at implementing and testing a corporate memory management framework based on agent technology, for two application scenarios: (1) ease the integration of a new employee to an organization and (2) assist the technology monitoring activities. The system does not directly manage documents, but annotations about documents referenced by their URIs. CoMMA focused on three functionalities: (a) improve precision and recall, to retrieve documents using semantic annotations; (b) proactively push information using organization and user models; (c) archive newly submitted annotations. The architecture of CoMMA as it was demonstrated at the end of the project will be detailed in the following section.

By annotating resources available on the open Web, the organizational memory may span the organizational boundaries. In CoMMA, some agents were in charge of assisting the annotation and archiving process, but the annotation of a resource was essentially a manual process. While it was acceptable then, for the scenarios involved people whose role included the annotation task, it is clear that tools are needed to ease this work by assisting the exploitation of structural clues in resources, by saving the user from repeating similar and tedious manipulations, by automating updates when resources change, *etc.* It is for this reason that we introduced a new society in CoMMA, as described in the third section.

The initial architecture of CoMMA

The software architecture of CoMMA is a multi-agent system that was built and tested to manage a corporate memory based on the semantic Web technologies [Gandon, 2002a]; we briefly present this architecture here.

Societies and social functionalities

The architecture was fixed at design time, considering the functionalities CoMMA was focusing on and the software components that were to be included in the agents. It is clear to us that multi-agent systems are both a design paradigm and an enabling technology. We followed an organizational top-down analysis [Gandon, 2002b] where the MAS architecture was tackled, as in a human society, in terms of groups, roles and relationships, starting from the highest level of abstraction of the system (*i.e.* the society) and going down by successive refinements (*i.e.* nested sub-societies) to the point where the needed agent roles and interactions could be identified. Thus the system was first divided into four dedicated sub-societies of agents: three sub-societies dedicated to resources (ontology and corporate model; annotations; yellow pages needed for managing interconnection) and one dedicated to users.

Analyzing the resource-dedicated sub-societies we found that there was a recurrent set of possible organizations for these sub-societies: hierarchical, peer-to-peer or replication. Depending on the type of tasks to be performed, the size and complexity of the resources manipulated, a sub-society organization was preferred to another. The sub-society dedicated to the ontology and model was organized as a replication sub-society (*i.e.* an ontologist agent has a complete copy of the ontology). The annotations-dedicated sub-society was designed as a hierarchical organization. The yellow pages agents are provided by the JADE platform [Bellifemine *et al.*, 2001] used in CoMMA and are organized as a peer-to-peer society. Agents from the user-dedicated sub-society are not related to a resource type like the previous ones, thus they were studied separately. In analyzing and organizing these four societies, ten agent roles were identified and specified; we present them in the next section.

Agent roles

The user-dedicated sub-society comprises three roles:

- The *Interface Controller* (IC) manages and monitors the user interface; it makes the user looks like another agents to the whole MAS.
- The *User Profile Manager* (UPM) analyses the users' requests and feedback to learn from them and improve the reactions of the systems (esp. the result ranking).
- The *User Profile Archivist* (UPA) stores, retrieves and queries the user profiles when requested by other agents. It also compares new annotations and user profiles to detect new documents that are potentially interesting for a user and proactively push the information.

Precise querying on user profiles is handled by another agent type (AA) from the annotation-dedicated society.

CoMMA uses the JADE platform, thus the agents of the connection sub-society play two roles defined by FIPA[1]:

- The *Agent Management System* (AMS) that maintains white pages where agents register themselves and ask for addresses of other agents on the basis of their name.
- The *Directory Facilitator* (DF) that maintains yellow pages where agents register themselves and ask for addresses of other agents on the basis of a description of the services they can provide.

The society dedicated to ontology and model relies on:

- The *Ontology Archivist* (OA) that stores and retrieves the O'CoMMA ontology in RDFS.
- The *Enterprise Model Archivist* (EMA) that stores and retrieves the organizational model in RDF.

The annotation-dedicated society comprises two roles:

- The *Annotation Archivist* (AA) that stores and searches RDF annotations in a local repository it is associated to.
- The *Annotation Mediator* (AM) that distributes subtasks involved in query solving and annotation allocation and provides a subscription service for agents that whish to be notified of any newly submitted annotation.

Annotation management and limits

The annotation-dedicated society is in charge of handling annotations and queries in the distributed memory; the AM is in charge of handling annotations distributed over the AAs. The stake was to find mechanisms to decide where to store newly submitted annotations and how to distribute a query in order not to miss answers just because the needed information are split over several AAs [Gandon *et al.*, 2002]. To allocate a newly posted annotation, an AM broadcasts a call for proposal to the AAs. Each AA measures how close the annotation is, semantically, from the types of concepts and relations present in its archive. The closer AA wins the bid. Thus we defined a pseudo-distance using the ontology hierarchy and we used it to compare the bids of the different AAs following a contract-net protocol. The solving of a query may involve several annotation bases distributed over several AAs; the result is a merging of partial results. To determine if and when an AA should participate to the solving of a query, the AAs calculate the overlap between the list of types present in their base and the list of types used in the query. With these descriptions the AM is able to identify at each step of the query decomposition the AAs to be consulted. Once the AA and AM roles had been specified properly together with their interactions, modules of CORESE (a semantic search engine and API [Corby & Faron-Zucker 2002.]) have been integrated in the agent behaviors to provide the needed technical abilities.

[1] http://www.fipa.org/

The submissions of queries and annotations are generated by the IC in the user-dedicated society. Users are provided with graphical interfaces to guide them in the process of building semantic annotations and queries using concepts and relations from the ontology. Although we are aware that the multiple activities and actors of organizations require multiple ontologies, we focused on individual scenarios based on a single ontology that can be different from one scenario to another; the ontology is designed and tailored from the point of view adopted for the considered scenario. In other extensions of the systems we consider the problems of translating annotations and queries from one ontology to another, but this point is not addressed here. Therefore O'CoMMA provides a conceptual vocabulary that is relevant for the Knowledge Management scenarios envisaged in CoMMA and it only aims at being sufficient to annotate information resources with those aspects that are relevant for the considered scenarios (if they have such aspects). In other words, we do not aim at generating every possible annotation for a resource, but only the ones that are possible in the considered ontology and thus relevant in the considered application scenarios.

From an annotation built in the GUI, the Interface Controller agent generates an RDF annotation that will be archived. The task of annotating a resource can quickly become tedious and repetitive, and if a large set of relevant documents is discovered, the prospect of having to annotate each one of them may be off-putting and the hypothesis of manual annotation becomes unrealistic. However, some Web sites have a rather static structure which, even if it is implicit, provides structural clues (font style, table, separators, *etc.*) that can be exploited to automate some extraction rules that enable the user to automatically generate annotations from the content of the resource. It is for this special case that we introduced a new society of agents, providing this new service, as described in the following sections.

Introducing wrapping services

No organization is an island; it is included in a culture, a country, a society, a market, *etc.* and a lot of interesting information is available on the open Web, relevant to the organization's environment, core activities, domain, *etc.* Being relevant to the organization, these resources can be annotated to integrate the corporate semantic web. In the initial scenarios of CoMMA, the effort of annotation was part of the role of some stakeholders (e.g. technology monitors, mentors). However, as the available Web pages continually increase their volume and change their content, manual annotations becomes very hard; for this reason it is interesting to consider semi-automated tools in order to ease the annotation process. We see this as the dual problem of the usual vision of a corporate portal. A corporate portal usually offers services from the organization to the outside-web, it is the shop window of the organization; we call it the outer portal to the inner services. Conversely the organization resources can be used to filter and mine the outside 'world *wild* web' and provide its internal communities of interest with a portal enabling them to access carefully chosen, selected and validated external sources providing resources interesting for them; we call it the inner portal to the outside services.

A way to implement an inner portal in CoMMA is to introduce a society of wrappers as shown in bold in Figure 1. The agents of this society automate the extraction of relevant information and their integration to the organizational memory.

The open Web is human-intended and full of unstructured or semi-structured resources. The extraction of semantic annotations, to provide the organization with internal pointers to relevant resources outside, raises the problem of specific wrapper development for each relevant source. This led researchers to consider generic wrappers and parameterization scripts or customization workshop tools where one can quickly develop a new specialized wrapper using automatic rule generalization and other machine learning and text mining techniques. The following sub-sections present our approach to this problem.

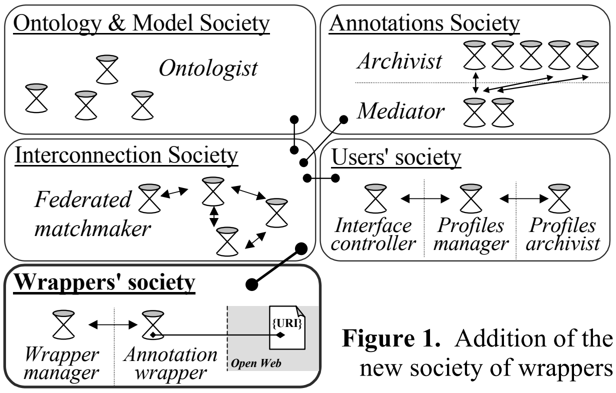


Figure 1. Addition of the new society of wrappers

Data extraction process

There are two options to implement the wrapper:

- *On-the-fly conversion* where the wrapper uses its skills to convert information whenever it is solicited for solving a query. This approach has the advantage of always providing the requester with up-to-date information but the conversion process may slow down the agent's answer.
- An *image generator* wrapper, triggering checks at chosen intervals and, if needed (*i.e.*, a new version of the monitored Web page is detected), applying conversion mechanisms to update its annotations. This approach has the advantage of providing information very quickly since the agents work on the structured image of the information, and to be available even if the outside Web site is off-line. However, depending on the monitoring settings, the data may not be up-to-date and moreover the amount of information duplicated in the intranet may be important.

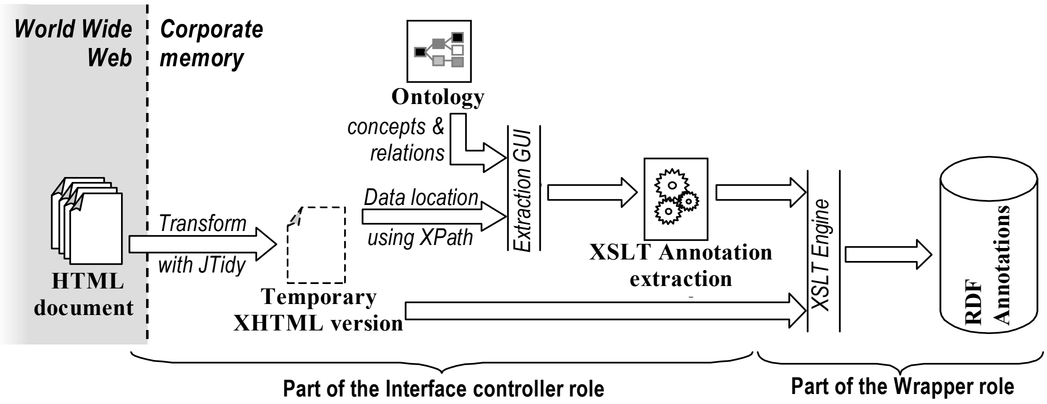


Figure 2. Extraction process

We chose the second option because it is faster when dealing with a query (since the structured data are always directly available when needed) and it decouples and isolates the intranet from the Internet, which is an appreciable feature both for security and availability.

For some Web sites such as online catalogs or digital libraries we observed that a major part of the information used for annotating the Web pages is usually present in the content of the document itself. Thus we designed a solution for automatically generating RDF annotations by extracting semi-structured data from the Web pages. The solution entirely relies on XML technologies (XHTML, XSLT, XML syntax of RDF) and is guided by an ontology in RDFS. The tool we developed is able to generate wrappers responsible for downloading documents from a Web site and extracting data from them using a mapping (XSL Templates) based on an ontology to generate the annotations to be added to the memory.

In [Bergamaschi and Beneventano, 1999], a mechanism is described for generating translators. The idea of facilitating the annotation generator design is indeed interesting since agents have to be customized for each new source. A library of useful functions or a toolkit for Web Wrappers can also be a valuable asset for feasibility and acceleration of the process of developing such agents. For example [Muslea *et al.*, 1999] describes an approach to wrapper induction, based on the idea of hierarchical information extraction where extraction rules are described as finite automata learned by the agent. Whereas projects like TSIMMIS [Garcia-Molina *et al.*, 1995] actually looked at the full integration of dynamically changing heterogeneous sources, we only consider here the generation of semantic annotations to assist the localization and access to information resources; the actual use of a given information resource is not supported here.

Our approach is focusing on well-structured Web sites; the information can be frequently updated, but the layout and organization of these pages must remain the same, or at least do not change too much. A lot of Web sites of this type exist on the Web; typical examples are online libraries of documents, weather forecast, stock exchange quotations, *etc.* As illustrated in Figure 2, we proceed in three steps:

Step 1: Retrieve the HTML source code and convert it into XHTML. The extraction template is built using a sample Web page chosen among the set of pages to be annotated and which structures are similar. We suppose here that these pages are located by URLs containing a running counter, so we used a structure called URLSet which contains the prefix and the postfix of the URL and a counter range, to identify this set of pages.

To extract data from a Web page, the system needs to access the appropriate portion of data in the source document. Much of the HTML content currently on the Web is ill-formed as a result of the leniency of HTML parsers used by Web browsers (missing closing tags, incorrect nesting of tags, *etc.*). Therefore, the system converts the HTML document into an XHTML document following their hierarchical structure and correcting errors thanks to the JTidy Package. After this step, we obtain a well-formed XML document using the Extensible Hypertext Markup Language (XHTML). Doing so, in the subsequent steps, we can use XML tools to manipulate the web page as a DOM (Document Object Model) and we represent it using the JDOM package.

Step 2: Generating extraction template based on the ontology O'CoMMA. The next step consists in describing the extraction process to provide the system with an annotation template using the vocabulary provided by the O'CoMMA ontology, which is itself encoded using the XML syntax of RDF(S). An annotation contains information extracted from the document and concepts and properties chosen in the ontology. We chose the Extensible Style Sheet Language Transformations (XSLT) as a format for data extraction rules in order to benefit from its advantages in manipulating XML document and to use XML Path expressions (XPath) to locate data to be extracted. To assure the preciseness and the automation for the extraction process, we created some built-in templates providing high level extraction functions such as: recursive extraction of a list of data delimited by a given separator (e.g. for the list of authors - see figure 3), or replacing some data extracted by a corresponding concept in the ontology (e.g. for keywords). These built-in templates are transparent to the users of system, and are embedded in the overall extraction template produced by the system. Thus the system assists the generation of extraction templates in XSLT that can be exchanged, reused, merged, compound and nested.

Step3: Applying the XSL Template to the HTML source codes to build an annotation base. Once the template is created, it is used by an XSLT processor to transform all Web pages into annotations of their content.

```
<xsl:template name="getListItem">
<xsl:param name="list" />
<xsl:param name="delimiter"/>
<xsl:param name="openning"/>
<xsl:param name="closing"/>
<xsl:choose>
 <xsl:when test ="$delimiter = 'br'">
  <xsl:for-each select="$list">
  <xsl:value-of select="$openning" disable-output-
                                        escaping="yes"/>
     <xsl:value-of select="normalize-space()" />
  <xsl:value-of select="$closing" disable-output-
                                        escaping="yes"/>
  </xsl:for-each>
 </xsl:when>
 <xsl:otherwise>
   <xsl:choose>
   <xsl:when test="string-length($list) = 0" />
   <xsl:otherwise>
       <xsl:choose>
          <xsl:when test="contains($list, $delimiter)">
          <xsl:value-of select="$openning"
                        disable-output-escaping="yes"/>
            <xsl:value-of select="concat(normalize-space
      (substring-before($list, $delimiter)),'&#10;')" />
          <xsl:value-of select="$closing"
                         disable-output-escaping="yes"/>
          </xsl:when>
          <xsl:otherwise>
         <xsl:value-of select="$openning"
                         disable-output-escaping="yes"/>
          <xsl:value-of select="concat(normalize-space
                                    ($list),'&#10;')" />
         <xsl:value-of select="$closing"
                         disable-output-escaping="yes"/>
          </xsl:otherwise>
       </xsl:choose>
    <xsl:call-template name="getListItem">
     <xsl:with-param name="list" select="substring-
                             after($list, $delimiter)" />
     <xsl:with-param name="delimiter"
                                   select="$delimiter" />
     <xsl:with-param name="openning"
                                    select="$openning" />
     <xsl:with-param name="closing" select="$closing" />
    </xsl:call-template>
   </xsl:otherwise>
  </xsl:choose>
 </xsl:otherwise>
</xsl:choose>
</xsl:template>
```

Figure 3. Recursive extraction of a list of data

Agent interaction scenario

The agent interaction scenario we envisaged to support the data extraction process is in 6 stages:

1 In the Interface Controller (IC), the users select a source of web pages having a similar structure, and annotate one of the pages, giving an example of annotation;

2 The IC derives an XSLT template to extract information from the Web page and automatically build the corresponding RDF annotation;

3 Once the template is validated, the IC contacts a Wrapper Manager (WM) and requires the creation of an Annotation Wrapper Archivist (AWA) to handle this new source of annotations.

4 The IC sends the template and the URLSet to the newly created AWA;

5 The AWA creates its base of annotations applying the template and then registers with an Annotation Mediator (AM) like any Annotation Archivist, so that it is ready to participate to query solving;

6 The AWA maintains its base, monitoring changes in the source.

This scenario allows users to develop and launch a population of AWA, each one of them monitoring an assigned source.

Modifications of the interface controller

To assist the two first steps of the scenario, we developed a specific graphic tool called WebAGe (Web Annotation Generator) that was inserted into the existing interface of the IC of CoMMA. Figure 4 shows the screenshot of the interface of WebAGe in CoMMA.

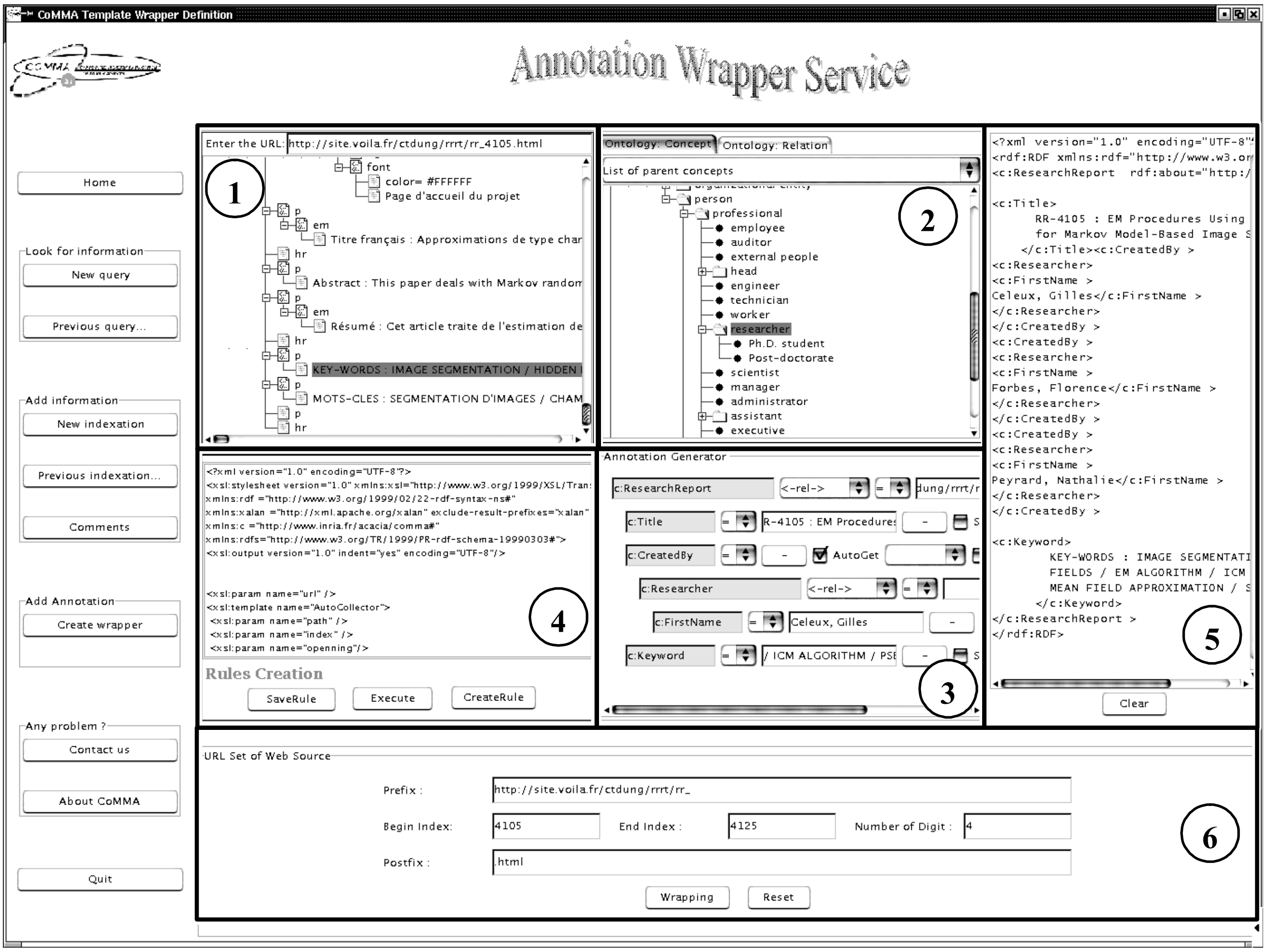
Users can specify the sample Web page in the box at the top of the widget ①. The Web page is then downloaded and converted into XHTML. The resulting hierarchical DOM structure of the document is visualized, allowing users to choose the data to be extracted and to specify the XPath just by manipulating the graphical representation.

In ② is an ontology navigation widget allowing users to browse the taxonomic structure of the ontology and choose concepts and relations to be used in the annotations.

The area ③ is the template definition widget for defining the extraction rule. The data targeted in the widget ① and concepts and relations chosen in widget ② are dragged and dropped here by the users to build the template.

The widget ④ shows the code of the template in XSLT, as it was automatically derived by the system. If wanted, a human expert can intervene here to improve the result of the wrapping process in case a special tuning is required.

Users can view and check the resulting annotation in area ⑤ and in area ⑥ they can require the creation of an AWA by specifying the set of Web pages to which this template has to be applied and by submitting it in a request together with the XSLT template.



Enter the URL: http://site.voila.fr/ctdung/rrrt/rr_4105.html

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Tran:
xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xalan ="http://xml.apache.org/xalan" exclude-result-prefixes="xalan"
xmlns:c ="http://www.inria.fr/acacia/comma#"
xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">
<xsl:output version="1.0" indent="yes" encoding="UTF-8"/>


<xsl:param name="url" />
<xsl:template name="AutoCollector">
 <xsl:param name="path" />
 <xsl:param name="index" />
 <xsl:param name="openning"/>
```

c:ResearchReport	<-rel->	=	dung/rrrt/r	
c:Title	=	R-4105 : EM Procedures	-	S
c:CreatedBy	=	-	☑ AutoGet	
c:Researcher	<-rel->	=		
c:FirstName	=	Celeux, Gilles	-	
c:Keyword	=	/ ICM ALGORITHM / PSE	-	S

```
<?xml version="1.0" encoding="UTF-8"
<rdf:RDF xmlns:rdf="http://www.w3.or
<c:ResearchReport  rdf:about="http:/

<c:Title>
      RR-4105 : EM Procedures Using
      for Markov Model-Based Image S
    </c:Title><c:CreatedBy >
<c:Researcher>
<c:FirstName >
Celeux, Gilles</c:FirstName >
</c:Researcher>
</c:CreatedBy >
<c:CreatedBy >
<c:Researcher>
<c:FirstName >
Forbes, Florence</c:FirstName >
</c:Researcher>
</c:CreatedBy >
<c:CreatedBy >
<c:Researcher>
<c:FirstName >
Peyrard, Nathalie</c:FirstName >
</c:Researcher>
</c:CreatedBy >

<c:Keyword>
        KEY-WORDS : IMAGE SEGMENTATI
        FIELDS / EM ALGORITHM / ICM
        MEAN FIELD APPROXIMATION / S
      </c:Keyword>
</c:ResearchReport >
</rdf:RDF>
```

Prefix : http://site.voila.fr/ctdung/rrrt/rr_

Begin Index: 4105 End Index : 4125 Number of Digit : 4

Postfix : .html

Figure 4. Interface to generate annotation extraction templates

Hierarchical society of wrappers

To operationalise the extraction process, we implemented a prototype of the wrapper society. It is a hierarchical society, with a Wrapper Manager role (WM) in charge of creating and managing the population of wrappers and an Annotation Wrapper Archivist role (AWA) in charge of wrapping a targeted source and making the obtained annotation available for query solving.

Wrapper manager (WM). The WM manages the AWAs. Currently its task is to create an AWA when it receives requests from the IC. It does so by sending a request to the AMS to create a new agent of the AWA class. In the future, we envisage to add new features to the WM such as recreating an AWA if it died, managing the addresses of the AWAs active in the system, *etc.* Table 1 provides the role card of the WM.

Wrapper Manager (WM)	
Role Model	Part of the wrapper-dedicated society.
Responsibilities	Archivist management : create, recreate, monitor and destroy Annotation Wrapper Archivists
Collaborators	IC, AMS, AWA, DF
External Interfaces	-
Expertise	FIPA-Agent-Management Ontology CoMMA-Ontology.
Coordination and Negotiation	Initiates *FIPA-Request* protocol to register and deregister itself with a *Directory Facilitator*. Responds to *FIPA-Request* protocol to create new AWA.

Table 1. Wrapper manager role card

Annotation Wrapper Archivist (AWA). The AWA is the main worker in the wrapper-dedicated society, it provides two services: wrapping a source and providing the annotation archive content for solving queries.

Thus the AWA role is divided into two roles. The first role of the web page wrapper is new: the AWA receives a request for wrapping a web source from the IC; it contains the extraction template and the location of the source; the AWA proceeds to perform all the step of the data extraction process as mentioned earlier and generates a annotation base. The second role already existed in CoMMA: the Annotation Archivist role (AA). The AA role was thus integrated in the AWA with the only modification that it systematically refuses to archive annotations other than the ones it generated through the Web page data extraction. The AA role registers the AWA with an Annotation Mediator (AM), and participates to the distributed query resolution process. Table 2 provides the role card of the AWA.

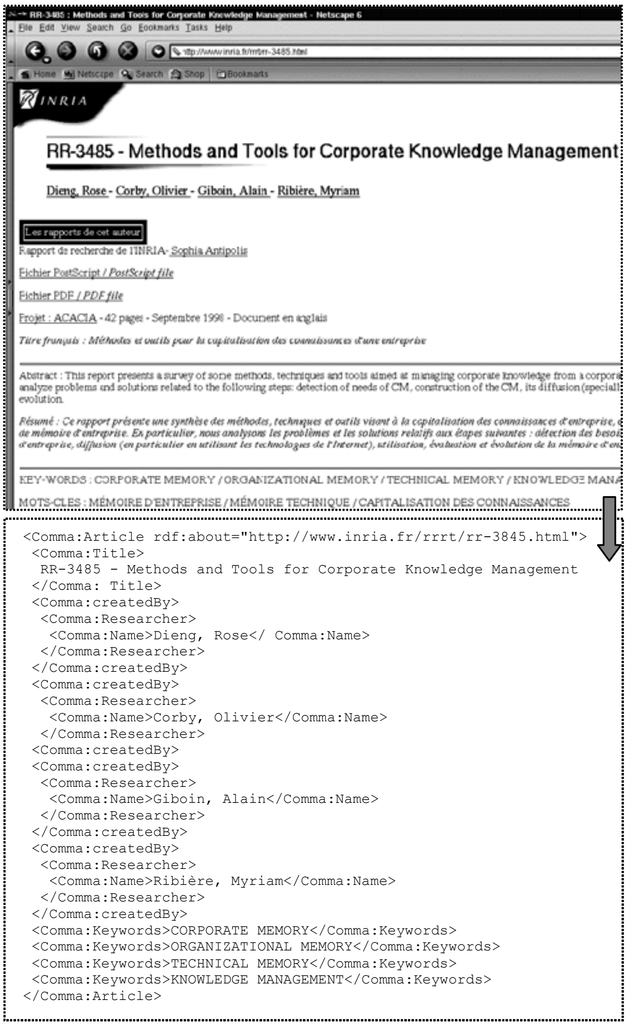
Annotation Wrapper Archivist	
Role Model	Part of the wrapper-dedicated society.
Responsibilities	Wrap the Web source to generate annotations RDF. Store and manage these annotations. Search annotations to respond to the request of an AM
Collaborators	IC, AM, WM, DF, OA
External Interfaces	RDF(S) manipulation using CORESE, XSLT engine and JTidy.
Expertise	CoMMA-Ontology. RDF and query language.
Coordination and Negotiation	Responds to *FIPA-Request* protocol to wrap a web site. Responds to *FIPA-Query* protocol to solve queries

Table 2. Annotation wrapper archivist role card

Outcomes and further work

We have successfully tested the wrapping society on two Web sites: the digital library of research reports of INRIA[1] and the digital library of the technical reports of the software engineering institute of the Carnegie Mellon University[2]. The wrappers correctly annotated these sites with a template we generated from an example page and that was systematically applicable to all the pages. Figure 5 shows an example of annotation extracted from a page describing a research report of INRIA.

Several improvements can be considered. First, in the extraction templates, the data location is represented as an absolute XPath, which is less robust to a change of structure of the Web page than relative paths. We envisage introducing the use of relative XPath, so that data location may be specified with regard to the location of other data and not systematically from the root tag of the page.


RR-3485 : Methods and Tools for Corporate Knowledge Management - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://www.inria.fr/rrrt/rr-3485.html
Home Netscape Search Shop Bookmarks
INRIA
Les rapports de cet auteur

RR-3485 - Methods and Tools for Corporate Knowledge Management

Dieng, Rose - Corby, Olivier - Giboin, Alain - Ribière, Myriam

Rapport de recherche de l'INRIA- Sophia Antipolis

Fichier PostScript / *PostScript file*

Fichier PDF / *PDF file*

Projet : ACACIA - 42 pages - Septembre 1998 - Document en anglais

Titre français : Méthodes et outils pour la capitalisation des connaissances d'une entreprise

Abstract : This report presents a survey of some methods, techniques and tools aimed at managing corporate knowledge from a corpora
analyze problems and solutions related to the following steps: detection of needs of CM, construction of the CM, its diffusion (speciall
evolution.

*Résumé : Ce rapport présente une synthèse des méthodes, techniques et outils visant à la capitalisation des connaissances d'entreprise, e
de mémoire d'entreprise. En particulier, nous analysons les problèmes et les solutions relatifs aux étapes suivantes : détection des besoi
d'entreprise, diffusion (en particulier en utilisant les technologies de l'Internet), utilisation, évaluation et évolution de la mémoire d'en*

KEY-WORDS : CORPORATE MEMORY / ORGANIZATIONAL MEMORY / TECHNICAL MEMORY / KNOWLEDGE MANA

MOTS-CLES : MÉMOIRE D'ENTREPRISE / MÉMOIRE TECHNIQUE / CAPITALISATION DES CONNAISSANCES

```
<Comma:Article rdf:about="http://www.inria.fr/rrrt/rr-3845.html">
 <Comma:Title>
  RR-3485 - Methods and Tools for Corporate Knowledge Management
 </Comma: Title>
 <Comma:createdBy>
  <Comma:Researcher>
   <Comma:Name>Dieng, Rose</ Comma:Name>
  </Comma:Researcher>
 </Comma:createdBy>
 <Comma:createdBy>
  <Comma:Researcher>
   <Comma:Name>Corby, Olivier</Comma:Name>
  </Comma:Researcher>
 </Comma:createdBy>
 <Comma:createdBy>
  <Comma:Researcher>
   <Comma:Name>Giboin, Alain</Comma:Name>
  </Comma:Researcher>
 </Comma:createdBy>
 <Comma:createdBy>
  <Comma:Researcher>
   <Comma:Name>Ribière, Myriam</Comma:Name>
  </Comma:Researcher>
 </Comma:createdBy>
 <Comma:Keywords>CORPORATE MEMORY</Comma:Keywords>
 <Comma:Keywords>ORGANIZATIONAL MEMORY</Comma:Keywords>
 <Comma:Keywords>TECHNICAL MEMORY</Comma:Keywords>
 <Comma:Keywords>KNOWLEDGE MANAGEMENT</Comma:Keywords>
</Comma:Article>
```

Figure 5. Small example of an extracted annotation

The relative path specification would require an additional human intervention to decide which structural clue should be used as a starting root for the relative path. Moreover, to improve the generation and update of templates, the application of machine learning techniques could be envisaged. Secondly, additional functionalities are currently considered for the Wrapper Manager such as killing or resurrecting an AWA to maintain the population of wrappers. Likewise, additional functionalities are envisaged for the AWA, in particular notification services: notify a change in the annotation base, notify a template failure, notify the detection of a keyword that could not be mapped to a concept of the ontology, *etc.* More generally, the management of the life-cycle of agents and their deployment should allow monitoring of important events and the optimization of the distribution of the work: compared to a centralized solution, each wrapper is a source of annotation customized through its XSLT template and the population of these wrappers can be distributed over the intranet to allow scale up.

[1] http://www.inria.fr/rrrt/

[2] http://www.sei.cmu.edu/about/website/indexes/siteIndex/siteIndexTRnum.html

While the original sources are structured in different ways, we can translate and restructure them according to our ontology in a way that may be completely different than the original information structure. Being guided by ontology, the data extraction process can exploit a model of the domain to generate not only the annotation structure but also concept instances replacing keywords that can then be used in inferences. The quality of the information retrieval process is critically dependent on the quality of the annotations since CoMMA uses inferences to exploit the semantic of the annotations, for instance to widen or narrow the users' queries. Relying on automatically generated annotations may not provide the required quality of the semantic annotations and a semi-automated environment in which a human operator is still involved, as proposed here, is the warranty of an acceptable quality.

The traditional Web is not yet a semantic web and this is not going to happen in one night. The approach proposed here could be used in the meantime and could participate to a progressive migration by generating semantic annotations for the semantic Web from the HTML pages of the traditional Web. By using XML technologies the extraction process implementation was simplified using available XML manipulation tools especially for the pre-processing and analysis of the documents. It should also reduce the cost of maintenance. By using the XSLT stylesheets for the representation of the extraction rules, we rely on a standard and thus the components developed for the system may be reused. Moreover, using XSLT and XML, it is possible to merge information extracted from several resources in a single XML result; therefore this approach may be used with heterogeneous sources. It should also be stressed that the same approach can be used to extract customized annotations from other structured sources such as XML documents and RDFS, DAML+OIL or OWL annotations, which could be an application inside the semantic Web.

References

Aguirre, Brena, Cantu-Ortiz, 2000. *Multiagent-based Knowledge Networks*. To appear in the special issue on Knowledge Management of the journal Expert Systems with Applications.

Bellifemine, F., Poggi, A., Rimassa, G., 2001. *Developing multi-agent systems with a FIPA-compliant agent framework.* Software Practice & Experience, (2001) 31:103-128, See also JADE : Java Agent Development Framework at http://sharon.cselt.it/projects/jade.

Bergamaschi, S., Beneventano, D., 1999. *Integration of Information from Multiple Sources of Textual Data*, In Intelligent Information Agent: Agent-Based Information Discovery and Management on the Internet p53-77, Matthias Klusch, Springer 1999

Berney, Ferneley, 1999. *CASMIR: Information Retrieval Based on Collaborative User Profiling*, In Proceedings of PAAM'99, pp. 41-56. www.casmir.net

Bothorel, Thomas, 1999. *A Distributed Agent Based-Platform for Internet User Communities*, In Proceedings of PAAM'99, Lancashire, pp. 23-40.

CoMMA Consortium, 2000. *Corporate Memory Management through Agents*, In Proceedings E-Work & E-Business 2002, Madrid, pp 383-406

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 2002 Hawaii

Dzbor, Paralic, Paralic, 2000. *Knowledge Management in a Distributed Organisation*, In Proceedings of the BASYS'2000 - 4th IEEE/IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing, Kluwer Academic Publishers, London, September 2000, ISBN 0-7923-7958-6, pp. 339-348

Gandon, F., 2002a. *Distributed Artificial Intelligence and Knowledge Management: ontologies and multi-agent systems for a corporate semantic web*, Scientific Philosopher Doctorate Thesis in Informatics, 7th November 2002, INRIA and University of Nice - Sophia Antipolis, www-sop.inria.fr/acacia/personnel/Fabien.Gandon/research/PhD2002/

Gandon, F., 2001. *Engineering an Ontology for a Multi-Agents Corporate Memory System*, In Proceedings ISMICK'01, Université de Technologie de Compiègne, p209-228.

Gandon F., 2002b. *A Multi-Agent Architecture for Distributed Corporate Memories*, Proceedings 16th European Meeting on Cybernetics and Systems Research (EMCSR) April 3 - 5, 2002, Vienna, Austria, pp 623-628.

Gandon, F., Berthelot, L., Dieng-Kuntz R., *A Multi-Agent Platform for a Corporate Semantic Web*, in Proceedings of AAMAS, Castelfranchi, C., Johnson, W.L., (eds) p.1025-1032, July 15-19, 2002, Bologna, Italy

Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., and Widom., J., 1995. *Integrating and Accessing Heterogeneous Information Sources in TSIMMIS*, in Proceedings of the AAAI Symposium on Information Gathering, Stanford, California, March 1995, pp. 61-64.

Muslea, I., Minton, S., Knoblock, C., 1999. *A Hierarchical Approach to Wrapper Induction*, In Proceedings of the Third Annual Conference on Autonomous Agents, p190-197, Seattle, WA USA MAY 1-5, 1999 Edited By Oreb Etzioni ; Jörg P. Müller ; Jeffrey M. Bradshaw, ACM Press / ACM SIGART

Van Elst, L., Abecker, A., 2002. *Domain Ontology Agents in Distributed Organizational Memories* in Knowledge Management and Organizational Memories, Dieng-Kuntz, R., Matta, N., (eds), Kluwer Academic Publishers, p145-158,Boston, July 2002, ISBN 0-7923-7659-5