Semantic Web and Multi-Agents Approach to Corporate Memory Management

Fabien Gandon, Rose Dieng-Kuntz, Olivier Corby, Alain Giboin
ACACIA project, INRIA Sophia Antipolis, <Firstname>.<Name>@sophia.inria.fr

Abstract: Organisations have increasingly large amount of heterogeneous documents to manage and organise in order to turn them into active and helpful corporate memories. We present an approach based on semantic Web and multi-agents systems to implement a framework for corporate semantic Web management.

Key words: semantic web, multi-agents system, corporate memory, knowledge management, ontologies, information retrieval.

1. INTRODUCTION

Increasingly rapid staff turnover, swiftly changing environments, ever growing size and spreading of infrastructures lead organisations to look for tools and methodologies to manage a persistent active memory of their experience. This memory is more and more often taking the form of an intraweb *i.e.* an intranet based on the Web technologies. It leads to amounts of semi-structured information internally available on-line but buried and dormant in their mass. In the CoMMA [1] IST project, we developed a system in charge of managing an intraweb for two knowledge management scenarios: (1) assistance to the integration of newcomers in an organisation and (2) support to the technology monitoring processes. This prototype exploits the semantic Web technologies and it relies on the O'CoMMA ontology used to semantically annotate the intraweb resources. To manage these annotations, information agents were developed to constitute a multi-agent system (MAS) *i.e.* a loosely coupled network of agents that work together as a society. A MAS is heterogeneous when it includes agents of at least two types. A Multi-Agents Information System (MAIS) is a MAS aiming at providing some or full range of functionalities for managing and

exploiting information resources. The application of MAIS to corporate memories means that the co-operation of agents aims at enhancing information capitalisation in the company. The MAIS projects CASMIR [4] and Ricochet [5] focus on the gathering of information and adapting interaction to the user's preferences, learning interest to build communities and collaborative filtering inside an organisation. KnowWeb [13] relies on mobile agents to support dynamically changing networked environment and exploits a domain model to extract concepts describing a documents in order to use them to answer queries. RICA [1] maintains a shared taxonomy in which nodes are attached to documents and uses it to push suggestions to interface agents according to user profiles. Finally FRODO [20] is dedicated to building and maintaining distributed organisational memories with an emphasis on the management of domain ontologies.

The CoMMA software architecture is an heterogeneous MAIS that focuses on providing retrieval, pull and push functionalities to support the exploitation of the intraweb during the two application scenarios. The different tasks involved in the exploitation process were allocated to different agent types, the instances of which are distributed over the intranet.

This paper details our approach in three sections: first we present the notion of a *corporate semantic Web* relying on an *ontology*; then we explain the role of *models and the global architecture of the memory*; last, we portray the *multi-agents architecture* for managing the memory. In our conclusion we discuss the evaluation of the prototype.

2. TOWARDS A CORPORATE SEMANTIC WEB

A corporate memory is, by nature, an heterogeneous and distributed information landscape. Corporate memories are facing the same problem of information retrieval and information overload as the Web. Therefore semantic Web technologies can be helpful as emphasised in this section.

2.1 The concept of a corporate semantic Web

XML is becoming an industry standard for exchanging data or documents. In CoMMA, we are especially interested in RDF, the Resource Description Framework [17], and its XML syntax. RDF is the foundation of the semantic Web [3], a promising approach where the semantics of documents is made explicit through annotations to guide later exploitation. RDF allows us to annotate the resources of the memory semantically. It uses a simple data model as the basis for a language for representing properties of resources (anything that can be pointed by an URI such as Web pages or

images) and the relationships between them. The corporate memory is thus studied as a *corporate semantic Web*: we describe the semantic content of corporate documents through semantic annotations then used to search the mass of information of the corporate memory.

Just as an important feature of new software systems is the ability to integrate legacy systems, an important feature of a corporate memory management framework is the ability to integrate the legacy archives. Since RDF annotations can be either internal or external to the document, existing documents may be kept intact and annotated externally. This is complementary to the MAS ability to include legacy systems by wrapping them into an agent. Even if wrappers are not addressed in CoMMA, a new agent could be added to wrap, for instance, the access to a database using a mapping between the DB schema and the O'CoMMA ontology.

RDF makes no assumption about a particular application domain, nor defines *a priori* the semantics of any application domain; the annotations are based on an ontology which is described and shared thanks to the primitives provided by RDF Schema [6] (RDFS). The idea is (a) to specify the corporate memory concepts and their relationships in an ontology formalised in a schema in RDFS, (b) to annotate the documents of the memory in RDF using the schema (c) to exploit the annotations to search the memory.

2.2 Ontology engineering and its result: O'CoMMA

We proposed a method to build ontologies and applied it to obtain O'CoMMA (see [15] for more details). The method relies on three stages:

1. Scenario analysis and Data collection: Scenarios are textual descriptions of the organisational activities and interactions concerning the intended application. They were used for data-collection together with semi-structured interviews, work-place observation and document analysis. This last technique can be coupled with natural language processing tools for scaling-up the approach. Whenever possible, existing ontologies were partially reused (mainly TOVE[1] and Cyc[2]): we manually revisited the parts that were interesting for our scenarios ; if the informal definition of a notion had the meaning we were looking for, the terms denoting this notion and the definition were added to the lexicon from which we built the ontology. Other non company-specific sources or standards helped us structure upper parts of the ontology or list the leaves of some precise specialised area (e.g. MIME).

2. Terms collection, analysis and organisation: The terms denoting notions appearing relevant for the application scenarios are collected, analysed and organised in a set of informal tables forming a lexicon on

[1] www.eil.utoronto.ca/tove/ontoTOC.html
[2] www.cyc.com/cyc-2-1/cover.html

which the ontology will be built. The synonyms and ambiguous terms are spotted and marked as such. Definitions in natural language are proposed, discussed and refined especially to eliminate fuzziness, circular definitions and incoherence.

3. Structuring the ontology: Combining bottom-up, top-down and middle-out approaches as three complementary perspectives of a complete methodology, the obtained concepts are iteratively structured in a taxonomy. The initial tables evolve from a semi-informal representation (terminological tables of terms & notions) towards semi-formal representation (subsumption links, signatures of relations) until each notion has a unique formal identifier (usually one of its terms) and a position in the hierarchy of concepts or relations. Tables are then translated in RDFS using scripts.

O'CoMMA contains: 470 concepts organised in a taxonomy with a depth of 13 subsumption links; 79 relations organised in a taxonomy with a depth of 2 subsumption links; 715 terms in English and 699 in French to label these primitives; 547 definitions in French and 550 in English to explain the meaning of these notions. In the ontology three layers appear: (1) a general top that roughly looks like other top-ontologies, (2) a large and ever growing middle layer divided in two main branches: one generic to corporate memory domain (document, organisation, people...) and one dedicated to the application domain (e.g. telecom: wireless, network, etc.), (3) an extension layer, specific to the scenario and to the company, with complex concepts (Trend analysis report, New Employee Route Card, etc.). The upper part, which is quite abstract, and the first part of the middle layer, which describes concepts common to corporate memory applications, are reusable in other corporate memory application. The second part of the middle layer, which deals with the application domain, is reusable only for scenarios in the same domain. The last layer containing specific concepts is not reusable as soon as the organisation, the scenario or the application domain changes. However, this last layer is by far the closest to day-to-day users' interest.

Concepts are formalised as RDFS classes. Relations and attributes are formalised as RDFS properties. Instances of these classes and properties are created to formulate annotations. Terms are formalised as RDFS labels of classes and properties and are independent from the internal unique system identifier of the class or property. Likewise the natural language definitions are captured as RDFS comments. The ability to specify the natural language used enables us to have multilingual ontologies. A notion (concept or property) with several terms linked to it is characteristic from the synonymy of these terms. A term associated to several notions is ambiguous.

Using XSLT style sheets, we reproduce the intermediate documents that were used to build the ontology and we propose different views of the ontology: (a) initial terminological table representing a lexicon of the

memory; (b) tables of concepts and properties; (c) pages for browsing and searching at the conceptual or terminological levels: they allow search for concepts or relations linked to a term, navigation in the taxonomy, search for relations having a signature compatible with a given concept; (d) list of instances of a notion: a sample of instances plays the role of examples to ease understanding of a notion; (e) filtered view of the ontology using a user's profile so as to propose preferred entrance points in the ontology; (f) indented tree of concepts or relations.

The choice of RDF(S) enables us to base our system on a standard that benefits from the web-based technologies for networking, display and browsing, and this is an asset for the integration to a corporate intranet.

2.3 CORESE: Conceptual Resource Search Engine

As CoMMA aims at offering information retrieval from the corporate memory, we needed to rely on a search engine. Keyword-based search engines works at the term level. Ontologies are a means to enable software to reason at the semantic level. To manipulate the ontology, the annotations, and infer from them, we developed CORESE [8] a prototype of search engine enabling inferences on RDF annotations and information retrieval from them. CORESE combines the advantages of using (a) the RDF(S) framework for expressing and exchanging metadata, and (b) the query and inference mechanisms available for Conceptual Graph (CG) formalism [18]. CORESE is an alternative to SiLRi [10] which uses frame logic. There is an adequacy between RDF(S) and CG: RDF annotations are mapped to factual CGs; the class hierarchy and the property hierarchy of an RDF schema are mapped to a concept type hierarchy and a relation type hierarchy in CGs.

CORESE queries are RDF statements with wildcard characters to describe the pattern to be found, the values to be returned and the co-references. Regular expressions are used to constrain literal values and additional operators are used to express disjunction and negation. The RDF query is translated into a CG which is projected on the CG base in order to find matching graphs and to extract the requested values. The answers are then translated back into RDF. The CG projection mechanism takes into account the specialisation links described in the hierarchies translated from the RDF schema. Both precision and recall are thus improved.

As a lesson of CoMMA, a limitation of RDFS appeared when formalising implicit information and background knowledge. For instance, when we declare that someone manages a group, it is implicit that this person is a manager. Thus the 'manager' concept should be a 'defined concept', *i.e.* a concept having an explicit definition enabling this concept to be derived from other existing concepts whenever possible. However the

notion of defined concept does not exist in RDFS, even thought the ability to factorise knowledge in an ontology requires the ability to express formal definitions. In the current version of the CoMMA system, the formal definitions are coded in rules written in an RDF/XML rule language specially created for RDF(S) and CORESE. As explained in [9], an inference engine exploits these rules to complete the annotation base with deducible implicit facts. Instead, one could extend the RDFS model to add the missing expressiveness as in DRDF(S) [11], OIL [14], or DAML+OIL [21]. For instance, symmetry, transitivity and reflexivity characteristics of properties required CORESE-specific extensions of RDFS.

Although CORESE can be used in a client-server fashion, it also offers an API; thus, in CoMMA, modules of CORESE are integrated in the agents handling the ontology or the annotations, so as to provide them with the abilities needed for their roles.

3. MODEL-BASED MEMORY

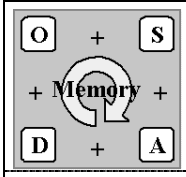
Users of the corporate memory are, by nature, heterogeneous and distributed in the corporation. In order to give the CoMMA system an insight of its environment and of the users it is interacting with, the memory is based on models of the organisational structure and on user profiles enabling customisation, learning of preferences and push technologies.

To materialise the user profiles, we *annotate people* using primitives defined in the ontology. A user's profile captures aspects of the user that we identified as relevant for the system behaviour. It contains administrative information and explicit preferences (e.g. topic interests). It also positions the user in the organisation: role, location and potential acquaintance network, enabling the system to target push actions. In addition, the system derives information from the usage made by the user. It collects the history of visited documents and user's feedback and from this it learns some of the user's interests [16]. These derived criteria are then used for result presentation or push technology enabling the emergence of communities of interest. The user’s profile also records preferred entrance points into the ontology in order to hide the ontology upper level and to propose middle concepts (e.g. person, document, domain topics) from which the user can start browsing the ontology in a MyYahoo fashion.

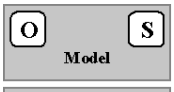
An enterprise model is an oriented, focused and somewhat simplified explicit representation of the organisation. So far, the enterprise modelling field has been mainly concerned with simulation and optimisation of the production system design. It provides benchmark for business processes and are used for re-engineering them. But the shift in the market rules led

organisations to become aware of the value of their memory and the fact that organisation models have a role to play in this application too [19]. In CoMMA, the model aims at supporting corporate memory activities involved in the application scenario. The system exploits the aspects described in the model for the interaction between agents and above all between agents and users. We used RDF to implement our organisational description, *annotating the organisational entities* (departments, activities, laboratories, etc.) with their relations (manages, employs, includes, etc.).

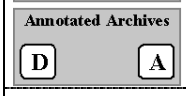
Annotated environments containing explanations of the purpose and the uses of spaces and activities allow agents to quickly become intelligent actors in those spaces [12]. In CoMMA, the corporate memory is an annotated world: with RDF(S), we describe the semantic content of documents and the organisational state of affair through semantic annotations (*Figure 1*); then agents use and infer from these annotations in order to search the mass of information of the corporate memory.



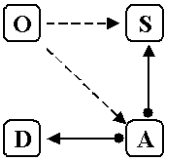
The memory is composed of the **D**ocuments, their **A**nnotations, the **S**tate of Affairs (user profiles and organisation model) and the **O**ntology. The whole follows a prototypical life-cycle, evolving and interacting with each other.



The **O**ntology and the **S**tate of Affairs form the model on which is based the structuring of the memory.



The archive structure relies on the **A**nnotations of the **D**ocumentary resources.



The **A**nnotations and the **S**tate of Affairs are formalised using the conceptual vocabulary provided by the **O**ntology. The **A**nnotations refer to the **D**ocuments (ex: report http://www...) and to the objects of the **S**tate of Affair (ex: written by Mr. X for the division ABCD)

Figure 1. The Architecture of the Memory

4. MULTI-AGENTS SOFTWARE ARCHITECTURE

The tasks to be performed on the corporate memory, the corporate memory itself and the population of users are distributed and heterogeneous. Therefore, it is interesting to have a heterogeneous and distributed software architecture. Multi-agents systems have been acknowledged as an excellent candidate to provide a software architecture supporting the semantic Web framework [3]. The MAS paradigm appeared very well suited for the deployment of a software architecture above the distributed information

landscape of the corporate memory: on the one hand, individual agents locally adapt to users and resources they are dedicated to; on the other hand, thanks to co-operating software agents distributed over the intranet, the system capitalises an integrated and global view of the corporate memory.

A MAS architecture is a structure that portrays the different families of agents and their relationships. A configuration is an instantiation of an architecture with a chosen arrangement and an appropriate number of agents of each type. One given architecture can lead to several configurations and a given configuration is tightly linked to the topography and context of the place where it is deployed (organisational and intranet layout, stakeholders location). Thus, the architecture must be designed so that the set of possible configurations covers the different corporate organisational layouts foreseeable. The configuration is studied and documented at deployment time whereas the architectural description is studied and fixed at design time. The architectural analysis starts from the highest level of abstraction (i.e. the society) and by successive refinements (i.e. nested sub-societies) it goes down to the point where agent roles and interactions can be identified.

4.1 From the Macro level to the Micro level

We adopted an organisational approach: the MAS architecture is tackled, as in a human society, in terms of roles and relationships. The functional requirements of the system do not simply map to some agent functionality but influence and are finally diluted in the dynamic social interactions of individual agents and in the set of abilities, roles and behaviours attached to them. Considering the system functionalities, we identified three sub-societies of agents dedicated to resources (ontology and model; annotations; yellow pages needed for managing interconnection) and one dedicated to users (*Figure 2*). Analysing the resource-dedicated sub-societies, we found that there was a recurrent set of possible organisations 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 organisation is preferred to another.

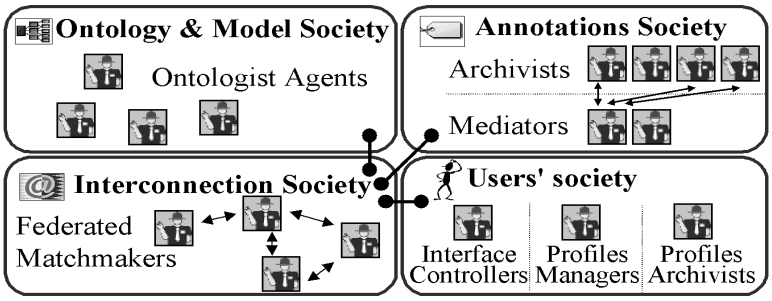


Figure 2. Multi-Agents Architecture of CoMMA

The sub-society dedicated to the ontology and model is currently organised as a replication sub-society (*i.e.* an ontologist agent has a complete copy of the ontology). The annotation-dedicated sub-society is a hierarchical organisation as described in the last section. The yellow pages agents are in a peer-to-peer organisation and are provided by the JADE platform [2] used in CoMMA. Agents from the user-dedicated sub-society are concerned with interface, monitoring, assistance and adaptation to the user. Because they are not related to a resource type like the previous ones, they cannot be studied using our typology. We can distinguish at least two recurrent roles in this type of sub-society: (1) user interface management: to dialogue with the users for enabling them to express their request and refine them, and to present results in an adequate format; (2) management of user profiles: to store the profiles and make them available for interface purposes, learning techniques and pro-active searches.

From the architecture analysis, we identified agent roles and we studied their characteristics and interactions in order to implement the corresponding behaviours in a set of agent types. Roles represent the position of an agent in a society and the responsibilities and activities assigned to this position and expected by others to be fulfilled. Then comes the specification of role interactions specified with protocols that the agents must follow for the MAS to work properly. The definition of a protocol starts with an acquaintance graph at role level, that is a directed graph identifying communication pathways between agents playing the considered roles. Then we specified the possible sequences of messages. Both the acquaintance network and the protocols derived from the organisational analysis and the use cases dictated by the application scenarios.

From the role and interaction descriptions, the different partners of CoMMA proposed and implemented agent types that fulfil one or more roles. Behaviours come from the implementation choices determining the responses, actions and reactions of the agent. The implementation of a behaviour is constrained by the associated role and is subject to the toolbox of technical abilities available to the designers.

4.2 Example of the annotations-dedicated society

In this sub-society the Annotation Mediator (AM) is in charge of handling annotations distributed over Annotation Archivists (AAs). The stake is 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. To allocate a newly posted annotation, an AM broadcasts a call for proposal to the AAs. Each AA measures how semantically close the annotation is, , from the types

of concepts and relations present in its archive. The closest AA wins the bid. We defined a pseudo-distance based on the ontology hierarchy and AM uses 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 of notions 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, we integrated modules of CORESE [8] in the agent types implementing these roles to provide the needed technical abilities.

5. EVALUATION & CONCLUSION

The prototype was evaluated by end-users from a telecom company (T-Nova System) and a construction research centre (CSTB) through two trials at the 8th month and the 22nd month. The very last prototype was presented and discussed during an open day at the end of the project.

During the first trial we performed: (a) an evaluation of the *architecture*, (b) an evaluation of the design *methodology*, and (c) an evaluation from the *user*'s point of view of *usefulness* and *usability*. Four T-Nova employees participated for the new employee insertion (NEI) scenario. Three CSTB librarians participated for the technology monitoring (TM) scenario. As a result, the system meet the needs (usefulness) but its interfaces were not user-friendly (usability). The reason was that the first interfaces were built for designers and knowledge engineers to test the integration, and not for end-users. Thus users could not have a clear view of the system functionalities. Interface were reengineered for the second trial.

The second trial was prepared by a series of iterative evaluations with end-users participating directly to the re-design of the interfaces. Then we made a final evaluation in two steps: (a) users (6 for TM scenario and 4 for NEI) used the system to perform scenario-related tasks ; their comments were classified in terms of positive and negative usability aspects and spontaneous recommendations. (b) 4 TM users and 1 NEI user were rated the severity of the negative usability aspects that had been identified. Then we asked a GUI designer to assess the design effort necessary to implement the proposed recommendations. Both ratings were used to determine the importance of the critics made. Results showed that the CoMMA system was still *useful*, but also *usable*: the GUIs being less complex (*Figure 3*), users accepted them, and were not reluctant to manipulate them.

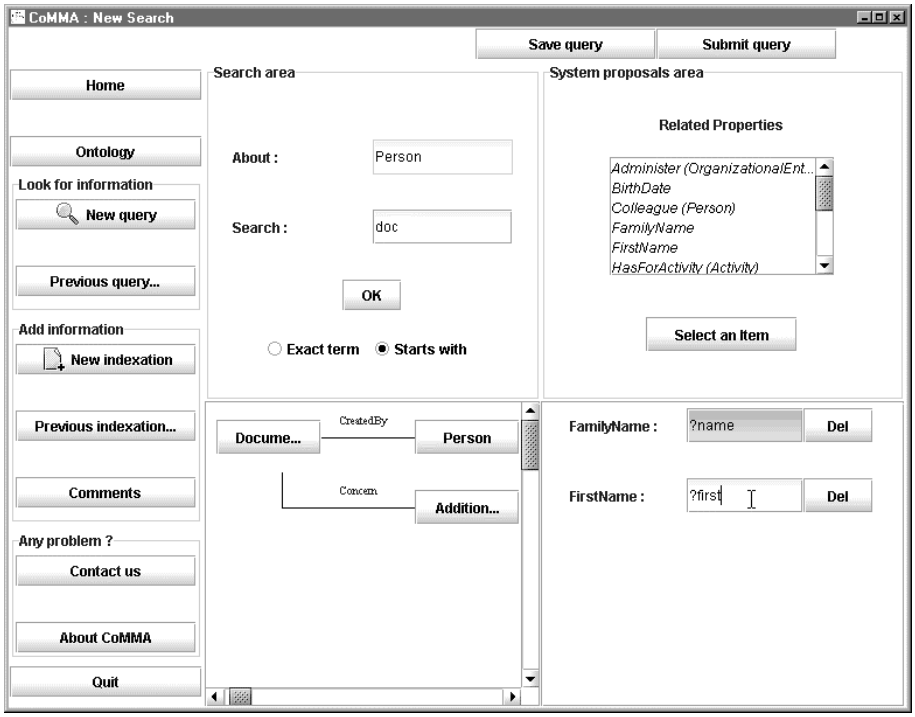


Figure 3. Query interface in CoMMA

Both evaluations were "small-scale" evaluations: small number of users, small number of annotations (about 1000), and small duration of use. This short-period of use did not allow us to observe searching, indexing, and learning phenomena. However, a middle-scale test was successfully performed in our organisation with 10000 annotations about 850 documents.

From the developer point of view, we appreciated the ontology-oriented and agent-oriented approach because it supported specification and distribution of implementation while smoothing the integration phase. We are convinced that an approach based on knowledge engineering (*i.e.* formalising knowledge about resources of an intraweb through semantic annotations based on an ontology) and distributed artificial intelligence (*i.e.* multi-agents information system loosely coupled by a cooperation based on semantic message exchanges) can provide a powerful paradigm to solve complex distributed problems such as organisational memory management.

6. ACKNOWLEDGEMENTS

We thank our colleagues of ACACIA and CoMMA (IST-1999-12217) for our discussions, and the European Commission that funded the project.

7. REFERENCES

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

[2] Bellifemine, Poggi, Rimassa, Developing multi agent systems with a FIPA-compliant agent framework. Software Practice & Experience, (2001) 31:103-128

[3] Berners-Lee, Hendler, Lassila, The Semantic Web, *Scientific American*, May 2001:35-43

[4] Berney, Ferneley, CASMIR: Information Retrieval Based on Collaborative User Profiling, Proc. of PAAM'99, pp. 41-56. www.casmir.net

[5] Bothorel, Thomas, A Distributed Agent Based-Platform for Internet User Communities, In Proc. of PAAM'99, Lancashire, pp. 23-40.

[6] Brickley, Guha, Resource Description Framework Schema Specification 1.0, W3C Candidate Recommendation 27 March 2000

[7] CoMMA http://www.si.fr.atosorigin.com/sophia/comma/Htm/HomePage.htm

[8] Corby, Dieng, Hébert, A Conceptual Graph Model for W3C Resource Description Framework. In Proc. ICCS'2000 Darmstadt Germany

[9] Corby, Faron, CORESE: a corporate semantic web engine, Proc. of WWW02 Workshop on Real World RDF & Semantic Web, Hawai 2002.

[10] Decker, Brickley, Saarela, Angele. A Query Service for RDF. *Query Languages* 98, W3C Workshop.

[11] Delteil, Faron, Dieng, Extension of RDFS based on the CG formalism, *Proc. ICCS'01*

[12] Doyle, Hayes-Roth, Agents in Annotated Worlds, In Proc. Autonomous Agents, ACM Press / ACM SIGART, Minneapolis, MN USA (1998) p173-180

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

[14] Fensel, Van Harmelen, Horrocks, McGuinness, Patel-Schneider. OIL: An ontology infrastructure for the semantic web. IEEE Intelligent Systems, 16(2):38-45, 2001.

[15] Gandon, Engineering an Ontology for a Multi-Agents Corporate Memory System, In Proc. ISMICK'01, Université de Technologie de Compiègne, p209-228.

[16] Kiss, Quinqueton, Multiagent Cooperative Learning of User Preferences, Proc. of European CMLP & PKDD, 2001.

[17] Lassila, Swick, Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999

[18] Sowa, *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, 1984.

[19] Rolstadås, Development trends to support Enterprise Modeling, in Rolstadås and Andersen *Enterprise Modeling: Improving Global Industrial Competitiveness*, Kluwer Academic Publisher, p3-16, 2000

[20] Van Elst, Abecker, Domain Ontology Agents in Distributed Organizational Memories. To appear in Dieng & Matta eds, Knowledge Management and Organizational Memories, Kluwer, 2002 .

[21] Van Harmelen, Patel-schneider, Horrocks , Reference description of the DAML+OIL ontology markup language. http://pride.daml.org/2000/12/reference.html. March 2001.