To supply organization views, suited to users: an approach to the design of organizational memories

J.-Y. Fortier[1], **C. Cormier**[1], **G. Kassel**[1], **C. Barry**[2], **C. Irastorza**[1], **S. Bruaux**[1]

Abstract. Our work concerns the elaboration of organizational memories (OM). We investigate the feasibility and the benefit of a strong coupling between a knowledge base and a documents' base. Such a coupling supposes that the knowledge to manage is distributed, at one and the same time, in a knowledge model and documents. This distribution raises many questions such as: what knowledge to model?, and how to diffuse the modeled knowledge? In reply to the first question, we recommend modeling the organization for which the memory is elaborated, while insisting on the benefits and the genericity of the approach. For the diffusion of the modeled knowledge, we suggest introducing a mechanism of generation of documents adapted to the user's expectations. This paper presents our first results, in particular a generic software architecture which is currently developed. We illustrate these results with the elaboration of an OM for our research team, which constitutes a privileged experimental field for our work.

Key words: Methodologies and tools for Knowledge Management, Knowledge modeling and enterprise modeling, Semantic Portals, Ontologies and Information Sharing.

1 INTRODUCTION

In this paper, we present the first results of a project whose aim is to define a method and a software architecture to develop hybrid organizational memories (OM). Such OMs are based on the "strong" coupling between a knowledge base (KB) and a documents' base (DB).

The current approaches for the elaboration of hybrid memories consist in the coupling of an ontology with a DB to index documents and/or enable their annotation [2] [12]. These couplings are "weak" because the unique aim of the knowledge model – the ontology, and possibly, some annotations – is to facilitate the access to the documents. The knowledge which is managed is only in the documents. In our project, on the contrary, we study the feasibility and the benefit of a "strong" coupling. We consider that the knowledge to manage is both in the knowledge model and in the documents. We therefore seek to place the two forms of knowledge explicitation on the same level, searching for the strongest complementarity to manage knowledge.

Concerning the use of the knowledge, each explicitation form has indeed its own characteristics: the more the knowledge is structured, the easier it is to transform and to diffuse it; on the other hand, the more important is the initial cost for the formalization and the more complex is its maintenance. Then these different explicitation forms are complementary and they must be chosen by considering the value of the knowledge for the organization and its cost of acquisition and formalization [5].

The notion of documents' enrichment with the help of formal models of knowledge, as defined by [19] and exemplified in the project ScholOnto [6], is going towards a strong coupling. However, concerning the access to knowledge and its diffusion, an important dissymmetry persists. On the one hand, with the DB, we have many documents whose contents and form obey to an aim of communication on a fixed subject, for a fixed readership. On the other hand, with the formal knowledge model, we have a monolithic model whose role is not to inform about a fixed subject but to provide contextual knowledge to facilitate the access to documents and the interpretation of their contents. The knowledge model is then always oriented towards the documents and its contents don't benefit of the same facilities of diffusion.

In order to realize a true strong coupling between a KB and a DB and to provide the modeled knowledge with the same facilities of diffusion, we suggest to introduce a mechanism of generation of documents from the modeled knowledge. In response to an user's request concerning a theme tackled in the knowledge model, a structured description, whose contents are adapted to the expectation of the user, is constructed. This mechanism is integrated in a larger document, that we call "knowledge book", playing the role of portal for the memory [24]. The knowledge book presents the subjects tackled in the knowledge model and assembles, for a fixed user, a set of predefined requests.

In order to define our approach and evaluate our software architecture, we have chosen to manage the knowledge of our

[1] LaRIA, Knowledge Engineering team, University of Picardie Jules Verne 5, rue du Moulin Neuf – 80000 Amiens, France, <name>@laria.u-picardie.fr

[2] PSI, INSA of Rouen, Place Émile Blondel, BP 08, 76131 Mont Saint Aignan Cedex, France greboval@insa-rouen.fr

research team [16]. We have also developed a prototype of OM, in a near context of industrial research [11].

In the paper, we complete the description of our work according to the following plan:

- Section 2 describes the principles followed to achieve a strong coupling between the knowledge model and the documents.
- Section 3 presents a generic software architecture of OM, whose components are currently implemented. In particular, we see how the module in charge of the construction of documents, called "Writer", collaborates with the modules implementing our knowledge representation language DefOnto [17]: a compiler and an inference engine.
- Section 4 presents the method of ontologies construction, OntoSpec [15], which is associated with the approach, and the role played by the two manifestations of the ontology that the method considers: a semi-informal ontology and a computational ontology specified in DefOnto.
- Section 5 emphasizes the contribution of DefOnto in the elaboration of the memory, in terms of power of expression and inferential services.
- Section 6 presents the prototype of OM realized in the project of managing our team knowledge.
- Section 7 assembles comparisons between our work and other projects.

2 PRINCIPLES OF A STRONG COUPLING

This section presents principles retained to realize a strong coupling. They concern the choice of knowledge to model §2.1 and the way the modeled knowledge is diffused §2.2.

2.1 To model the organization: interest and genericity of the approach

As the choice of the knowledge to model depends, as we said before, on many parameters, notably on the value of knowledge for the organization, one has to expect that this choice varies from one application to the other. In this section, we present the choice that we have undertaken in our projects. In addition we put forward arguments to consider this choice as generic, that is to say susceptible to be retained in a large number of applications.

We suggest modeling the organization for which the memory is constructed and to elaborate this model by successive refinements.

A model of the organization describes the structure of the organization, its members, its partners, its activities, its products or results, and its documentation. Such a model satisfies different objectives, often put forward to motivate projects of OM construction:

- *To help diffuse the documentation*. The model contains many references to documents. For example, the description of actors or partners of the organization makes reference to documents of which they are authors or publishers. Similarly, the description of activities led within the organization (e.g, meetings, projects) makes reference to related documents (e.g., meeting reviews, documentation of project). Therefore the model of the organization offers a context to access to documents.
- *To help integrate a new actor in the organization.* Such a model notably informs the newcomer of activities to which the different members of the organization participate, their responsibilities in these activities, the state of project furtherance. In addition, the newcomer is aware of documents that he/she has to consult as a priority and of their location.

As we can see, a simple model of the organization, making no reference to the business knowledge of the organization, can offer a good return on the investment in time required for modeling knowledge. It allows to construct, at a low cost, a first version of the memory and to interest actors of the organization in the knowledge management project.

Thereafter, such a model can be complicated, notably by modeling business knowledge of the organization. It becomes then possible, for example, to describe the competence of organization members by connecting them to business objects (e.g.: [7]), to preserve knowledge for activities judged crucial (e.g.: [13]) or to index documents by their contents to facilitate their access (e.g.: [22]). We thus recommend an incremental development of the model of the organization, each supplementary effort of modeling leading to the addition of a new service to the memory.

Should be noted that elaborating the model of the organization supposes that the notions necessary for its expression have been beforehand defined, for example the notions of "partner", "confidential document" or "internal project". Here we meet the question of the construction of ontologies. It is necessary therefore to consider that the KB contains both the model of the organization and its associated ontology. In section 4, we deal with the method of construction of the ontology and we emphasize the role played by the ontology in the exploitation of the memory.

2.2 Diffusion of the modeled knowledge by means of documents suited to the profile of the user

The KB is specified in a formal language of representation, as far as we want the OM to reason about the modeled knowledge, and we therefore face the question of the mode of diffusion of this knowledge.

The contents of the KB can't be indeed diffused as they stand, for several reasons. On the one hand, representations of knowledge are specified in a computer language and are therefore hardly understandable for a human being. On the other hand, the order of the entities of representation in the KB corresponds to a KB development logic, not to a knowledge transmission logic. In addition, it is out of the question to present the totality of the KB to a member of the organization, if only because of access rights to some information.

We therefore suggest generating from the KB a "readable" document, whose contents are adapted to the needs of the user, that we call "knowledge book". Such a document contains a summary (a table of contents) gathering, in the way of a classic book, an ordered set of themes. Each theme corresponds to a subject tackled in the knowledge model, in our case to a partial view of the organization. More precisely, a theme corresponds to an object, or a list of objects, described in the knowledge model, for example a person, a project or an organization. The selection of a theme in the summary, to consult the contents of the corresponding section of the book, leads to the construction on the fly of the presentation of the corresponding object (or set of objects), by a software module called "Writer". The Writer takes into account both predefined models of presentation of types of objects (one does not present a person like a project), the ontology and the profile of the user, to elaborate a structured description.

Thus, each user receives a book, where both the summary and the contents of sections are adapted to his expectations and to his profile. This supposes that the memory disposes of different summaries, adapted to the different profiles of users. These summaries are defined by a member of the organization playing the role of administrator of the memory.

3 SOFTWARE ARCHITECTURE

Let's see how the different functionalities we have just described are materialized in term of a software architecture. Figure 1 shows a general view of our OM architecture (the main software modules are graphically represented by rectangles). It defines two different roles that may be played by organization actors, the roles of editor and consultant.

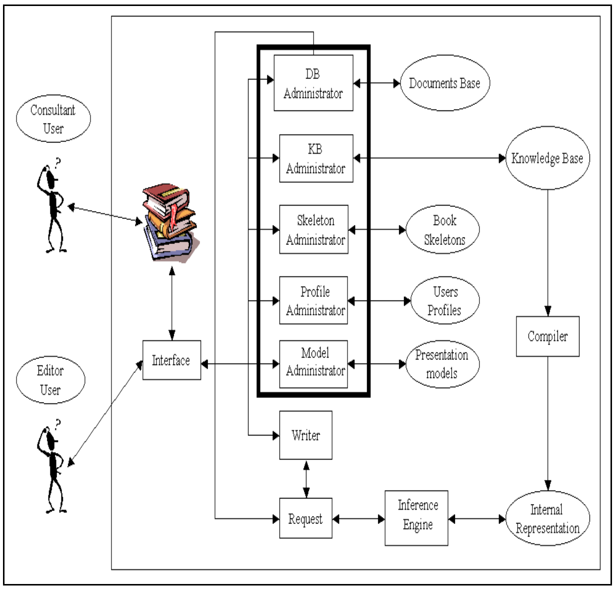

Fig. 1. OM architecture

Acting as an editor, the user edits different knowledge sources in order to build them up and maintain them:

– The Knowledge Base, composed of the ontology and the organization' model, is formalized in the DefOnto language. This KB is translated by a compiler to an internal representation (in JAVA), whose structure is optimized to allow inferences. The addition of new descriptions in the organization' model leads to extensions of the internal representation (incremental compilation). The inferences are realized by the Inference Engine.
– The Documents' Base. The addition and the deletion of documents are assumed by the document administrator, who also maintains the DefOnto description of the documents in the KB, with the help of the KB Administrator. We can notice that some documents can be described in the KB without being stored in the DB. This is the case, for instance, with papers, archived in cupboards, or with web sites for which we only archive the address.
– The user profiles, which consist in a list of user "types": consultant (internal or external to the organization), editor (of the ontology, of the organization model, etc.). Each type of user is linked to a book skeleton and some rights of access regarding the level of confidentiality of the knowledge or the documents.
– The book skeletons, which contain the structured summaries of the books. A summary is defined by a list of themes and sub-themes corresponding to partial views of the model of the organization. Each terminal element of the summary corresponds to a precise information request, related to one object or a list of objects, that is sent to the Writer by the Interface.
– The presentation models, which consist in an ordered set of properties for a given type of object. Yet for a same type of object to present, the model may differ according to the rights of the consultant (an external person can't see confidential information), the level of description expected (we don't expect the same description in an introduction and in a sub-section dedicated to this object).

As we can see, there are different types of editors, each one needing some special abilities: the editor of users profiles must be aware of consultants needs whereas the ontology editor should be a knowledge engineer with knowledge modeling skills and so, he may not be an employee of the organization.

Acting as a consultant, the user has access to his knowledge book. At that time, he is able to perform different actions:

– To select an entry of the summary, that leads to the writing of the corresponding section by the Writer and its visualization by the Interface. The related information request is translated by the Writer to elementary requests that are transmitted to the Request module. The latter searches into the KB using the Inference Engine to answer these elementary requests. In the end, the Writer exploits the produced answers, the user profile and presentation models predefined for each type of entity (e.g.: person, project) to write the section.
– To express an information request, not anticipated in the summary, and whose answer will flesh out the book. The user has to deal with the Writer, which will help him to express his request. The rest of the processing is similar to that of a predefined request of the summary.

4 BUILDING AND ROLES PLAYED BY AN ONTOLOGY IN THE CONCEPTION AND EXPLOITATION OF THE MEMORY

We have seen that the KB is composed of a model of the organization and of an ontology. The latter has different functions, constituting an help for the conception and exploitation of the model of the organization. In this section, we emphasized the contribution of the ontology to strong coupling, from methodological point of view, successively presenting: the method "OntoSpec" for building ontologies [15], which is integrated in our method of OM building §4.1; the two manifestations of the ontology considered by OntoSpec, a semi-informal conceptual ontology §4.2 and a computational ontology §4.3, specifying their respective role; the ontology OntoOrg, built in different projects of memories construction, and which constitutes a resource bound to our method §4.4.

4.1 Ontology construction

The method OntoSpec [15] results from an evolution of the methodological framework defined in [18]. OntoSpec suggests to organize the development of an ontology with two main steps named "ontologization" and "operationalization":

– *Ontologization* corresponds to acquisition and modeling of ontological knowledge (the notions). It is guided by modeling primitives (e.g., concept, relation, essential property), the specification being made at the "knowledge level", which means that no computer constraint is taken into account (e.g., language syntax, inference time). This step leads to a *conceptual ontology*, specified in a semi-informal way.
– *Operationalization* takes as data the *conceptual ontology* to code it into the language of representation DefOnto. As DefOnto is also a programming language, this step leads to a *computational ontology.*

Such a decomposition is inherited from KBS building methods which, like the CommonKADS methodology [23], distinguish two levels of modeling: a modeling to make sense and a modeling to implement a system. We also find it in methodologies for ontology building like METHONTOLOGY [10] and TERMINAE [4][3].

Among these two steps, ontologization is certainly the more crucial step. It allows cooperative work between a knowledge engineer and the actors of the organization receiving the memory, to get a coherent, complete and consensual, system of concepts. It is led by the tasks that the memory must assist, tasks which determine the nature of the organization' model to consider. Once the conceptual ontology obtained, the operationalization consists in coding the modeled knowledge using the language DefOnto. This step can be done by a computer scientist who knows the constructions of the language and its inferential services.

4.2 The conceptual ontology

The conceptual ontology is specified in a semi-informal way, which means that definitions of conceptual entities (concepts and relations) are expressed in a strongly structured and controlled natural language.

The structure of a definition (*cf.* examples in figure 2) is based on a classification of propositions which are likely to contribute to the contents of the definition:

– Some propositions are used to express properties of objects denoted by the conceptual entity. At a first level, the properties are classified as "essential" properties (EP) or "incidental" (IP)[4]. At a second level, the properties are classified according to "roles" they play regarding the defined conceptual entity. These roles can be abstract (e.g., Necessary Condition (NC), Sufficient Condition (SC)) or more specific, and in this case specialize the previous (e.g., Subsumption Link (SL), Subsumption Link with Differentia (SLD), Link of Mutual Exclusion (LME), Relational Link (RL), Domain Restriction (DR), Range Restriction (RR)).
– Other propositions are used to express comments, aiming, either at clarifying the definition supplying examples and/or counter examples, or, for the modeler, at memorizing choices of modeling. An important example of comment, intended to reinforce the understanding of definitions, consists in explaining the presence of "semantic axes" (SA) when a notion is specialized according to several dimensions. So, the notion of "document" can give birth to notions of "electronic document" and "paper document", according to the physical support used, notions of "document in French" and "document in English", according to the language used, finally notions of "announcement of thesis presentation" and of "call to participation to scientific event", according to the communicating intention of the document's author.

Employee: [EP/SLD] an EMPLOYEE is a PERSON who WORKS ON BEHALF OF an EMPLOYER. [EP/RL] Every EMPLOYEE IS PAID BY the EMPLOYER who employs him. [SA] The concept EMPLOYEE is specialized in ENGINEER, RESEARCHER according to the nature of work realized by the EMPLOYEE.

Electronic document: [EP/SLD] An ELECTRONIC DOCUMENT is a DOCUMENT which HAS A SUPPORT electronic. [EP/RL] Every ELECTRONIC DOCUMENT HAS FOR FORMAT a FORMAT. [EP/LME] The ELECTRONIC DOCUMENTS are opposed to PAPER DOCUMENTS.

Works on behalf of; is employed by: [EP/SL] WORK ON BEHALF OF implies TO BE USED AS A RESOURCE BY. [DR & RR].An EMPLOYEE WORKS ON BEHALF OF an EMPLOYER.

Fig. 2. Semi-informal definitions of two concepts and of one relation

In addition to this catalogue of propositions, whose role is to control the contents of a definition, the knowledge engineer has rules at his disposal to control the expression in language, such as:

– Rules to paraphrase each type of proposition, in order to get homogenous definitions.
– Typographical conventions to place words used in the definition in relation to the current conceptualization. For ex-

[3] A comparison of OntoSpec with these methods is out of range of this paper. The interested reader will read (Kassel, 2002).

[4] The essential properties are verified by the objects denoted by the concept in every situation, or world, possible. They are thus "really" definitional. Conversely, the incidental properties are satisfied only in the subrange of situations where memory is likely to be confronted.

ample, when a term is used in a meaning corresponding to a notion of the ontology, this word is written in capital letters.

In addition to being used as a basis for coding the computational ontology in DefOnto, the conceptual ontology becomes encapsulated, as it is, in the computational ontology, which allows it to be exploited by the Writer. So, to answer a query on the sense of a term, the Writer exploits the semi-informal structured definition associated to the concept, to extract the definitional properties and to suggest a definition of the term.

4.3 The computational ontology

The computational ontology is specified with the language DefOnto [17]. It is obtained by coding semi-informal propositions into formal propositions (*cf*. figure 3). However, due to a limited propositional power of expression of DefOnto, some semi-informal propositions don't have their equivalent in DefOnto. In the definition of the concept EMPLOYEE, it is for example the case for the second proposition[5]. This reduction of sense justifies that we keep the conceptual ontology encapsulated in the computational ontology.

DefOnto is a compiled language. The formal ontology, and declarations of objects which are instances of the concepts of ontology, are translated into an internal data structure (*cf.* figure 1). The compiler successively makes a lexical and syntactic analysis, then a semantic one of the internal representation of DefOnto. The internal data structure is optimized to provide inferential services. The latter are described in §5.2.

```
(DefGenConcept #employee
    = [#person] -> (ME#works_on_behalf_of) -> [#employer])
(DefGenConcept #electronic_document
    = [#document] -> (MI#has_a_support) -> "electronic"
    ObjectsProperties
        -> (AE#has_for_format) -> [#format]
    SetProperties
        -> (#is_disjoint_with) -> [#paper_document])
(DefRelation #works_on_behalf_of
    IsA [#is_used_as_resource_by]
    RelationProperties
        -> (#has_for_domain) -> [#employee]
        -> (#has_for_range) -> [#employer])
```

Fig. 3. Definition in DefOnto of two generic concepts and one relation

[5] This proposition "every employee is paid by the employer who employs him" has for equivalent in first order logic : *"x"y employee(x) ® (employer(y) Ù employs(y,x) ® ispaidby(x,y))*. The use of the variable y, in logic, allows to bind the entity which employs to the one which pays. The lack of variable in DefOnto explains why we can't represent this proposition.

4.4 OntoOrg: an ontology dedicated to the management of organizational knowledge

The experience that we have accumulated in different projects of memory building shows that the construction of the ontology remains, in spite of the existence of methodological guides, a complex process which constitutes a real bottle neck for the step of knowledge modeling. This fact explains why we consider the reuse of existing ontologies as a critical aspect for the process of ontology building. In our project, we approach this aspect from the point of view of the management of the life-cycle of ontologies developed for different applications.

To build the ontology OntoPME, within the framework of our project of OM for our team [18] [16], we mainly reused the ontology of the project $(KA)^2$ [2] and, to a lesser extent, Enterprise Ontology [25]. Recently, for a second project, we reused OntoPME to build OntoDCRIT by adapting OntoPME to the needs of a new organization [11]. A work in progress consists in integrating the two ontologies into a generic ontology OntoOrg, which builds on the needs met in the two projects, while erasing the particularities of the concerned organizations. The stake of this work is to have a resource with growing quality to reduce the cost of ontology building for future applications.

OntoOrg ontology is composed of five sub-ontologies corresponding to five great themes, or types of objects: activities, documents, events, organizations and persons. Figure 4 graphically shows semantic axes structuring the sub-ontology of documents[6].

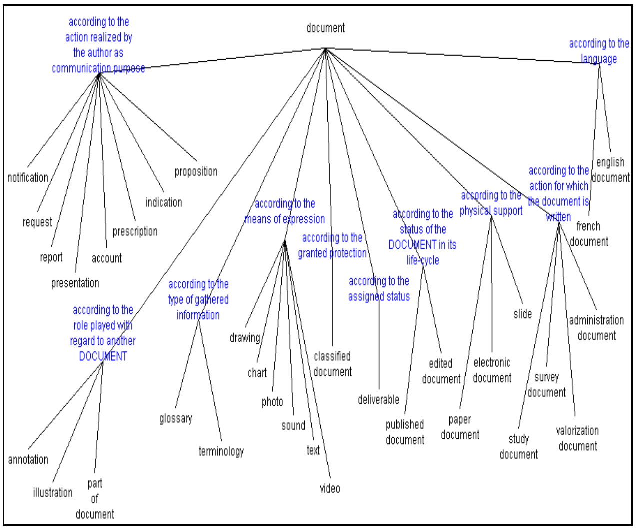

Fig. 4. Different specializations of the notion: document

5 CONTRIBUTION OF THE LANGUAGE DefOnto TO STRONG COUPLING

In this section we go back to DefOnto to put ahead two important aspects of the language regarding our goal of strong cou-

[6] A french version of OntoPME can be consulted at URL : http://www.laria.u-picardie.fr/EQUIPES/ic/demo/onto-pme.html. OntoOrg will be on-line on July 2002.

pling. On the one hand, we present the power of expression offered by DefOnto to formalize the model of the organization §5.1. On the other hand, we describe the inferential services provided by DefOnto and the query language used to make the coupling between the Inference engine and the Writer §5.2.

5.1 Formalization in DefOnto of the model of the organization

In addition to the representation of generic concepts, or classes of objects, DefOnto allows to represent individual concepts, *i.e.* points of view on individual objects, and this capacity is used to formalize the model of the organization. Two original characteristics, which are not shared by other languages [17][7], confer to DefOnto a great power of expression for this purpose. They are illustrated in the representations presented in figure 5.

A first characteristic is the possibility to define relations on relations, which confers to DefOnto a large propositional power of expression. This possibility is used in the description of the object *#KE_team* to represent the complex proposition: "KE team takes part in A2C2 project with HEUDIASYC partner, since January 1st 2002".

A second important characteristic is the possibility to define meta-knowledge, which allows the definition of classes of concepts, propositions and entities of representation. It becomes thus possible to represent the following knowledge: "the fact that the KE team takes part in mounting XX007 project is a confidential information" (the concept *#confidential_information* is defined as a class of propositions); "The entity representing the document (Fortier, 2001) was put in the KB on October 5th 2001" (the property *#has_for_intrance_date_in_KB* bears on the entity of representation and not on the object).

We have just seen with these examples that DefOnto allows to represent a relatively complex model of the organization, that is assuredly an important point regarding our goal to manage knowledge at once in the knowledge model and in the documents.

```
(DefIndConcept #KE_team
   IsA [#research_team]
   ObjectProperties
      -> (#is_a_component_of) -> [#LaRIA]
      -> (#has_for_responsible) -> [#Gilles_Kassel]
      -> (#takes_part_in_project) –
         -> [#A2C2]
         -> (#with_partner) -> [#Heudiasyc]
         -> (#since) -> [#january_1st_2002],
      -> (#takes_part_in_mounting_project) –
         -> [#XX007]
         -> (#proposition_belongs_to) -> [#confidential_information],)
(DefIndConcept #(Fortier, 2001)
   IsA {[#LaRIA_internal_report], [#stage_report]}
   ObjectProperties
      -> (#has_for_publishing_date) -> [#september-1st-2001]
   EntityProperties
      -> (#has_for_intrance_date_in_KB) -> [#october-5th-2001])
```

Fig. 5. Definition of two individual concepts in DefOnto

5.2 Inference services of DefOnto

DefOnto provides a query language constituted of a range of filters types. Each filter type corresponds to a particular type of query bearing on contents of the KB, notably:

- To return the explicit extension of a concept of the ontology, for example to return all internal reports: [#internal_report *x].
- To compute the extension of a concept of the ontology taking into account the ontological knowledge, for example to determine all the internal reports taking into account the fact that an activity report is an internal report: (can-infer-than [#internal_report *x]).
- To determine if an object explicitly (resp. implicitly) belongs to the extension of a concept of the ontology, for example to determine if (Cormier *& al.*, 2002) is an internal report: [#internal_report #(Cormier & al., 2002)], or (can-infer-than [#internal_report #(Cormier & al., 2002)]).
- To determine the set of linked objects to a given object according to a given relation, for example to determine who are the authors of (Cormier & al., 2002): [#has_for_author #(Cormier & al., 2002) *y].

Theses queries are transmitted to the Query module by the Writer and are evaluated by the Inference engine. A request of the user, for example, find all internal reports published from a given date, can correspond to a conjunction of filters. The role of the Query module is also to integrate results of the evaluation of different filters.

[7] Comparisons of DefOnto with other languages of representation (e.g., LOOM and OIL) are gathered on site: http://www.laria.u-picardie.fr/EQUIPES/ic/LangComp/

6 REALIZATION OF AN OM PROTOTYPE

Within the framework of our project PME (project of team memory), we have developed an OM prototype. This one is composed of two knowledge books using the same KB. A first book, accessible on Internet[8], presents the KE team of LaRIA and more widely the KE community in France and abroad by presenting teams, projects and documents, of reference. A second book, only accessible on a team's intranet, constitutes a work tool for the team. In addition to the information available in the first book, it permits to edit documents with restricted diffusion (work notes, reports of meetings, etc.) and indicates more detailed information on the team's projects.

Such a book (*cf.* figure 6) consists in two parts: the left part corresponds to the visualization of a table of contents and the right part corresponds to the visualization of the contents of the sections. The latter corresponds to a partial view of the organizational model generated by the Writer.

The table of contents is made up of a set of ordered themes. For example, the editor of the book (accessible on the Internet) has estimated that the presentation of the KE team of LaRIA should begin with the general presentation of the team and should continue with a presentation of its members, then of its partners. The themes can be broken down in sub-themes. The theme "Its projects", in our example, is broken down in two sub-themes: "internal projects" and "projects in collaboration".

A navigation in the table of contents allows the user to select a theme. The activation of the theme generates the construction of the corresponding view. This view corresponds to the structured description of objects[9].

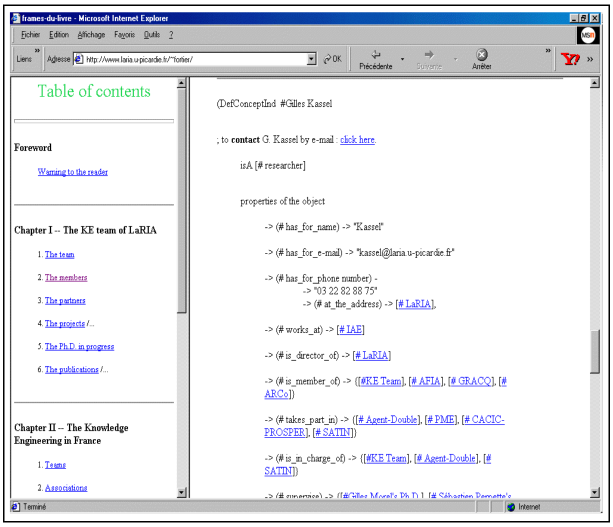

Fig. 6. Visualization of the knowledge book

In our example (*cf.* figure 6), the contents in the right window correspond to the results of the activation of the theme "its members". After the Writer retrieved all the necessary elements, he sends a request to the Request component to obtain the objects and the properties to show to the consultant. Then, he writes a XML file with these results and sends it to the Interface which can easily process it to adapt the order of presentation. This possibility is interesting only for a list of objects. The interface can change the order of presentation according to the user' choice. In our example, the Interface shows a response to the consultant which consists in an ordered list of descriptions of the members of the team; it begins with the presentation of the team leader. Then the permanent members are presented. The Ph.D. students and the associated members terminate the presentation. For each member, his/her name and his/her address are first indicated, then the projects in which he/she participates, and, if they exist, his/her responsibilities. The consultant may prefer a presentation according to the project in which the person participates rather than according to the administrative function. Concerning documents, the consultant can prefer to order the presentation by date, by author or by subject.

The objects' descriptions mention different entities: concepts and relations which are part of the ontology (e.g.: researcher, supervisor), and other objects which structure the model of the organization. Links on these entities (*cf.* figure 6) allow the user to get other knowledge. They allow to see a definition of concept or relation, or the description of another object. When following the links, the user can in particular reach the description of the set of referenced documents.

Finally, some actions are allowed on certain objects with the purpose of interacting with the entities of the physical world that the objects model. For example, it is possible to edit documents for which we have an electronic version with the help of the document administrator which is in charge of maintaining the document' base or to contact someone with the mail. In our example in figure 5, we can contact Gilles Kassel in activating the link "click here".

7 RELATED WORKS

In this section we compare our memory architecture to other architectures relying on a KB and DB coupling.

We find in the (KA)² project [2] and its recent continuations [24] a memory architecture close to ours: the KB is made with web pages annotations, and the ontology is used both to model annotations and to infer implicit knowledge during the queries. The replies consist however in objects lists, not in structured objects descriptions. Moreover, these replies don't take a user profile into consideration.

The CoMMA European project [12] mainly aims at evaluating the contribution of a multi-agents approach to design and to implement OM. It exploits emergent web technologies (XML, RDF(S)) for the annotations and ontology specification. With the translation of these RDF(S) specifications to conceptual graphs and the use of CORESE inference engine [8], we find again an architecture close to ours. Moreover this project has recently led to an expansion of RDF(S) to extend the expression capability for the ontology and annotations specification [9]. Nevertheless, as in the (KA)² project, the query replies only consist in elementary objects lists.

---

8 http://www.laria.u-picardie.fr/EQUIPES/ic/demo/livre-ic.html

9 Actually, the prototype that implements our software architecture directly uses the files which compose the KB formalized in DefOnto. A new version integrating the functionalities of the "writer" module will be soon accessible.

In the ScholOnto and myPlanet projects [6][14], which rely on the notion of documents enrichment [19], the KB contains knowledge to facilitate the documents access and their contents interpretation. The KB/DB coupling is therefore used, as in $(KA)^2$ and CoMMA, to make the information retrieval easier by using knowledge models.

In our approach, the model of the organization plays the same role of documentation contextualization, but it is besides exploited for itself, in a strong coupling perspective. This exploitation goes through the addition of a diffusion mechanism for the modeled knowledge, which takes the form of a generation of customized virtual documents.

8 PROSPECTS

The works we have just exposed are going into different directions.

A first version of the set of software modules, which constitutes our OM architecture, is currently being built. The multi-agents approach, already used in different OM projects [1] [12], seems well suited for the implementation of such software architectures. We have chosen to use the JADE platform [3] as in the CoMMA project.

Currently, the presentation models are linked to the objects in the book skeleton, which leads to duplicate these models and also to incorporate the user' profiles in these models. To overcome these limitations, we plan to adopt a knowledge-based approach for the Writer which will dispose of generation methods for the elaboration of structured descriptions. Such an approach will provide us more flexibility to take into account the users profiles.

At the same time we plan to carry out other experiments and capitalize on the experience both in the software architecture and in the associated OM development method.

9 REFERENCES

[1] BAEK S., LIEBOWITZ J., PRASAD S.Y., GRANGIER M. (1999). Intelligent Agents for Knowledge Management - Towards Intelligent Web-Based Collaboration within Virtual Teams. In J. LIEBOWITZ Eds., Knowledge Management Handbook, CRC Press, p. 11-1, 11-23.

[2] BENJAMINS V.R., FENSEL, D., DECKER, S., GÓMEZ-PÉREZ A. (1999). $(KA)^2$: Building Ontologies for the Internet: a Mid Term Report. In *Proceedings of the International Workshop on ontological Engineering on the Global Information Infrastructure*, Dagsthul Castle (Germany), p. 1-23.

[3] BERGENTI F., POGGI A., RIMASSA G. (2000). Agent Architecture and Interaction Protocols for Corporate Memory Management Systems. In *Proceedings of the ECAI'2000 Workshop: Knowledge Management and Organizational Memories*, p. 39-47.

[4] BIÉBOW B., SZULMAN S. (1999). TERMINAE: A linguistic-based tool for the building of a domain ontology. In *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management: EKAW'99*, Dagstuhl (Germany).

[5] BUCKINGHAM SHUM S. (1997). Balancing Formality with Informality: User-Centred Requirements for Knowledge Management Techniques. In *Proceedings of the AAAI Spring Symposium Workshop "AI in Knowledge Management"*, Stanford University, p. 127-133.

[6] BUCKINGHAM SHUM S., MOTTA E., DOMINGUE J. (2000) ScholOnto: An Ontology-Based Digital Library Server for Research Documents and Discourse. *International Journal on Digital Libraries*, 3(3), p. 237-248.

[7] CLARK P., THOMPSON J., HOLMBACK H., DUNCAN L. (2000). Exploiting a Thesaurus-Based Semantic Net for Knowledge-Based Search. In *Proceedings of the Twelfth Conference on Innovative Applications of Artificial Intelligence: IAAI-2000*, Austin (Texas, USAA), p. 988-995.

[8] CORBY O., DIENG R., HEBERT C. (2000). A conceptual Graph Model for W3C Resource Description Framework. In *Proceedings of the 8th International Conference on Conceptual Structures (ICCS'2000)*, August 2000, Darmstadt, Germany.

[9] DELTEIL A., FARON C., DIENG R. (2001). Extension of RDFS based on the CG formalism. *To appear in the Proceedings of ICCS'01*, August 2001, Stanford (USA), p. 275.

[10] FERNÁNDEZ-LÓPEZ M., GÓMEZ-PÉREZ A., PAZOS-SIERRA J., PAZOS-SIERRA A. (1999). Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems*, January/February, p. 37-46.

[11] FORTIER J.-Y. (2001). Construction d'une ontologie pour gérer la documentation d'un centre de recherche. Application au Centre Commun de Recherche d'EADS. Internal report 2001-08 of the LaRIA, University of Picardie Jules Verne, October 2001. *Title in english: Design of an ontology for managing the documentation of a research center - Application to the Common Research Center of EADS.*

[12] GANDON F., DIENG R., CORBY O., GIBOIN A. (2000). A Multi-Agent System to Support Exploiting an XML-based Corporate Memory. In *Proceedings of the 3rd International Conference on Practical Aspect of Knowledge Management (PAKM'2000)*, p. 10-1, 10-12.

[13] GOLEBIOWSKA J., DIENG-KUNTZ R., CORBY O., MOUSSEAU D. (2001). Exploitation of Ontologies for Building A Vehicule Project Memory. In *Proceedings of the IJCAI'01 Workshop on Knowledge Management and Organizational Memories*, Seattle (USA), p. 49-58.

[14] KALFOGLOU Y., DOMINGUE J., MOTTA E., VARGAS-VERA M., BUCKINGHAM SHUM S. (2001). *myPlanet*: an ontology-driven Web-based personalised news service. In *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, Seattle (USA), p. 44-52.

[15] KASSEL G. (2002). OntoSpec : une méthode de spécification semi-informelle d'ontologies. In Actes des Journées Francophones d'Ingénierie des Connaissances: IC'2002, Rouen, May 2002. Title in english : OntoSpec: a method for the semi-informal specification of ontologies.

[16] KASSEL G., ABEL M.-H., BARRY C., BOULITREAU P., IRASTORZA C., PERPETTE S. (2000). Construction et exploitation d'une ontologie pour la gestion des connaissances d'une équipe de recherche. In *Actes des Journées Francophones d'Ingénierie des Connaissances: IC'2000*, Toulouse, p. 251-259. *Title in english: Design and implementation of an ontology for managing knowledge of a research team.*

[17] KASSEL G., BARRY C., ABEL M.-H. (2000). Programmer au niveau connaissance en Def-*. In J. CHARLET, G. KASSEL, D. BOURIGAULT Eds., *Ingénierie des Connaissances, évolutions récentes et nouveaux défis*, Eyrolle, p. 145-160. *Title in english: Programming at the knowledge level in Def-*.*

[18] KASSEL G., PERPETTE S. (1999). Co-operative ontology construction needs to carefully articulate terms, notions and objects. In *Proceedings of the International Workshop on ontological Engineering on the Global Information Infrastructure*, Dagsthul Castle (Germany), p. 57-70.

[19] MOTTA E., BUCKINGHAM SHUM S., DOMINGUE J. (1999). Ontology-Driven Document Enrichment: Principles and Case Studies. In *Proceedings of the Twefth Workshop on Knowledge Acquisition, Modeling and Management: KAW-99*, Banff (Canada).

[20] O'LEARY D.E. (1998). Using AI in Knowledge Management: Knowledge Bases and Ontologies. *IEEE Intelligent Systems*, May/June 1998, p. 34-39.

[21] RANWEZ S., CRAMPES M. (1999). Conceptual Documents and Hypertext Documents are two different Forms of Virtual Document. In *Proceedings of the Workshop on Virtual Documents, Hypertext Functionality and the Web at the 8th International World Wide Web Conference*, May 1999, Toronto (Canada), p. 21-27.

[22] ROUSSEY C., CALABRETTO S., PINON J.-M. (2001). SyDoM: A multilingual Information Retrieval System for Digital Libraries. In

Proceedings of the 5th International ICCC/IFIP Conference on Electronic Publishing: ELPUB'2001, Canterbury (UK), p. 150-160.
[23] SCHREIBER G., AKKERMANS H., ANJEWIERDEN A., DE HOOG R., SHADBOLT N., VAN DE VELDE W. & WIELINGA B. Knowledge Engineering and Management; The CommonKADS Methodology. MIT Press, 1999.
[24] STAAB S., MAEDCHE A. (2001). Knowledge Portals, Ontologies at Work. *AI Magazine*, Summer 2001, p. 63-75.
[25] USCHOLD M., KING M., MORALEE S., ZORGIOS Y. (1998). The Enterprise Ontology. *Knowledge Engineering Review*, vol 13(1).