A CommonKADS Expertise Model
Web Server

Olivier Corby & Rose Dieng

INRIA
2004 route des lucioles, BP 93
06902 Sophia Antipolis Cedex France

Olivier.Corby@sophia.inria.fr
Rose.Dieng@sophia.inria.fr

Abstract: After showing the possible interest of CommonKADS method for building some kinds of knowledge-based corporate memories, this paper describes a CommonKADS Expertise Model Web Server. It enables the distribution of CommonKADS Expertise Models (EM) on the Internet or on an Intranet by means of a Web server. The software offers several functions such as:

- Hypertextual navigation within EM
- Information retrieval in EM by means of a query language
- Hypertextual navigation between EM and expertise documents
- Generation of graphic representation of EM

The system can be viewed as a CommonKADS generic library server on the Web but also as a knowledge-based component for corporate memory management. The system is built on top of Cokace which is an environment for CommonKADS CML expertise model development.

Keywords: Knowledge Web Server, Knowledge distribution, CommonKADS

1. Introduction

1.1 Corporate Memory

There is an increasing industrial interest in the capitalization of knowledge (i.e. both theoretical knowledge and practical know-how) of groups of people in an organization, such groups being possibly dispersed geographically. The coherent integration of this dispersed knowledge in a corporation is called *«corporate memory»*. It has the objective to *«promote knowledge growth, promote knowledge communication and in general preserve knowledge within an organization»* [Steels 93]. As Tourtier [Tourtier 95] we can distinguish:

- corporation memory (referential, documents, tools, methods),
- society memory (organization, activities, products, participants such as customers, suppliers, sub-contractors),
- individual memory (status, competencies, know-how, activities),
- project memory (definition, activities, history, results).

By the same way, [Pomian 96] distinguishes: technical memory, project memory and organizational memory.

The building of the corporate memory can rely on the following steps:

1. *Detection of needs in corporate memory:* it can be based on opportunity study allowing to determine strategic knowledge to be kept in the corporate memory [Grunstein 95], or on exploitation of enterprise models [Fraser 94, Uschold 96]. This phase may help to determine the potential users of the corporate memory, and perhaps the possible modes of exploitation useful and adapted to their work environment.
2. *Construction of the corporate memory:* the adopted techniques depend on the sources from which the corporate memory can be built (specialists, written or electronic documents such as reports or technical documentation, existing databases or case libraries, dictionaries), and on the nature of the needed corporate memory according to the intended users (written documents vs electronic documents, documentary system, formal knowledge vs informal knowledge, knowledge base vs case base). Both construction of the corporate memory contents, and adequate organization of this corporate memory, must be tackled. As emphasized during KAW'96 track on «Corporate Memory and Enterprise Modeling», a corporate memory is of course different from a knowledge-based system: it may be a written document making explicit the enterprise adequate members' knowledge, that had never been yet elicited and modeled [Dieng 96]; it may also be implemented in a *computational memory*, such as an intelligent documentary system [Poitou 95, Ballay 96], a knowledge base [Dieng 96], a case-based system [Simon 95, 97], a combination of documents, knowledge bases and case bases...
3. *Diffusion of adequate elements of the corporate memory to the adequate members of the enterprise:* this distribution may be passive or active [VanHeijst 96], as either the user can search himself needed information, or knowledge distribution can be systematically decided and taken in charge by an adequate service of the enterprise. Individuals and organizations can take advantage of the remarkable possibilities of

access to data, to information and to knowledge provided by the Internet. Knowledge diffusion can for example exploit the possible access to the Internet or to an Intranet inside the enterprise. It can thus rely on a knowledge server on the Web or on publication on the Web [Euzenat 96].

4. *Use of the corporate memory* by adequate members of the enterprise: in all cases (documentary system, knowledge base, case base, Web-based system, etc.), we must notice the importance of information search, if possible adapted to the users' needs, their activities and their work environment.
5. *Maintenance and evolution of the corporate memory:* problems linked to addition of new knowledge, removal of obsolete knowledge, coherence problems underlying a cooperative extension of the corporate memory, must be tackled.

1.2 Interest of CommonKADS for a Knowledge-based Corporate Memory

Let us consider an enterprise interested in building a corporate memory with knowledge-based components. In this case, it can be interesting to *exploit the results of research on knowledge engineering and on knowledge acquisition, and adapt or extend them to knowledge capitalization and management*. Therefore the use of adequate extensions of CommonKADS [Schreiber 94a, Breuker 94], a standard knowledge acquisition methodology for knowledge-based system development, can be thought out. Even though CommonKADS was not primarily dedicated to corporate memory building, several features of CommonKADS can be interesting in this framework [Kingston 94, VanderSpeck 94]: in particular, some models offered by CommonKADS (organization, task, agent, communication, and expertise models) give a good basis for knowledge-based corporate memory. CommonKADS principles are different from those methods aimed at knowledge capitalization such as REX that is based on elements of experience [Malvache 93] or MKSM [Ermine 96]. CommonKADS organization model allows to model the organization in which the future knowledge-based corporate memory will be integrated. Task model allows to describe the activities in the enterprise. Agent model allows to model the members of the enterprise (i.e. «human agents») or the software (i.e. «computational agents») that realize tasks in the enterprise. Communication model helps to describe exchanges among such agents for task realization. Expertise model helps to describe problem-solving knowledge and the know-how of an agent involved in task realization. All such models seem useful for the different kinds of corporate memory, distinguished in Tourtier's and Pomian's typologies:

- Several aspects of CommonKADS expertise model are useful for describing elements of a *corporation memory*: the notions of ontology and domain model offered in CommonKADS expertise model are helpful for describing the referential needed in a corporation memory, while CommonKADS problem-solving methods (PSM) may help to describe the methods needed in a corporation memory.
- Several aspects of a *society memory* can be described thanks to CommonKADS organization model, task, and agent models.
- CommonKADS expertise and agent models are interesting for *individual memory.*

- The description of activities needed in *project memory* can partly rely on CommonKADS task and communication models. Adequate extensions can help to adapt CommonKADS to other aspects of project memory [Tourtier 95].
- CommonKADS task, agent and expertise models are useful for *technical memory,*
- CommonKADS organization, task, agent and communication models are useful for description of some aspects of *organizational memory.*

Some extensions of CommonKADS for corporate memory building are proposed in [VanHeijst 96]. CommonKADS may be integrated in a method for building a corporate memory [Dieng 96].

CommonKADS proposes the CML Conceptual Modeling Language [Schreiber 94b] to describe expertise models. Several environments exist for CML: *Void* developed in the Kactus Esprit project [Schreiber 94c], *Cokace* developed in the Acacia project at the Inria [Corby 96].

1.3 Aim of the Paper

In this purpose of *diffusion of a corporate memory contents to the enterprise adequate members*, the *exploitation of the Internet or of an Intranet inside the enterpris*e may be interesting. Such knowledge may be kept implicit in documents but it may also be formalized through knowledge bases or through expertise models. Our work presented in this paper is situated in the framework of *knowledge servers* when knowledge is described through expertise models expressed in the CommonKADS methodology.

In this context we have developed a CommonKADS Expertise Model Web Server, hereafter called *WebCokace.*

The main purpose of WebCokace is to provide the user with access to a CommonKADS expertise model repository, by means of a Web server. The user connects to the Web server with his preferred browser and has access to expertise in the standard hypertextual Web fashion. In such a way, expertise models are accessible to a community of users on the Internet or on an Intranet. WebCokace can be used by knowledge engineers to browse a generic modeling library, by experts to check the expertise model of a knowledge based system or by members of an enterprise to consult a knowledge-based corporate memory.

WebCokace offers hypertextual navigation, a query language and search functions, annotation with documents and graphic generation. After describing the needs of an expertise model user, we will detail the functions of our expertise model server. Then we will present the implementation and the application of the WebCokace system. Last, we will compare our approach to related research and discuss its advantages and weaknesses.

2. An Expertise Model Web Server

2.1 CommonKADS Expertise Model

The CommonKADS Expertise Model aims at modeling a problem solving behaviour from a Knowledge Level perspective. It is structured in three layers: the domain level and the task and inference levels.

The domain level provides ontology modeling, e.g. conceptualizing and formalizing a domain by means of concepts, expressions and relations. According to the ontologies, the domain level also describes knowledge chunks called *domain models.* A domain model describes knowledge on part of the domain, from a specific point of view. For example, one can describe a causal model of a system, a behavioural model, a structural decomposition, etc.

The task and inference levels aim at modeling reasoning and problem solving. These levels are supposed to be as much as possible independent of the domain: they should be described in a task specific way. For example, if one models electrical diagnosis, the task and inference levels should talk mainly about diagnosis and not (much) about electrical matter. Tasks and inferences use a task oriented ontology, the terms of which are called *roles.* For example diagnosis talks about symptoms, fault, hypothesis, candidate, solution, measure, etc.

Tasks describe the goals of the problem solving agent and the actions that can lead to reach the goals, including goals and tasks decomposition. Basic tasks are implemented by means of inferences. An inference is close to the notion of inference rule in logic. It specifies the basic reasoning steps that are feasible at the domain level. It describes the type of knowledge that it is possible to deduce given inputs and domain models. Inferences also describe the mapping between task oriented roles and domain level entity types.

2.2 Potential Needs of an Expertise Model Server User

A user that accesses an expertise model via a knowledge server may be interested in several kinds of use:

- Visualize this model through different formats: a) through a synthetic summary of the contents of this expertise model (for example through an index), (b) in a graphic format (cf. the pure CML textual format may be considered as not enough friendly by some end-users).
- Visualize only some subparts of this model, such subparts being selected according to adequate criteria.
- Visualize a given part of the model at different detail levels.
- Understand this model, for example by studying the associated documentation: the user may examine the source documents from which this model was built (e.g. experts' interviews, technical documentation, scientific articles...), but also specific documentation built on purpose for this expertise model or some bibliographic references presenting other view than this expertise model, etc.

- Compare several models by visualizing them simultaneously or even making simultaneous search on several models or making computation on such models. It can be interesting for a user in a company that needs to explore the corporate memory parts corresponding to viewpoints of several experts exploiting different problem solving methods for task realization. For example, in a hospital network, the models may describe different therapeutic protocols, and a physician could be interested in comparing the protocols adopted by his colleagues of the same hospital or of other hospitals.
- Build his own expertise model by choosing other models to import thanks to the knowledge server. This can be interesting for a user who wants to extend the corporate memory or to adapt a specific version of the corporate memory for his own use.

Therefore the functions of WebCokace aim at fulfilling some of such needs. We distinguish two important phases:

- building the knowledge-based corporate memory with Cokace,
- consulting the corporate memory via the Web.

When the knowledge-based corporate memory builders (e.g. knowledge engineers, or adequate members of the enterprise) use Cokace, they can give some indications in the expertise models in order to prepare the future consultations. The knowledge server users (i.e. members of the enterprise consulting the knowledge server for their task or for extending the corporate memory) can then consult the models through WebCokace.

In the next sections, we will describe the WebCokace system. We will indicate what is prepared during the phase of corporate memory development by the corporate memory builder (that we will call knowledge engineer) and what is exploited during the corporate memory consultation by the user of the knowledge server. We will first focus on the hypertextual navigation and the advanced search offered to the user, then we will describe the exploitation of annotation of expertise models by documents and last, we will describe the implementation of the WebCokace system and its applications.

2.3 Hypertextual Navigation by the Knowledge Server User

First of all, the user accesses the knowledge server by means of a Web browser at an adequate URL in the enterprise (for example at the INRIA: `http://www.inria.fr/acacia/Cokace`).

A user interface proposes to the user to browse generic libraries and target applications by means of the following functions:

- Access to source documents
- Index of the expertise models
- Graphic views on expertise models
- Advanced search features

Figure 1 shows WebCokace user interface home page.

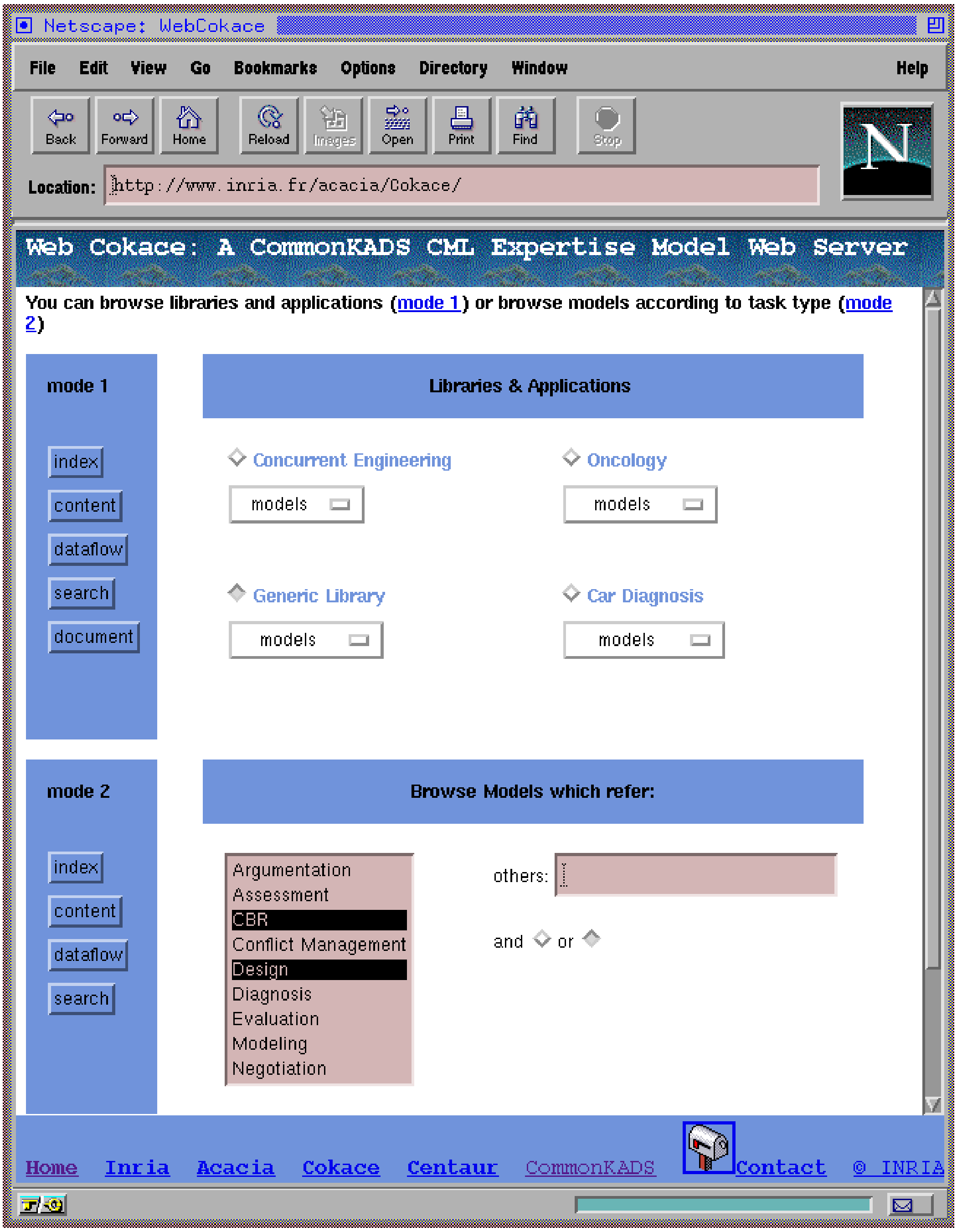

FIGURE 1. WebCokace User Interface Home Page

2.3.1 Navigation in Expertise Models

WebCokace provides hypertextual navigation through CommonKADS expertise models. The Expertise Models are managed by a Cokace server which is connected to a Web server by means of a CGI interface.

The Cokace environment [Corby 96] enables to develop and validate CommonKADS expertise models in CML. It provides a structured editor and a type-checker. EM are internally manipulated by means of an object oriented abstract representation (so-called abstract syntax tree). This representation enables high level and abstract operations on expertise models, such as search by pattern matching and pretty-printing. We wrote several pretty-printers allowing to display an expertise model through different formats (CML source, HTML, text only, graphics, etc.).

We have extended Cokace so that it behaves like an expertise model server accessible by a client. The client emits a request and Cokace delivers information in response to the request. In our case, the client is a Web server as shown in Figure 2.

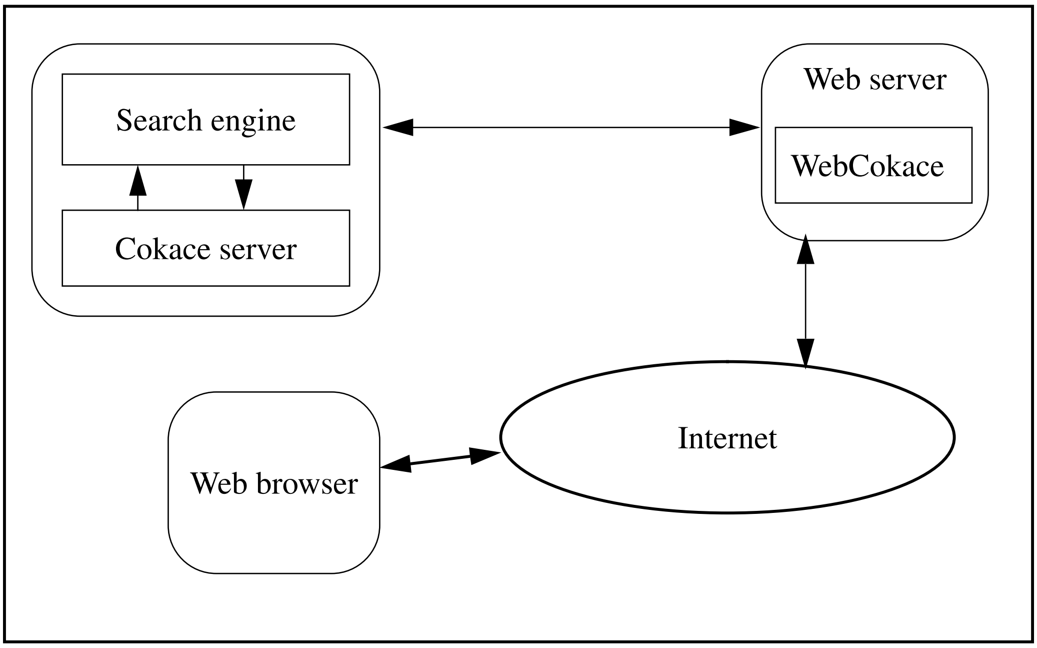

FIGURE 2. WebCokace Architecture

The CGI interface routes queries from the Web server to Cokace and routes answers back from Cokace to the Web server.

On request, Cokace extracts data from expertise model abstract representations. Then, Cokace transforms abstract data into textual or graphic format by means of pretty-printers. Furthermore, textual representation is *dynamically annotated* by Cokace with HTML hypertext links to symbolize relations between KADS entities.

Once present in the Web browser of the knowledge server user, the hypertext links are activated and the system *simulates hypertextual navigation in CommonKADS entities.* When the user clicks on a hypertext link, he activates a CGI script which sends a request to the Cokace server. The answer to the user's request is another HTML page with new hypertext links to related CommonKADS entities, and so on.

The user can navigate from one entity to another: concept to super concepts, task to subtasks, inference role to domain reference, etc.

Here is an example where the ‘‘sub-type-of’’ field is represented in the system by means of an URL to the related concept:

```
concept Patient_prognostic_factor ;
sub-type-of: <a href="/cgi-bin/select.pl?
  request=ck-select&from=concept&name=Prognostic_factor">
  Prognostic_factor</a> ;
end concept Patient_prognostic_factor;
```

When the user clicks on the “sub-type-of” link, it leads to the following concept:

```
concept Prognostic_factor ;
end concept Prognostic_factor;
```

Cokace generates hyperlinks for concept references (super concept, differentiation-of, slot access in expressions and in domain model, etc.), domain reference in relations and in inference sorts, in subtasks, etc. Furthermore, Cokace automatically generates hypertext links for subconcepts, task super-task and inference super-tasks (the tasks which call an inference).

This *dynamic creation of hyperlinks* by Cokace is an important feature: it allows a dynamic generation of adequate HTML documents during consultation time, in answer to the user’s requests.

Static links.

During expertise model development in Cokace, the CML entities (concept, task, inference, etc.) may explicitly be annotated by the knowledge engineer with URL references to other CML entities. In that case, at consultation time, the URL become active when the annotated entities are loaded in the Web browser of the knowledge server user.

For example, a task may reference a problem solving method in a comment:

```
task SolveConflict
   /* <a href="/cgi-bin/select.pl?request=ck-select&
      from=psm&name=CBR">Alternative PSM</a> */
```

2.3.2 Presentation Formats

WebCokace can generate *indexes* for coarse grain entities (domain, ontology, task, inference). The knowledge server user can then have a summary of the content of given expertise models.

WebCokace can also supply CML source versions of CommonKADS expertise models, thus enabling a user to get and reuse modeling components from a library. The CML source version does not contain generated hypertext links.

As the Cokace environment manages an abstract representation of expertise models and provides a formalism and an engine to develop pretty-printers, it is easy to design new presentation formats or to extend existing pretty-printers.

2.4 Advanced search

In addition to index and dataflow, WebCokace provides advanced search functions in order to examine expertise models in more details. For these functions, we decided to rely on standard information retrieval techniques and choose the OQL approach of the ODMG Object Query Language [Cattel 96]. In order to perform search according to OQL like requests, CML entities are considered as objects with slots. For example inferences have input-roles and output-roles slots, concepts have name and super slots, etc.

Search requests rely on underlying select/from/where patterns. But, as an ergonomic evaluation of the user interface showed that knowledge engineers had difficulties in understanding such OQL like statements in the context of CML, they have been embedded in specialized search panels as showed in Figure 3.

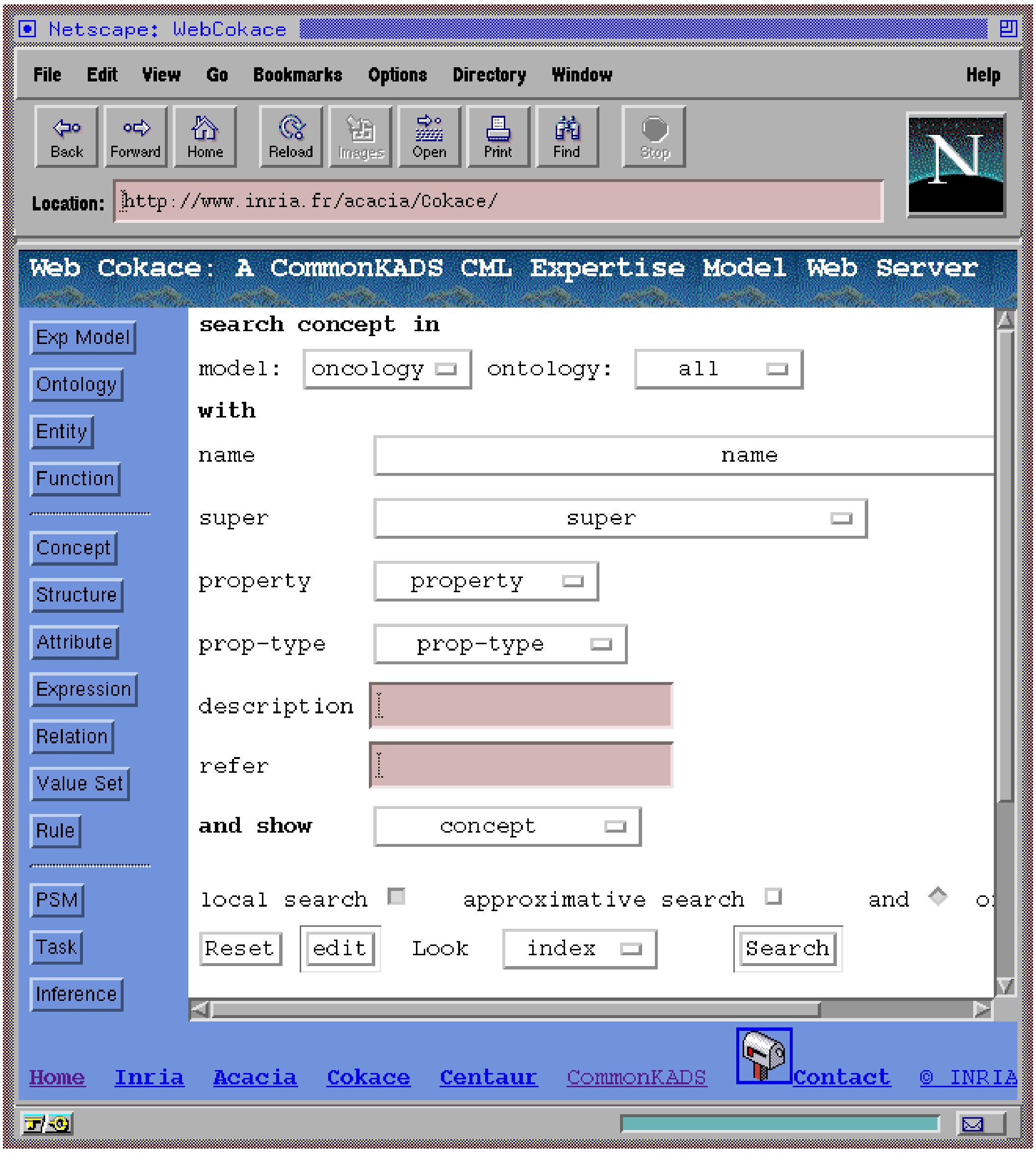

FIGURE 3. Advanced search interface

We adopted the following syntax to state search requests:

```
search concept in <model>
with
name = <name>
super = <supertype>
...
and show <property>
```

We have implemented a prototype of such a query language, without variables, and plan to extend it to match a subset of OQL. We have also implemented the engine which searches the expertise models in order to answer the queries.

The results of a search are CommonKADS entities that match the query. The entities are pretty-printed in the HTML format, with possible HTML links towards related CommonKADS entities. Hypertext navigation can then follow a search by means of the query language.

2.5 Annotation with Documents

In an enterprise, many documents can play an important role for building a corporate memory: technical documentation, reports (e.g. project reports - interesting in the framework of a project memory), transcription of experts' interviews, etc. If such documents are put in an electronic format, a knowledge server inside the enterprise must have an access to such documents.

A knowledge server can help to have access to: other servers relevant for the considered area, bibliographic references (cf. articles), the different kinds of documents cited above. If an adequate part of such documents is modeled through expertise models or through a knowledge base, it is interesting to keep a link between the expertise model (or the knowledge base) and the document.

CommonKADS expertise models are usually built on the basis of documents such as expert interviews, articles, technical reports, etc. During the construction of the corporate memory, the expertise model definition can be annotated by the knowledge engineer with references to such documents by means of URL. During navigation through the knowledge server, when loaded into the user's browser, the hypertext link may be activated by the user in order to have an access to the referenced documents. A document may be *a document in the HTML format, a Web server, a Wais server or anything accessible by an URL.*

Conversely, the knowledge engineer can annotate an HTML source document with references to CommonKADS entities by means of URL, in fact appropriate references to CGI scripts. When browsing the documents, these URL become active: the user can click on them and thus have a transparent access to related CommonKADS entities. The user has at his disposal active source documents with direct access to expertise models.

2.6 Graphical Representation

The default presentation format is text, but WebCokace can also generate *graphic representations for CML entities*. It generates graphic representations for concept hierarchies and for task sub-task hierarchies.

When the knowledge server user clicks on a hierarchy node, he activates an URL that leads to the corresponding CML entity definition. The user can thus have another, more concise, point of view on the expertise model by means of graphical representations *dynamically generated during consultation tim*e. Figure 4 below shows an example of such a graphical concept hierarchy.

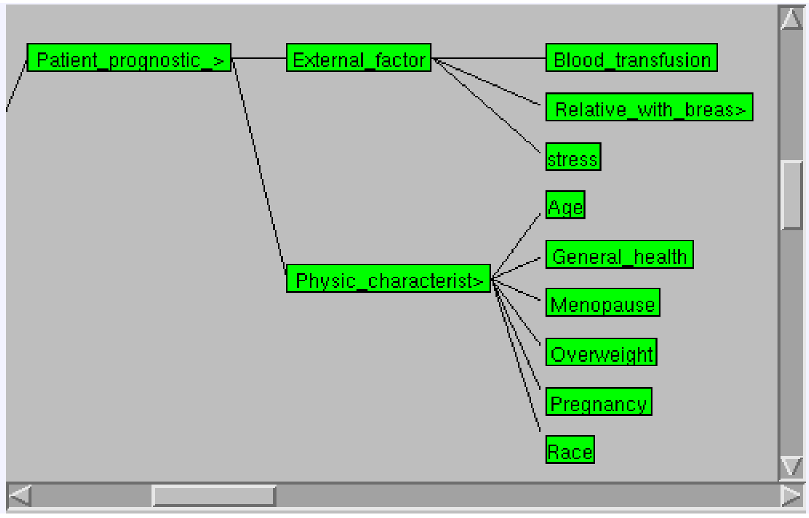

FIGURE 4. Graphic Display of a Concept Hierarchy

WebCokace also generate graphic representations of task dataflow as shown in Figure 5. Nodes are active and when clicking on a subtask, its dataflow is generated. It is possible to browse graphically task dataflows following the task decomposition into subtask.

3. The WebCokace System

This section presents the system implementation and its applications.

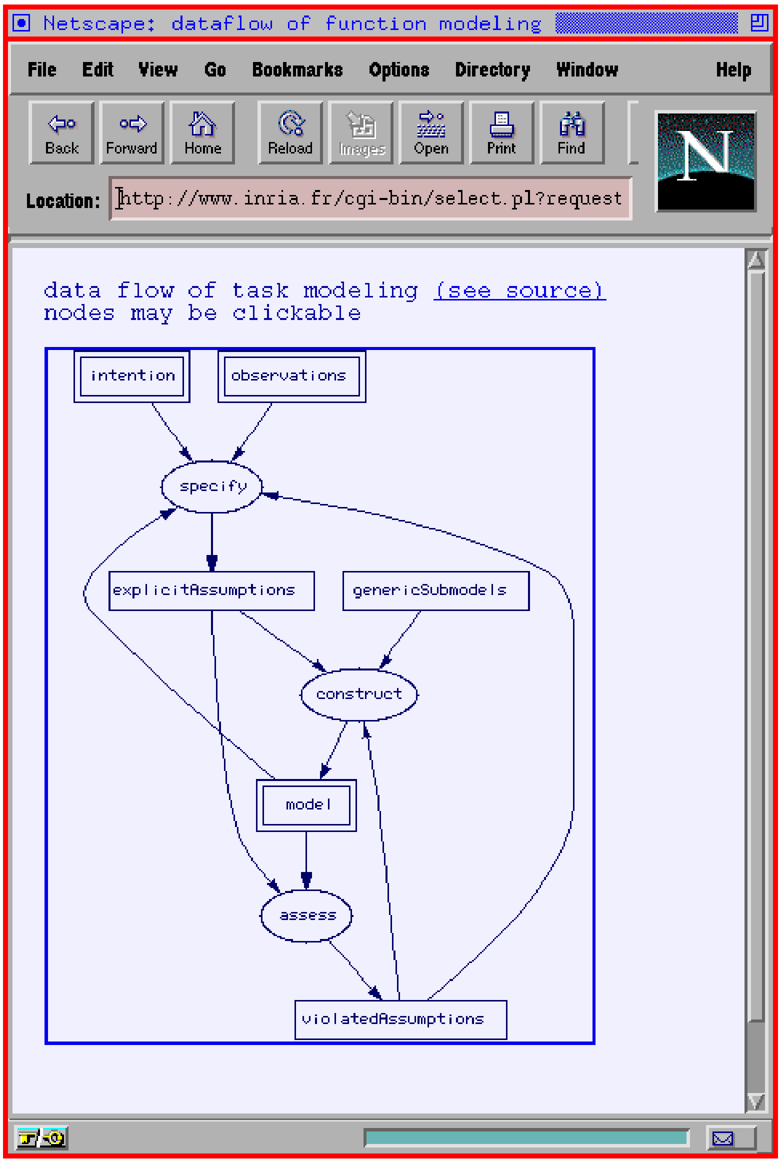

FIGURE 5. Dataflow of the *Modeling* Task

3.1 Implementation

As indicated above, the CommonKADS expertise model Web server is implemented as a connection between a Cokace server and a Web server by means of a CGI interface.

Cokace is itself built on top of the Centaur programming environment generator. Centaur has been developed at the Inria in the Croap team [Borras 88]. Given the specifications of a language, Centaur generates an integrated set of tools such as *parsers, structured editors, pretty-printers, type-checkers and interpreters*. Cokace was generated by specifying the CML language in Centaur [Corby96].

Furthermore, the underlying Centaur system provides a set of generic pattern matching functions on abstract syntax trees. It also provides a pretty-printing specification language, called PPML, and a powerful pretty-printing machine. So, we specified easily a standard CML pretty-printer and derived a HTML pretty-printer.

This architecture is generic and can be reused to build servers for other languages since Centaur is a programming language environment *generator*.

3.2 Application

WebCokace has been applied to design a concurrent engineering modeling library in the GENIE project [Matta 96a,b]. This project aims at proposing a support for technical memory in the framework of a concurrent engineering task, in aeronautics. The application contains ten expertise models and 250 entities (concept, task, inference and PSM) which represents 3000 lines of verified CML code.

WebCokace has also been applied to design an oncology expertise model server during a Galileo cooperation with DIST (University of Genoa) [Sacile 95, 96]. It represents 1000 lines of verified CML code. This application was used at INRIA and at DIST, but it remains to be tested by oncologists of a hospital or of a laboratory.

3.3 Evaluation

We have conducted an ergonomic evaluation experiment with a cognitive ergonomist. Experiments have been conducted with knowledge engineers to assess the utility and the usability of WebCokace for the knowledge acquisition task. Results showed that the tool provided interesting searching and browsing functions, and also hypertext navigation and graphic views. The problem were that they were hardly accessible and that the user interface did not fit very well the knowledge modeling activity.

Among others, critics were:

- there were no means to have a quick idea of the content of the server, and no way to access to models according to task types (e.g. design models, diagnosis models, etc.).
- the lack of accessibility of interesting functions: two many steps were necessary to access functions. The necessity to distinguish direct access functions such as graphic views of models and advanced search functions that can be attained in more steps.

Consequently, we have modified the user interface according to the ergonomics recommendations by taking more into account the user activity, e.g. information retrieval for knowledge management.

3.4 Methodology of Use of WebCokace

Let us consider an enterprise interested in building a corporate memory with knowledge-based components. In this case, as we justified it earlier, the use of CommonKADS method that was originally aimed at knowledge-based systems development, can be interesting. In particular, CommonKADS expertise models are interesting for building a corporation memory (e.g. oncologist), an individual memory (e.g. case of retirement of a specialist), or a technical memory (e.g. aircraft landing gear design). If the analysis of the corporate memory needs in the enterprise shows the interest of a corporate memory with knowledge-based components, the corporate memory builder can choose to be guided by CommonKADS method. For the construction and verification of expertise models that will be part of the corporate memory, the corporate memory builder can exploit the Cokace tool. He can prepare the future consultation of the expertise models by indicating adequate links to documents (let us recall that many other hyperlinks, taking into account the user's request, will be dynamically generated during consultation time). The corporate memory builder can carry out several kinds of verification of the expertise models, in particular by simulating them.

For the consultation of the corporate memory via the Web, the corporate memory user can exploit WebCokace. He can visualize a summary of the contents of the corporate memory thanks to an index. He can choose specific parts of the corporate memory to visualize thanks to a textual form or through a graphic format. For a given expertise model, he can visualize selected subparts of this model, at several detail levels, with adequate formats (index, text, graphics). He can explore the source documents associated to this model. He can explore simultaneously several models, in particular if in the corporate memory several viewpoints were taken into account (for example, different therapeutic protocols used for the same cancer type in different hospitals of an hospital network considered as an enterprise).

This exploration of the corporate memory can also be useful for the corporate memory builder, in phase of extension of the corporate memory: in particular, he may need to verify whether the new elements to be added in the corporate memory are consistent with the already existing elements.

4. Related Work

More and more researchers focus on Distributed Knowledge Modeling over the Internet in particular by exploiting the Web:

- A corporate memory infrastructure relying on techniques of information search on the Internet is proposed in [Huynh 94].
- The Web can be exploited for publishing knowledge bases [Euzenat 96]. WebCokace allows a publication of expertise models through the Web, with, in addition, the possibility to make computations on such expertise models thanks to our Centaur-based implementation of Cokace.
- In [Farquar 96], the authors offer support tools for collaborative, distributed building of consensual common ontologies, shared by geographically distant groups. Such tools exploit the Web and allow to publish, browse, create and edit ontologies stored on an ontology server. The users can browse and retrieve ontologies in repositories; they can assemble, adapt and extend ontologies from repositories. Distant applications can use an Ontolingua server in order to learn about the vocabulary of an ontology and to ask requests. Even though the authors don't explicitly suggest an application for knowledge management, their work could be useful for building ontologies dedicated to different departments or sites of the same enterprise. Our expertise model server WebCokace also provides an ontology server, if such ontologies are described using CommonKADS framework (cf. the domain layer of an expertise model comprise an ontology part). But the use of WebCokace is not restricted to ontologies: domain models, inference structures, tasks and problem

solving methods are also parts of an expertise model and can thus be handled in our knowledge server. The edition of ontologies is performed within Cokace but, presently, it cannot yet be performed through WebCokace from a distant machine.

- In [Euzenat 96], the author presents a protocol for collaborative construction of a consistent knowledge base through the Web. It allows an incremental and concurrent construction of a knowledge base comprising formalized knowledge and annotations (texts, bibliography, images,...). It offers a support for expressing, annotating and handling knowledge and for submitting them to other persons. This protocol for building a consensual knowledge base is inspired of rules of review in scientific journals. The knowledge bases represented through objects are used as Web servers. By the same way, this work could be exploited for knowledge-based corporate memory building with a strong consistency constraint. In WebCokace, we also tackle «formal knowledge» annotated by documents, but we have not yet addressed the problems underlying cooperative construction of the expertise model base.
- Distributed components such as Problem-Solving Methods (PSM) can be reused, with exploitation of CORBA technology [Gennari 96]. WebCokace also allows PSM reuse, provided that they are described in the CommonKADS framework.
- Other examples of exploitation of the Web concern coordination of a project among several distributed enterprises and management of distributed knowledge [Gaines 96] or a groupware system allowing users to build software «requirements» collaboratively [Herlea 96].

This related work show that the Internet or an Intranet can be useful in an enterprise at several levels:

- for *diffusion of information or of knowledge* (documents, knowledge bases, ontologies, expertise models), such a diffusion involving capabilities for information search on Internet or Intranet [O'Leary 97],
- for *cooperative work* between members of a team, of a given department or of several departments (or even of several collaborative enterprises) working altogether throughout a given project. Cooperative construction of a corporate memory is a specific case of such a cooperative work.

At present, WebCokace mainly addresses problems of knowledge diffusion but extensions for corporate memory cooperative construction are planned.

5. Discussion and Conclusions

Clearly, our work is related to knowledge-based corporate memory, in the framework of CommonKADS: we previously justified the use of CommonKADS for corporate memory purposes, even though this method was not primarily aimed at such applications.

A Web server is interesting for distributing knowledge inside an enterprise having an Intranet: it enables information search by members of a team, of the same department or of several departments. Therefore in the framework of corporate memory, a knowledge server plays an important role.

In this context, we have developed a CommonKADS Expertise Model Web Server. It is built on top of the Cokace programming environment dedicated to CommonKADS CML. WebCokace provides the user with hypertextual navigation in expertise models and documents, a query language, a search engine and generation of graphic representations.

However, at present, WebCokace focused only on CommonKADS expertise model. Among possible extensions, we need to take into account the other CommonKADS models (organization, agent, task, communication) that are useful for different kinds of corporate memory, as noticed in Section 1.2, page 3. By the same way, an integration of an enterprise model [Uschold 96] or a user model would be interesting. Another remark is that CML textual form of expertise models is not adequate to all kinds of users: it may be handled by a knowledge engineer but for most knowledge server users, a graphic display of expertise models is more friendly and understandable. Some parts of the expertise models (concept hierarchy, task hierarchy and task dataflow) can be visualized graphically in WebCokace.

It can be wondered whether a knowledge server user prefers the display of expertise models or of texts. This question is partly addressed thanks to the hypertext links offered by WebCokace between expertise models and documents. The user can thus access both to formal knowledge (i.e. expertise models) and to informal knowledge (i.e. documents), as emphasized in [Euzenat 96] and he can navigate among both. Moreover, if the main interest of the knowledge server is navigation through documents, the expertise model base can be seen as an «intelligent indexing» of such documents.

The exploitation of WebCokace by end-users of an enterprise different from INRIA and DIST has not yet been tested. We plan such a test in order to validate the interest of WebCokace for knowledge-based corporate memory exploitation, in a real industrial context.

The integration of an *enterprise-based user model* [O'Leary 97] in WebCokace would allow to tailor the expertise model visibility and presentation format to the needs and preferences of identified categories of users: according to the enterprise department (e.g. financial service vs development office vs production center), some elements of the corporate memory could be hidden, simplified or presented differently. Moreover, in some cases, the corporate memory may be consulted outside the enterprise (for example, via an Extranet gathering some privileged partners of the enterprise such as its customers or its suppliers or some collaborative external partners). An enterprise-based user model could then help to tailor the parts of the corporate memory to be accessible to a given category of user. It could be useful to determine the presentation dynamically generated for such a user during consultation time.

We also plan to add editing facilities to WebCokace in order to enable *remote edition and collaborative development of CommonKADS expertise models*. Such remote edition will be useful for example for cooperative construction of the corporate memory by several members of the enterprise, such members being perhaps geographically dispersed.

As a conclusion, WebCokace provide a basis for expertise model development and distribution. These tools may be used for knowledge-based components of a corporate memory management system. We will focus on the validation of this approach in a real industrial context.

Let us notice that the architecture of WebCokace is generic. A similar work could be performed in the framework of another method than CommonKADS. If this method offers a modeling language, the specification of this language in Centaur generator could allow to generate an environment dedicated to this modeling language. A corporate memory developed using this method and this environment could then be consulted through a knowledge server built with the same principles and techniques as the ones presented in this paper.

Acknowledgments

We thank Francis Montagnac for the invaluable help that he provided for the implementation of the Web Server from a system point of view. We also thank the Croap team members who helped us so much to understand Centaur. Thanks to the ACACIA team members: Christophe Cointe and Myriam Ribière who developed Java applets for Cokace, Nada Matta for beta testing Cokace, Christine Ros and Alain Giboin for evaluating WebCokace user interface.

References

[Ballay 96] J.-F. Ballay, J. P. Poitou. Diadème : a Collective Knowledge Management System. *Proc. of the 4th Int. Symposium on the Management of Industrial and Corporate Knowledge (ISMICK'96)*, Rotterdam, the Netherlands, October 1994,

[Borras 88] P. Borras, D. Clément , T. Despeyroux, J. Incerpi, G. Kahn, B. Lang, and V. Pascual. CENTAUR: the system. In *Proc. of the 3rd Symp. on Software Development Environment, ACM SIGSOFT'88*, pp. 14-24, Boston, MA, November 1988.

[Breuker 94] J. Breuker and W. Van de Velde, eds. *CommonKADS Library for Expertise Modelling, Reusable Problem Solving Components*. IOS Press, Amsterdam, 1994.

[Cattell 96] R. Cattell. Ed *The Object database standard : ODMG-93, release 1.2* Morgan Kaufmann, 1996. Coll. / Revue : Morgan Kaufmann Series in Data Management Systems.

[Corby 96] O. Corby and R. Dieng. Cokace: a Centaur-based environment for CommonKADS Conceptual Modelling Language. In W. Wahlster ed, *Proc. of the 12th European Conference on Artificial Intelligence (ECAI'96)*, pp. 418-422, John Wiley & Sons, Ltd. Budapest, Hungary, 1996.

[Dieng 96]R. Dieng, A. Giboin, C. Amergé, O. Corby, S. Després, L. Alpay, S. Labidi, S. Lapalut. Building of a Corporate Memory for Traffic Accident Analysis. In B. Gaines, M. Musen eds, *Proc. of the 10th Banff Workshop on Knowledge Acquisition for Knowledge-Based Systems (KAW'96),* Banff, Canada, November 1996, pp. 35-1 – 35-20.

[Ermine 96] *Les systèmes de connaissances.* J. L. Ermine. Hermès, 1996.

[Euzenat 96] J. Euzenat. Corporate memory through cooperative creation of knowledge bases and hyper-documents. In B. Gaines, M. Musen eds, *Proc of KAW'96,* Banff, Canada, November 1996, pp. 36-1 – 36-18.

[Farquhar 96] A. Farquhar, R. Fikes and J. Rice. The Ontolingua Server: a Tool for Collaborative Ontology Construction, USA. In B. Gaines, M. Musen eds, *Proc of KAW'96,* Banff, Canada, November 1996, pp. 44-1 – 44-19.

[Fraser 94] J. Fraser. Managing Change through Enterprise Models, in *Proc. of Expert Systems '94, the 14th Annual Conference of the British Computer Society Specialist Group on Expert Systems*, Cambridge UK, December 1994.

[Gaines 96] B. Gaines and M. Shaw. A Networked, Open Architecture Knowledge Management System, Canada. In B. Gaines, M. Musen eds, *Proc of KAW'96,* Banff, Canada, November 1996, pp. 37-1 – 37-18.

[Gennari 96] J. Gennari, A. Stein and M. Musen. Reuse for knowledge-based systems and CORBA components. In B. Gaines, M. Musen eds, *Proc of KAW'96,* Banff, Canada, November 1996, pp. 46-1 – 46-16.

[Grunstein 95] La capitalisation des connaissances de l'entreprise, système de production de connaissances. *L'entreprise apprenante et les Sciences de la Complexité.* Aix-en-Provence, Mai 1995.

[Herlea 96] D. Herlea. Users' Involvement in the Requirements Engineering Process, In B. Gaines, M. Musen eds, *Proc of KAW'96,* Banff, Canada, November 1996, pp. 47-1 – 47-17.

[Huynh 94] M. Huynh, L Popkin, M. Stecker. Constructing a Corporate Memory Infrastructure from Internet Discovery Technology. White Papers, Marble Associates Inc., *<http://www.marble.com/chi-bin/list-white-papers>*, 1994.

[Kingston 94] J. K. C. Kingston. Modelling Business Processes using the Soft Systems Approach. In J. P. Barthès ed., *Proc. of the 2nd Int. Symposium on the Management of Industrial and Corporate Knowledge (ISMICK'94)*, Compiègne, October 1994, pp. 149-159.

[Malvache 93] P. Malvache, P. Prieur. Mastering corporate experience with the Rex method. In J. P. Barthès ed., *Proc. of ISMICK'93*, Compiègne, October 1993, pp. 33-41.

[Matta 96a] N. Matta. Conflict management in concurrent engineering: Modelling guides. In J. Mueller and R. Dieng, editors, *Proc. of ECAI'96 Workshop on Modelling Conflicts in AI*, pages 29–33, Budapest, Hungary, August 1996.

[Matta 96b] N. Matta, O. Corby. *Modèles génériques de gestion de conflits dans la conception concourante*. INRIA Research Report RR-3071, December 1996.

[O'Leary 97] D. E. O'Leary. The Internet, Intranets and the AI Renaissance. *Computer*, 30(1):71-78, Janvier 1997.

[Poitou 95] J. P. Poitou. Documentation is Knowledge: An Anthropological Approach to Corporate Knowledge Management. In J. P. Barthès ed., *Proc. of the 3rd Int. Symposium on the Management of Industrial and Corporate Knowledge (ISMICK'95)*, Compiègne, France, pp. 71-80.

[Pomian 96] F. Pomian. *Mémoire d'entreprise, techniques et outils de la gestion du savoir.* Ed Sapientia, 1996.

[Sacile 95] R. Sacile. *Using CommonKADS to build an expertise model for breast cancer prognosis and therapy.* INRIA Research Report, RR-2737, December 1995.

[Sacile 96] R. Sacile, C. Ruggiero, R. Dieng. Using CommonKADS to create a conceptual model of a guideline for breast cancer prognosis. *Medical Informatics* **21**(1):45-59. 1996.

[Schreiber 94a] G. Schreiber, B. J. Wielinga, R. de Hoog., H. Akkermans and W. Van de Velde. CommonKADS: a Comprehensive Methodology for KBS Development. *IEEE Expert*, **9**(6):28-37, Dec. 1994.

[Schreiber 94b] G. Schreiber, B. Wielinga, H. Akkermans, W. van de Velde, and A. Anjewierden. CML: The CommonKADS Conceptual Modelling Language. In L. Steels & al, eds, *A Future for Knowledge Acquisition: Proc. of EKAW'94*, pp. 1–25, Hoegaarden, Belgium, Sept. 1994. Springer-Verlag, LNAI n. 867.

[Schreiber 94c] G. Schreiber, B. Wielinga and W. Jansweijer. The KACTUS View on the 'O' Word. In *Ontology Dev. and Use Workshop*, 1994. Available at the URL: `http://www.swi.psy.uva.nl/projects/Kactus/home.html`

[Simon 95] G. Simon, M. Grandbastien. Corporate knowledge: a case study in the detection of metallurgical flaws. In J. P. Barthès ed., *Proc. of ISMICK'95*, Compiègne, October 1995, pp. 43-52.

[Simon 97] G. Simon. *Modèles et méthodes pour la conception des mémoires d'entreprise. Le système DOLMEN : une application en métallurgie.* PhD Thesis, Université Henri Poincaré, Nancy I, Janvier 1997.

[Steels 93] L. Steels. Corporate Knowledge Management. In J. P. Barthès ed., *Proc. of ISMICK'93*, Compiègne, October 1993, pp. 9-30.

[Tourtier 95] P.-A. Tourtier. *Analyse préliminaire des métiers et de leurs interactions.* Rapport intermédiaire du projet GENIE, INRIA-Dassault-Aviation, 1995.

[Uschold 96] M. Uschold, M. King, S. Moralee, Y. Zorgios. The Enterprise Ontology. Mars 1996. *http://www.aiai.ed.ac.uk/~entprise/enterprise/ontology.htm*l

[VanderSpek 94] R. Van der Spek. Towards a methodology for knowledge management. In J. P. Barthès ed., *Proc. of ISMICK'94*, Compiègne, October 1994, pp. 93-102.

[VanHeijst 96] G. van Heijst, R. van der Spek, E. Kruizinga. Organizing Corporate Memories. In B. Gaines, M. Musen eds, *Proc. of KAW'96*, Banff, Canada, November 1996, pp. 42-1 – 42-17