Corporate memory through cooperative creation of knowledge bases and hyper-documents

Jérôme Euzenat

INRIA Rhône-Alpes
655 avenue de l'Europe, 38330 Montbonnot Saint-Martin, France
Jerome.Euzenat@inrialpes.fr

ABSTRACT: The Co4 system is dedicated to the representation of formal knowledge in an object and task based manner. It is fully interleaved with hyper-documents and thus provides integration of formal and informal knowledge. Moreover, consensus about the content of the knowledge bases is enforced with the help of a protocol for integrating knowledge through several levels of consensual knowledge bases. CO4 is presented here as addressing three claims about corporate memory: (1) it must be formalised to the greatest possible extent so that its semantics is clear and its manipulation can be automated; (2) it cannot be totally formalised and thus formal and informal knowledge must be organised such that they refer to each other; (3) in order to be useful, it must be accepted by the people involved (providers and users) and thus must be non contradictory and consensual.

The development of corporate memories has several well-acknowledged reasons: people are leaving firms with their know-how; work-sharing is expanding; and, as professions are more specialised and the work more cooperative, it becomes difficult to establish conventions between people (Durstewitz, 1994; Conklin, 1996). Throughout the present paper, a corporate memory is considered as a repository of knowledge and know-how from a set of individuals working in a particular firm. This view is discussed later (§5). One can already imagine several laboratories and firms grouped together with the aim of building and maintaining an encyclopaedic server (whose contents can be text and images for instance) about a particular domain. For that purpose, their group can be implemented through one or several software agents aimed at collecting data and distributing it to those who are allowed to consult them (such a description roughly corresponds to a HTML server on the World-Wide Web and the famous Lotus Notes program does not work another way).

This "computer as medium" idea could be strengthened by extending it towards the knowledge itself. However, stating knowledge on computer medium is not enough for several reasons: it does not promote communication among individuals nor confrontation against standard or analysis tools. For that purpose, a computer environment called CO4 (for collaborative construction of consensual knowledge) is presented in (Rechenmann, 1993; Euzenat, 1995a). Three axioms about corporate memories are provided here for justifying the CO4 approach.

The first axiom of the approach is that knowledge must be stated as formally as possible. This has clear advantages since the knowledge can be given a semantics, can be manipulated by computers according to that semantics and properties of the repository (among which consistency and completeness but also subsumption) can be checked. However, not everything can and must be formalised and even if it were, the formal systems could suffer from serious limitations (complexity or incompleteness).

The second axiom states that it must be possible to wrap up a skeleton of formal knowledge with informal flesh made of text, pictures, animation… Thus, knowledge which has not yet reached a formal state, comments about the production of knowledge or informal explanations can be tied to the formal corpora.

The third and stronger axiom is that people must be supported in discussing about the knowledge introduced in the knowledge base. In this perspective, re-using, diffusing and maintaining knowledge should be a participatory activity of all the involved people (providers and users). Users will use the knowledge only if they understand it and they are assured that it is coherent. The point is to enforce discussion and consensus while the actors are still at hand rather than hurrying the storage of raw data and discovering far latter that it is of no help. The presented work is really cooperative: it is not restricted to give good or bad marks, but it tries to involve and commit each participant in the process. The objective is not to build a corporate record but to build a coherent corporate memory.

The CO4 (Euzenat, 1995a) system, which is presented here, is dedicated to the incremental and concurrent building of a knowledge base organising, around formalised knowledge, a set of various annotations (text, bibliography, image, experimental data, etc.). It should provide collaborators with support for, on one hand, expressing, annotating and manipulating their knowledge, and on the other hand, submitting it to other people.

To that extent we propose a protocol based on the agreement of every participant. It is based on formal rules inspired from the peer review process of scientific journals. Hence, instead of merely reproducing the paper journals in computers, the principles of scientific journals are applied to the knowledge formally expressed in a computer. The result should be a consensual knowledge base, i.e. which everybody in a group agrees to be a reasonable state of the art. It can be used for confronting new results and for learning new knowledge.

The knowledge expression formalisms and facilities are first presented (§1) before turning to consultation and modification of knowledge bases on the World-Wide Web (§2). Then the principles of the protocol for building knowledge bases and the way knowledge bases are related are given (§3). Afterwards, the definition of the protocol is briefly sketched (§4). The presentation aims at giving a global view of the system and how it addresses the problems raised above. More precise presentation of the tools can be found elsewhere — TROPES (Sherpa, 1995), HYTROPES (Euzenat, 1996) and the CO4 protocol (Euzenat, 1995b).

1. KNOWLEDGE REPOSITORY

Users interact with their knowledge bases. In such a base, knowledge can be formally expressed in a formal language carrying a precise semantics. The present section describes the components of a knowledge base (§1.1) and their manipulation by a user (§1.2). This is completed by a description of the software involved in base management (§1.3).

1.1. The knowledge model

CO4 comes from numerous experiments with knowledge bases in the domain of molecular genetics. ColiGene (Perrière and Gautier, 1993), for instance, describes the regulation mechanism of gene expression in the *E. coli* bacterial genome. PowerGene (Médigue, Vermat, Bisson, Viari, and Danchin, 1995) allows a library of programs (e.g. pattern learning, sequence comparison) to be applied to genomic data stored in databases and chained in order to generate knowledge. It is now extended towards the distribution of the programs and databases.

The design of these knowledge bases has led to the identification of four types of knowledge to be represented:

Descriptive knowledge on the biological entities involved is represented in an object-based knowledge representation system. This enables the representation of classes of objects (e.g. genes), subclasses (e.g. protein genes) and the identification of an object as belonging to such a class. In CO_4, the descriptive knowledge is available in the TROPES system (Sherpa, 1995). The semantics of the system is given in set-theoretic terms and the operations (e.g. filtering, classification) respect it. TROPES is able to organise objects into multiple independent taxonomies and allows the user to work on a subset of these taxonomies. It thus satisfies the need to express several viewpoints on object classifications (Overton, Koile, and Pastor, 1990; Karp and Mavrovouniotis, 1994).

Methodological knowledge specifies the ways to select and link up methods for a given task. It is represented through a task management system able to integrate, represent, process and monitor the many computer programs for analysing the results of experiments. These tasks can be understood as executable KADS tasks (Schreiber, Wielinga, and Breuker, 1996) and are given a semantics in terms of problems and methods. They are organised along a specialisation (more abstract/more specific) link and a composition (task/sub-task) link. The composition can be expressed through various operators (e.g. parallelism, sequence, alternative). The methodological system is available through the CONTROL system and will be integrated with TROPES under the name of TNT ("Task in Tropes").

Behavioural knowledge, which has not been introduced in the two knowledge bases above, concerns the modelling of dynamic phenomena, such as the dynamics of gene inhibition or activation, through a qualitative modelling system. Such kind of knowledge has already been used for representing metabolism (Karp and Mavrovouniotis, 1994). Behavioural knowledge is under study in the context of a knowledge base on the genetic regulation in the early development of the fruit fly egg.

Non formal annotations on the various objects and tasks involved are achieved through a hypertext system which connects hypertext nodes with the components of the descriptive and methodological knowledge. It allows browsing among texts, objects and tasks. The non formal (and especially hyper-textual) annotations are the subject of the next section (§2).

Some authors (Conklin, 1996) call formal knowledge the knowledge expressed in reports or recorded communication and informal knowledge the processes in a firm. As it can be understood from above, the process knowledge (as it seems better to call it) can be recorded and formalised as methodological knowledge.

1.2. Formal manipulation

An advantage of formal knowledge is the ability of the system to apply formal transformations (or operations) as far as they respect the semantics of the knowledge. The system is able to deal with sophisticated queries asking if a piece of knowledge is redundant, subsumed or similar (w.r.t. some distance) to a part of the knowledge base. These queries are subject to limitations drawn by the expressiveness of the knowledge representation language and the expected degree of completeness of the answer.

As an example, change (for instance the destruction or addition of some objects in the base) is the most critical operation. It can seem obvious that the user may query and modify the base and that modifications must leave the base in a consistent state. However, the way this is achieved in CO_4 is particular since it must work when the researchers want to communicate a part of their bases to another base. For instance, imagine that some user wants to add a particular task (a specialisation of the `filter-network` task called `extract-paths` which has as input a `network` of interactions and two `genes` and as output another `network`: the one which contains only interactions in paths from one of the genes to the other).

If the proposal is consistent with the base, the change is committed and the knowledge base is simply modified. But this can be otherwise, for instance because the user typed `fiter-network` instead of `filter-network` or because `filter-network` has a `contextual-network` as output while the user specified the more general TROPES class `network` for the new task. In such a case, the system is able to detect the problem: this is a first contribution for delivering trustful corporate knowledge.

However, the system is also able to reason on the knowledge and to help the user to repair some of these mistakes. For the first problem, a lexicographic corrector is sufficient for deciding that the closest task name is `filter-network` while for the second one a revision algorithm is necessary (Crampé and Euzenat, 1996). Such a module is able to propose several repairs to the user (e.g. restricting the type of output in `extract-paths`, expanding it in `filter-network`, making `extract-paths` a specialisation of a more generic task).

This ability to process formal manipulation of knowledge base is precious to the users of simple knowledge bases, however, it is invaluable to users who are not the authors of some pieces of knowledge (e.g. `filter-network`) involved in the problem.

1.3. Knowledge base architecture

For completeness purpose, figure 1 presents the software architecture of the CO4 system. It is made up of a communication layer (whose use is presented below) and five components:

- The knowledge base storage;
- The update and revision controller is what is usually the knowledge base management system: the module which detects inconsistency and provides possible repairs.
- The negotiation controller interacts with the outside world for signalling the mistakes and possible repairs (obtained from the revision controller). It also displays the queries generated by the cooperation with other users;
- The cooperation controller is a library of functions which implements the CO4 protocol;
- The base definition refers to the definition of the connection of the base with other ones.

While the revision and negotiation controllers have been exemplified above, the remainder is considered below.

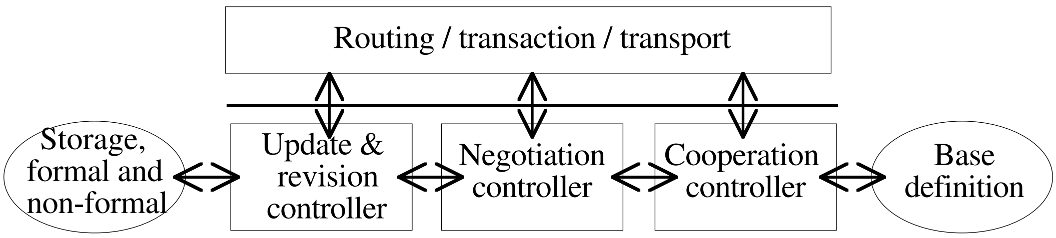

Figure 1. The software architecture. Each box represents a software module, each circled unit is a data/knowledge repository and each arrow represents the call of a program functionality. The complete description is given here for being compatible with other pictures.

Having sketched a representation enabling the expression of formal knowledge, the connection with informal knowledge must be considered.

2. INFORMAL KNOWLEDGE: HYTROPES

As anyone would agree, storing only formal knowledge in terms of objects, tasks and equations is far too formal and restrictive. So the formal knowledge is connected to informal knowledge (mainly in terms of text and images) structured in a hypertext network. This informal knowledge has two purposes: recording the reasons for acceptance and changes in the knowledge base and adding annotations to the formal knowledge. Firstly, the reasons for linking formal knowledge to informal knowledge are discussed (§2.1). Then are described the advantages of the WWW as a hypermedia management system related to a knowledge base (§2.2), but above all the advantages of generating HTML pages from a knowledge base instead of generating them by hand or from documents (§2.3).

2.1. Formal and textual knowledge

Mixing knowledge bases and hypermedia has been already achieved for long. Such an idea can be put forth for various reasons:

- The knowledge in knowledge bases cannot be considered in isolation from other knowledge sources available in firms or laboratories (e.g. bibliographic references, full text papers, experimental data and programs). This relationship must be established in one way or another. For anyone who wants to explain, to annotate or at least to document a knowledge base, "hypermedia" are now the main support;
- The World-Wide Web (WWW) enables the publication of a knowledge base. Thus even users with little computer knowledge can access it at low cost: one server can have many users on inexpensive workstations (Riva and Romani, 1996; Farquhar, Fikes, Pratt, and Rice 1995);
- Another deeper reason is that the possibility to consult a knowledge base like an encyclopaedia is considered a major interest by several researchers. Knowledge-based WWW pages can be read as hypermedia documents and also consulted for problem solving (Gaines, 1990). This idea is related to the knowledge medium concept (Stefik, 1986). It has been further investigated by (Gaines, 1990) and (Rechenmann, 1993).

The success of the WWW is an opportunity to test this latter idea in the large. The WWW is a very good support for the diffusion of knowledge. However we claim that the formal representation of knowledge is a very important issue for the WWW — see also (Weld, 1995; Hüser, Reichenberger, Rostek, and Streitz, 1995). As a matter of fact, the results actually provided by the various WWW worms are so huge and so often irrelevant that formal organisation of knowledge will soon be unavoidable. Moreover, such an organisation should help the user and the server to share some common ground.

2.2. WWW diffusion

As already noted by various authors — e.g. ISF (Rees, Edwards, Madsen, Beasley, and McClenaghan, 1995), WebMap (Gaines and Shaw, 1995) — an object-based representation system is already a web of related objects. The similarity between WWW pages and objects is quite obvious and the mapping from one to another is straightforward.

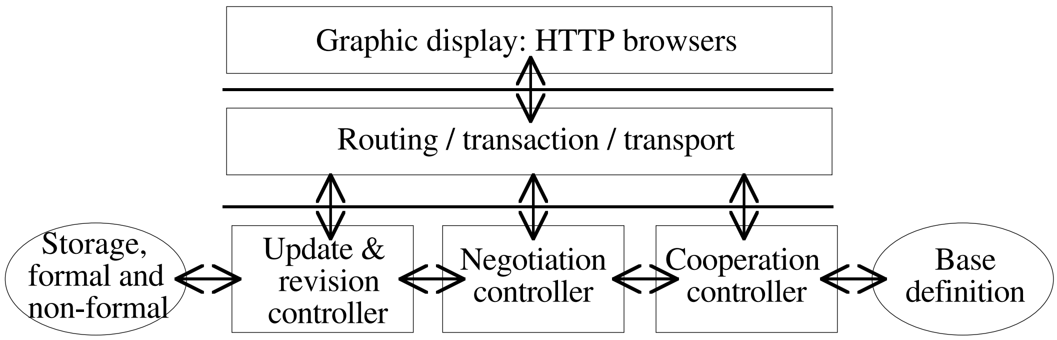

Figure 2. The base is accessed locally and remotely (through networks) via a HTML browser. This is achieved by the negotiation controller which handles HTML queries.

Knowledge bases can be used as WWW servers whose skeleton is the structure of formal knowledge (mainly in the object-based formalism) and whose flesh consists of pieces of texts, images, sounds and videos tied to the objects. Turning a knowledge base system into a WWW server is easily achieved by connecting it to a port and transforming each object reference into a URL. If the knowledge base is already documented by WWW pages, the latter remain linked to or integrated into the pages corresponding to these objects. Our old SHIRKA system was already able to link formal knowledge with pictures and texts through a proprietary hypermedia system; TROPES is provided with its HTTP server counterpart HYTROPES[1] (Euzenat, 1996). The link in term of CO_4 is presented in figure 2.

The advantages of such an approach with regard to the previous proprietary hypertext systems are chiefly the availability of the knowledge base content to a wide and untrained audience. HYTROPES has access to the internet, the internet has access to HYTROPES but access can also be easily restricted to the so-called “intranet”. However, other advantages are found in the consistency of the base (for instance, there is no dangling link since the skeleton is generated automatically and formal information is maintained sound).

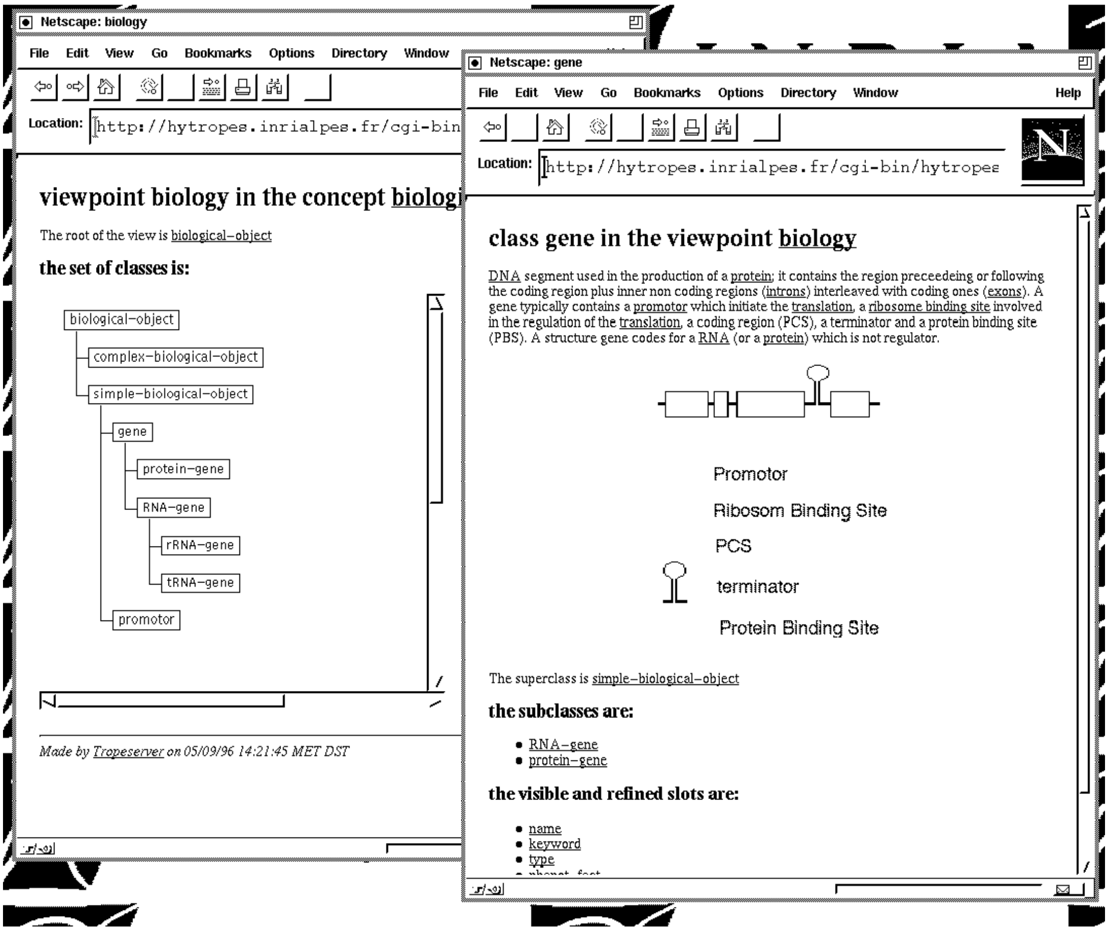

Figure 3. Presentation of a viewpoint and the gene class from a TROPES version of the COLIGENE knowledge base (Perrière and Gautier, 1993).

2.3. Intelligence added WWW servers

World-wide availability and safety are valuable contributions, however they are quite restricted with regard to the possibilities of knowledge bases. As a matter of fact, from a knowledge base server it is possible to build complex queries grounded on formal knowledge (see figure 4). For instance, a user looking for an apartment in a real estate knowledge base can first select a filter

---

[1] The technical aspects of the available TROPES server are presented in the HYTROPES home page (`http://hytropes.inrialpes.fr`) together with demonstrations of the program and the sources.

form from the `house` concept, ask the lexicon for the meaning of the slot/word `F3` and decide to fill the form with corresponding criteria; the user can select one of the objects given as answers and have a look at the ground map and at a picture of the house together with the usual precise information. This combines the advantages of a very structured server with the freedom of usual servers. Moreover, the answer will be given in function of a semantically sound method instead of using a simple full-text search.

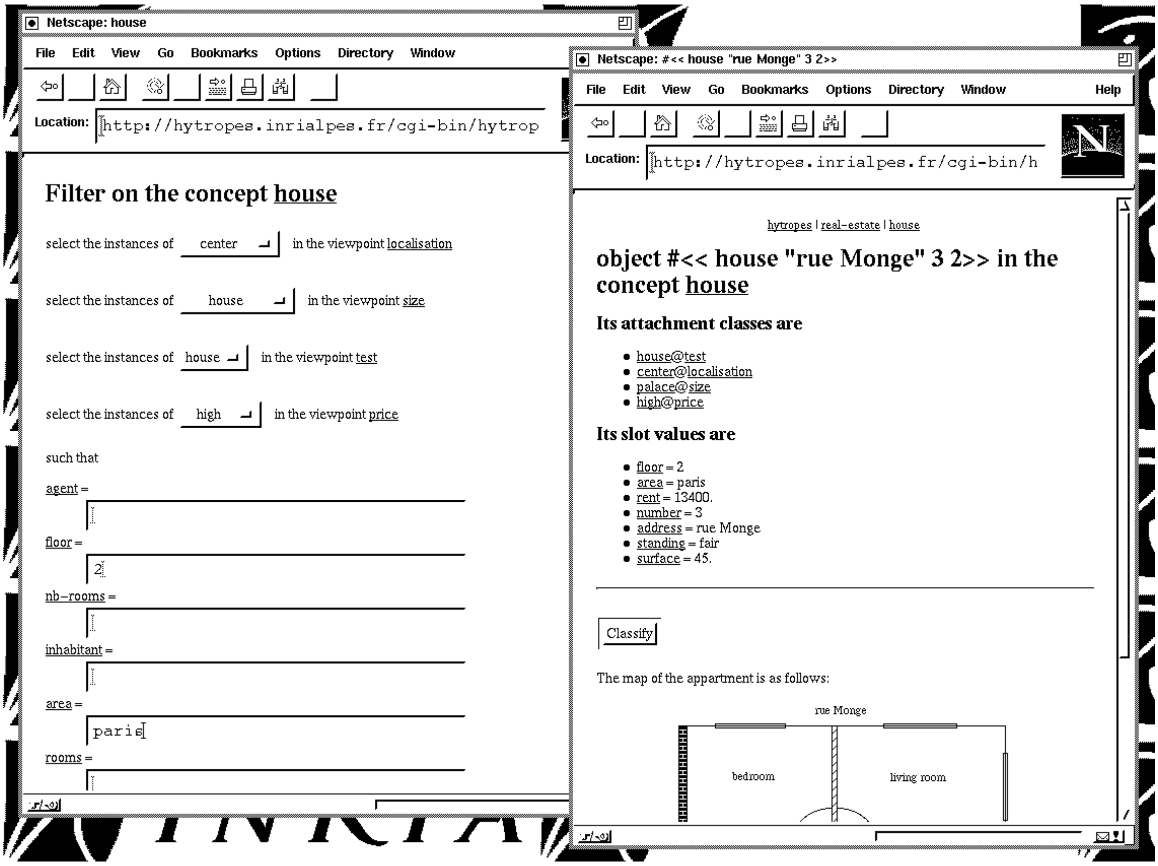

Figure 4. Filter and instance WWW page from a sample real estate knowledge base.

HYTROPES makes filtering and classification queries available. Actually the whole TROPES API is subject to “URLising”. So the next step will consist in publishing the URL rules (far simpler than the URL generated by the FORM tag) in order to enable any other application to:

- properly query a knowledge base;
- generate WWW pages linked with any knowledge base.

The queries may be as complex as needed: it will be possible not only to browse but also to build and modify knowledge bases. This raises problems of concurrent access and user support. Thus, the way it is achieved through the CO4 protocol is now presented.

3. TOWARDS PEER-REVIEWED KNOWLEDGE BASES

The primary aim of CO4 is the construction of a consensual base. The principles underlying CO4 are derived from those of peer-reviewed journals: before being introduced in a consensual knowledge base, the knowledge must be submitted to and accepted by the community. At the end of the process, it is eventually intended that knowledge stored in a consensual knowledge base is safe enough so that anybody can use it confidently and easily. The organisation of bases in order to contribute to a consensual knowledge base is presented (§3.1) before providing the principles of the CO4 protocol for integrating knowledge (§3.2) and an example of its use (§3.3).

3.1. The network of bases

In order to build a consensual knowledge base, the individual bases presented above must be linked together. In CO4, any cooperator is viewed by the system as a base. Bases are organised into a tree whose leaves are user bases and whose intermediate nodes are called group bases (see figure 5). Each group base represents the knowledge consensual among its sons (called subscriber bases). This structure imposed to the collaboration can be stuck on the structure of a particular firm or that of a particular group in the firm, but it can also be independent of it. A base can subscribe to only one group. A human user can create several bases (possibly subscribing to different group bases) representing different trends, and knowledge can be transferred from one base to another. Also, nothing prevents several human users from sharing the same base.

Some independent knowledge bases can subscribe to group bases as observers: these bases are sent by the group base whatever is introduced in the knowledge base but cannot modify it. Observers are not considered further in this presentation.

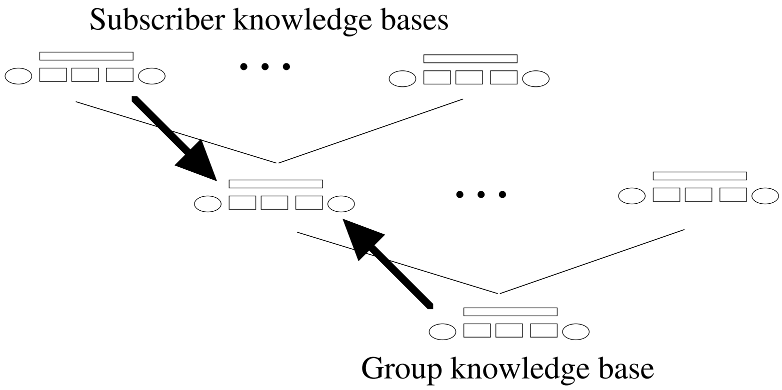

Figure 5. Hierarchical architecture and message flow (dark arrows). Bases are organised in a tree whose leaves are individual bases and nodes represent consensus between the connected individual bases.

Group bases have the same structure as individual ones: they are made of the same pieces of software. The main difference between group bases and individual workstations is that the former are completely automated and only respond to stimuli from other bases: they do not require human assistance.

Bases are linked together in such a way that a particular base knows its subscribers and its group base. These links are in fact virtual and are implemented by the flow of messages from one base to another. To its subscribers a group base mainly sends messages for broadcasting a change accepted by everyone and calls for comments in order to establish whether a change must be committed or not. The other way, a (group or individual) base sends to its group base changes which it wants the group base to integrate. Of course, any base, as a group base, also receives changes to commit and as an individual base also receives calls for comments and change broadcast.

3.2. Paper submission metaphor

Any system allowing the building of some artefact must have a particular change policy. The CO4 protocol mimics that of editorial boards: before being introduced in a consensual knowledge base, knowledge must be submitted and accepted by the community. To our knowledge, the peer-reviewing protocol (Peters, 1995) has never been used for building knowledge bases. The choice of such a protocol is not neutral: it proved to be practicable within the scientific community and, in the consensual version, it enforces the dialogue between people (rather than a simple majority or intersection protocol).

Consistency and formality require more strictness in the protocol than pure peer-reviewing; this leads to the consensus requirement (i.e. in which, a modification, for being accepted, must have been agreed by all other members — for instance, in the context of genome sequencing, a consensus map is a map that people involved in the research field think correct). Integrating knowledge goes through the process of submitting knowledge to the base, of letting it be reviewed by the other participants and of accepting or amending it according to their reactions. Informal knowledge is also subject to submissions, reviewing and so on. This protocol works at several levels: the group bases can be grouped together into a more important group base and so on. However, the behaviour of such a group base is still subject to the consensual approbation of its subscribers. Thus, for instance, a consensual representation could be achieved inside a particular firm before being submitted to the inter-institution base.

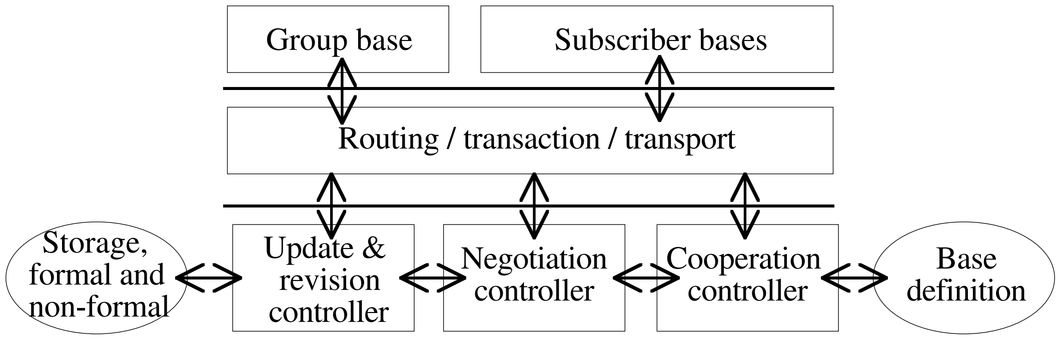

Figure 6. The base definition contains the situation of a base inside the tree of bases. It thus stores the addresses of group and subscriber bases. When a problem occurs, the negotiation controller has the choice between displaying it on a HTML browser (individual base) or passing it to the cooperation controller which sends it to the subscriber bases (group base).

In a first version of CO4, the group base applies non destructive modifications without discussion. These modifications are always possible in the group base, since they are in the subscriber base which contains it. In the case of destructive modifications, a call for comments, identified by a unique number, is issued to every subscriber. Among the answers provided by the subscribers, three cases may happen:

- All of them agree on the modification acceptance, then the modification is committed into the group base and broadcast to every subscriber knowledge bases;
- One of them rejects the proposal, then the changes are not committed and the comments provided by the rejecter are sent to the submitter (the call for comments is discarded in all the subscribers knowledge bases);
- One submitter sends an alternative proposal, then the call for comments is replaced by a call for comments about all the available proposals (those who already accepted the change, are asked to consider the new proposal and to answer again).

It can also happen that the submitter retracts the proposal thus leading to the retraction of the call for comments from all the knowledge bases.

So let briefly see how to connect a base and submit knowledge.

3.3. A glimpse at knowledge submission

The workstation user can subscribe to a consensual knowledge base by sending a request which has to be accepted by the group. Upon acceptance the respective knowledge base definitions of the group base and the individual base are aware of each other.

As soon as the base is part of a group base, it receives the complete contents of that base (to which it is supposed to subscribe), it is entitled to give its opinion on all submissions currently under examination and is allowed to submit knowledge. The interesting point is the submission of knowledge which is described right below.

The consultation or modification is initiated through the graphic interface and directly processed by the revision controller. However, in the usual mode, the modification is prepared by a confrontation query which asks for a comparison of a new piece of knowledge (made of objects, tasks, hypertext nodes and qualitative equations) and the knowledge base. The comparison of a corpus of knowledge with another results in a report about what is different, what is the same and what is contradictory. If the piece of knowledge does not contradict the knowledge base, it can be submitted for integration. If it contradicts it, the researcher can modify it in order to fit the group base. Concerning the informal documentation, if, for instance, the user wants to create a hypertext node it is possible to detect if a node with the same name already exists and to negotiate its modification.

When the subscribers are confident enough with some pieces of knowledge, they can submit them to the group knowledge base to which they subscribe. This is achieved by circumscribing the submitted part (which can include hypertext annotations justifying them) and calling the submission procedure of the negotiation controller. In order to complete the submission message, the negotiation controller collects the sets of differences between the consensual group base connected and the selected changes (they are logged in by the revision controller) and sends them to the group base. Usually, the group base, through its own revision and negotiation controllers, issues a report describing how the submitted knowledge can be added to the group base. Thus, as usual, the user can choose a better (and consistent) way to achieve the submission. This proposal will be submitted to the other subscribers and committed if it reaches consensus.

As a subscriber of the group base, the user also receives the call for comments issued by the group base in response to the submission of some material. Users can read the submission or apply it in their own knowledge base by submitting it to the revision controller. This can result in a favourable agreement report or an inconsistency detection that can be used by the user for issuing an alternative proposal. In response to the call for comments, users must answer by one of the following: accepted when they consider that the knowledge must be integrated in the consensual knowledge base, rejected when they do not, and alternate when they propose another change.

When the group base has gathered enough comments, it integrates, or not, the change in the base. The change being now consensual, it is broadcast to all subscribers. It may happen, however, that the research they are currently involved in contradicts what is in the group base. So users can refuse the new piece of knowledge (just as they can modify parts of the group base knowledge in their local base) which is then stored in a change logbook for further change submission.

The fact that anyone can maintain a knowledge base different from the consensus obviously allows the exploration of alternative paths. On a more basic ground, it enables communication, negotiation and acceptation to be asynchronous. This reproduces the way papers are submitted, discussed and accepted or rejected in a scientific journal: reviewers can take time for carefully examining a proposal since it will not stop the work of the base which issued it.

4. CO4 PROTOCOL POLICIES

Governed by the above principles, there exists a complete protocol for dealing with cooperation in CO4. It is presented below through general interaction schemes (policies). The way in which communication between bases can be represented is sketched (§4.1) as a prerequisite to the establishment of such policies (§4.2). The impact of the consensual approach of CO4 is then discussed (§4.3).

4.1. Network and messages

The cooperation protocol is based on the architecture of the knowledge bases (some of which being group bases, the other ones being only subscribers) and a complete set of behaviour rules (Euzenat, 1995b). The protocol has been kept flexible, extensible and general but non trivial. The formalism used for describing it has been kept simple: rules are triggered by a single event which is identified by the class of the sender and the name of the message, the reaction taken into account is the sending of other messages and the manipulation of ordinal counters and sets. In fact, it could be straightforwardly transformed into LOTOS (Bolognesi and Brinksma, 1987). The protocol has several particularities:

— there is no need for human intervention in the group bases;
— there is no message but from subscriber base to group bases and back;
— every decision has been approved by all the subscribers (and recursively for a group of group bases).

The messages sent from one base to another are expressed through a speech act (loosely inspired from "speech act theory"). Speech acts are used here for building a set of relevant "artificial" acts rather than trying to understand the "natural" speech acts occurring in ordinary conversation (in the same way that grammars have been borrowed by computer science from linguistics). Each particular act carries a precise semantics (taken as the goal of the sender). The notion of a speech act has several advantages for the particular architecture presented here:

- It allows the separation of knowledge from its use (its addition or retraction from a particular base, for instance);
- It is independent from the knowledge representation language and the protocol can thus be expressed abstractly and the library implementing it can be used with other systems;
- A speech act can refer to another speech act (retracting a submission, for instance).

The inter-base communication uses KQML — Knowledge Query and Manipulation Language (Finin, Fritzson, MacKay, and MacEntire, 1994) which has been chosen because, in its early stages, it clearly distinguished the three levels required by CO4:

Communication: to be used for the communication layer in order to route the message;
Performative: to be used by the negotiation and revision controllers in order to know what kind of action the message is intended to achieve;
Content: to be used by the knowledge base management system.

The actual protocol is routed automatically (once a knowledge base has subscribed to another), the performative and content levels are interpreted automatically in the group bases and the performative level is automatically interpreted by the individual knowledge bases (however, the system asks the user before committing these performatives).

4.2. Conversation policies

The protocol is presented below through the main conversation policies (i.e. protocol skeleton) implemented in the CO4 protocol. Conversation policies are presented as diagrams intended to express how a query from a knowledge base can be processed by the others. This is a very general and synthetic description of what happens. These policies, in CO4, are reduced to only two schemes depending on which knowledge base the initiative comes from: from the group bases to the subscriber bases or the other way around.

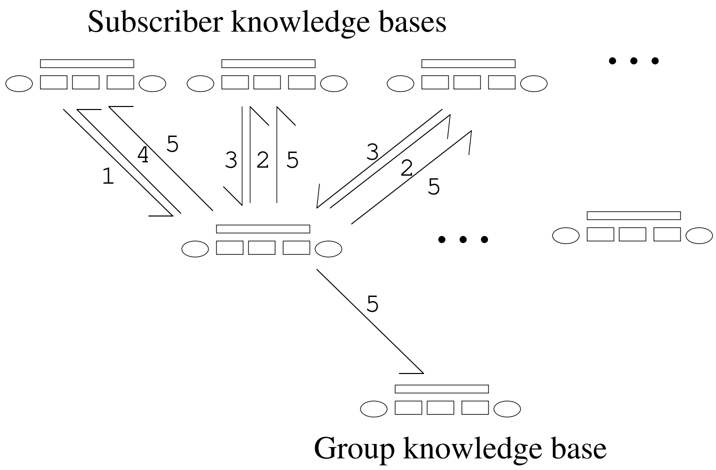

Figure 7. Downward policy: how a group base processes an initiative from a subscriber.

A policy can be schematised by a picture and a table. In the picture (see figure 7), arrows represent messages; numbers labelling them are a stratification of their occurrence order. All the messages carrying the same number must have been sent before the arrows carrying the successor can be instanciated. The table (see table 1) provides the name of the performative of each message in each instance of the policy (e.g. for subscribing or submitting a piece of knowledge the performative initiating the conversation is not the same). The arrows may or may not be instanciated. Moreover, additional communication may happen between two stages (for instance a group base which receives a call for comments, initiates its own call, and replies to the former only when the latter reaches completion).

downward policy	subscribe	submit	forward
1: query	`register`	`achieve`	`forward(P)`
2: call	`ask-all`	`ask-all`	`ask-all`
3: vote	`reply`	`reply`	`reply`
4: report	`notify`	`notify`	`notify`
5: commit	`tell`	`tell`	`P`

Table 1. Downward policy instanciation.

For downward policy, there are 5 stages which can be instanciated in three distinct processes. For instance, the submission is achieved through (1) a base sending an `achieve` message to its group base, (2) a call for comments emitted with the `ask-all` message from the group base to its subscribers, (3) a `reply` from the subscribers to the group base accepting or rejecting the proposal (which corresponds to their vote for or against the proposal), (4) a notification of the issue to the initial sender and (5) the introduction in the group base of the proposal and a broadcast of this to all the subscribers with the `tell` performative.

The same stages are found in the upward policy (see figure 8 and table 2) but they do not correspond to the same set of arrows. For instance, a broadcast is achieved through (1) the `tell` message considered above, (2) a call for comments to all the subscribers (for group base), (3) the same `reply` as above from the subscribers, (4) nothing in that case and (5) the introduction or not in the base of the content of the message and its broadcast to the subscribers (through a new `tell`).

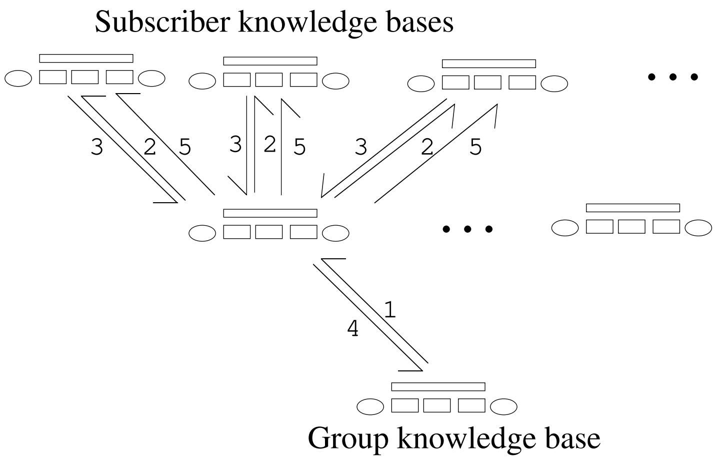

Figure 8. Upward policy: how a group base processes an initiative from its own group base.

upward policy	call	broadcast	retract
1: query	`ask-all`	`tell`	`deny`
2: call	`ask-all`	`ask-all`	`deny(ask-all)`
3: vote	`reply`	`reply`	
4: report	`reply`		
5: commit		`tell`	

Table 2. Upward policy instanciation.

An additional policy is used for the `deny` performative which depends on the wrapped performative and exactly undoes what the performative does and the usual policy for evaluate/reply is used.

4.3. Consensus?

While building the protocol, some formal properties are required and ensured. For instance, under the fairness and non dependency assumptions (quite usual in distributed voting protocols), it is established that each submission reaches the accepted or rejected status in a finite amount of time and this status is such that a submission is accepted if and only if it agrees each subscriber.

The consensus aspect is dealt with through the acceptation of proposals which achieve acceptance from every subscriber and the rejection of other proposals. Consensus could be replaced by some other definitions (like majority or intersection, see §4.1), but it has been retained for two reasons: (1) it enjoys interesting formal properties (if a consensual base contains only knowledge which is accepted by all the subscribers, this remains true if subscribers are added or retracted), and (2) it should lead to the discussion of proposals — not only conflicts — and thus the collaboration of the researchers. However, this does not mean that there is no conflict: the conflicts are to be treated by negotiation between the providers and the users of the knowledge (note that the hierarchy of bases allows to structure the discussion by circles).

So the knowledge stored in group knowledge bases is correct, consistent and consensual. This is in great contrast with what is currently developed over the networks: databases available through WAIS, Gopher or WWW do not have to be consensual nor consistent. Data are provided as such, without any warranty of consistency from the provider, and the retrievers have to make it consistent before introducing it into their own databases. Other frameworks are also different in this respect: neither the knowledge sharing (when the opportunity to modify the shared knowledge is considered) nor the software agents provide any warranty about the consistency of the knowledge they provide.

Difficulties with the cooperation protocol can arise from the fact that it must be closely suited to the needs of cooperative knowledge base building. Otherwise, subscribers would work around it. The ideal protocol should be mechanically interpretable (at least at the performative level) and rich enough for covering adequately all of the needs. The first requirement has been successfully achieved through the design of an automatic group base protocol. However, our present protocol is very simple (only 35 rules). It must be enhanced through experience and some assumptions may have to be relaxed. For instance, it does not take into account the demand from a reviewer to clarify some point.

The CO_4 protocol has some limitations since the communication protocol is very restrictive. These limitations are usually found in the manual peer review process (however, the same protocol does not prevent people from submitting to, reviewing for and reading scientific journals). There is still debate for knowing if peer review for scholar literature is the correct behaviour in the electronic age (Peters, 1995) and the answer will not be given here. The issue of authentication of the contributions could be ensured by the obligation of citing the sources in the hypertext annotations; this should be facilitated by the availability of a citation editor in CO_4 and a contribution tracing mechanism (authentication of who proposed a modification and who issued an alternative proposal). Other enhancements are deadlines — for returning answers, etc. (Winograd, 1987) — penalties — for these reviewers who take a very long time for reviewing (for instance, suspension of their submissions) — and an anonymity management system which provides both independence and recognition to reviewers (Stodolsky, 1990).

The knowledge base architecture presented above includes two particularly interesting design choices: it is the same for all of the bases, so they can be made out of the same software packages, and it is described in a modular fashion, so that it can be used with a very raw system or a very complex one. This leaves the door open for a trade-off between the complexity and power of each algorithm used. The CO_4 protocol development should take advantage of this since it allows to start with very simple components and to enhance them progressively.

For instance, the knowledge repository can be a simple hypertext system which identifies nodes by a reference and raises an error when two nodes have the same reference (the simplest system is a text repository with the UNIX “diff” and “patch” utilities). The *degré zéro* of change control consists in routing error messages to the user, in order to have the error corrected.

Nonetheless, the protocol can be easily replaced by another one, based for instance on a vote of the majority of the subscribers, on the intersection of all of the knowledge bases or on economic decision making (for determining the introduction or not of knowledge into the base). The protocol is also designed independently of the content and language of the knowledge base. This enables to use it for different pieces of knowledge (hypertext or classes as well as tasks or equations). One could imagine to refine the protocol for dealing specifically with these expressions. However, the specific aspect is actually assigned to the revision module which issues the reports.

As a summary, the architecture and the protocol provided above are independent of the knowledge repository and the modules which manage it.

5. RELATED WORKS

An extensive comparison with other works and situation can be found in (Euzenat, 1995a). Knowledge acquisition and corporate memory is more precisely considered here.

Corporate memory is not a clearly defined word. For some people, it consists in integrating the information system of an enterprise. This is usually carried out with the help of software which allow to publish the databases and software for accessing these publications (Huhns, Jacobs, Ksiezyk, Shen, Singh, and Cannata, 1993; Collabra, 1996). The knowledge is usually modified by a unique authorised person and consistency is not ensured. However, there is no acknowledged conflict in such systems. For others, corporate memory essentially contains the memory of how the enterprise works (Durstewitz, 1994; Conklin, 1996). This includes structure, work flow, information pathways and interaction protocols.

These approaches are clearly useful. The former allows the wide dissemination of the information — as does the HYTROPES system — but it is not sufficient for ensuring the cohesion of the knowledge and its acceptance by other people. The latter has the advantage of considering the context of knowledge production and use. This is clearly a very important topic complementary to the present work. Paying attention to the socio-psychological effects of such a memory should greatly facilitate its acceptance. However, we did not focus on that particular aspect.

The present work is concerned with a third approach which considers that the corporate memory must contain the knowledge which underlies the behaviour and work of individual people. This knowledge must be the common understanding of the people in a corporation. It is thus more related with the acquisition of a shared ontology (instead of a system for sharing an ontology otherwise built).

The CO4 principle is similar to that of the SHADE project (Gruber, Tenenbaum, and Weber, 1992) which builds a knowledge medium to support the collaborative design of an artefact; in CO4 the artefact is the knowledge base. Such a system must enforce both the consistency and the agreement of everyone (human or software agent) involved into the design process. However, there are several technical differences between both systems:

- The SHADE project uses pre-existing knowledge bases (in the knowledge sharing fashion). It thus puts less emphasis on knowledge base revision than CO4.
- In SHADE, the involved people can be very different while in CO4 all bases are equal (peers) and play different roles (submitter, reviewer, etc.) depending on the situation. The variety of agents constrains to take into account a variety of behaviours (which changes to notify, etc.) and requires tools for expressing how to deal with them (publication of interest) which have not been considered here.

In the continuation of the SHADE project is the ontology server (Farquhar, Fikes, Pratt, and Rice, 1995). This system offers similar features (on a larger scale) as CO4. Both systems share the same goal (cooperative construction of consensual "ontologies") through the same means (object-based representation linked to hypermedia, availability and manipulation through the WWW). The ontology server is more advanced than CO4 on the matter of access restriction, session management and, above all, world-wide use. However, consensus in the ontology server is a wish while in CO4 it is based on a formal protocol.

ICM (Fruchter, Clayton, Krawinkler, Kunz, and Teicholz, 1993) is another related system which uses the cycle "propose-interpret-criticise-explain" for the same purpose: confronting knowledge. The system is dedicated to architectural design and tries to make people of different professions communicate through graphic layout. The system aims at filling the gap between very expressive CAD drawing tools lacking semantics and formal knowledge base systems lacking intelligible drawing capabilities. So, the drawings are translated into symbolic representations that the system can manipulate. It can also elaborate critics (when confronting two viewpoints) and report these critics on the display. It thus differs from CO4 since (1) it considers that the drawings can be translated to a symbolic level and back and (2) it provides the critics by itself while CO_4 lets the users generate an important part of them. The two points are related since a critic can only be developed on formal requirements while CO_4 accepts totally informal knowledge. In summary, ICM focuses on informal communication of formal knowledge.

At another extremity (formal protocol for informal knowledge communication) CO_4 shares some features with the coordinator (Winograd, 1987). The coordinator is a system which allows people to submit requests and offers to others who can answer by declining, accepting or proposing an alternative. The system is able to store these proposals and to manage the state of each proposal. The main differences lie in the goal of building a consensual knowledge base (common to each individual), the formal treatment that CO_4 can apply to knowledge and the hierarchical construction of knowledge bases (and hence of the communication) instead of peer-to-peer communication. The CO_4 protocol also resembles argumentation protocols like IBIS (Kunz and Rittel, 1970). However, two important differences must be noted: the proposals are not questions but answers that someone wants to integrate in a base and CO_4 does not primarily aim at recording the argumentation process but the result of the process. Meanwhile, the IBIS framework could be used for structuring the discussion between subscribers and recording the decision rational.

Another system has been set up for supporting the peer review process (Mathews and Jacob, 1996). It supports the material aspects of the process and corresponds to the base protocol here. In fact, it sticks closely to the real process in which the peers are not so preoccupied by constructing a common memory (or artefact).

CONCLUSION

The presented system addresses three particular problems of building a repository of corporate knowledge: promoting formal expression (and consistency), allowing links with informal documentation and enforcing consensus. There are obviously other concerns (e.g. hierarchy, power) which have not been addressed but deserve attention.

The originality of the system are mainly:

- the use of a formal knowledge representation to collect knowledge and corresponding tools for manipulating it;
- its immediate access through the WWW (and thus the internal network) and its connection through other kinds of documentation;
- its formal protocol for integrating (formal and informal) knowledge which enforces discussion and consensus between people.

It is expected that such an architecture is able to promote a corporate memory as a useful reference tool rather than the “shelve of the last chance”.

The system presented here is either implemented in separate parts (TROPES and HYTROPES are separated from the task system) or under implementation (behavioural knowledge or the CO_4 protocol). The implemented parts have been used in molecular biology projects (representation of genetic regulation in *D. melanogaster* for HYTROPES and sequencing projects for the task system). Achieving a working CO_4 system requires more investigation on the aspect of formal revision and the full and modular implementation of the CO_4 protocol. However, such a system is not an end and it will have to prove its usefulness through real-life experiments. Beside pursuing work on molecular biology, the construction of a corporate memory from actual design processes is under way and the CO_4 protocol will be tested on collaborative paper writing.

ACKNOWLEDGEMENTS

This research has been partially supported by GREG (Groupement de Recherches et d’Études sur les Génomes) and by GdR CNRS «Informatique et Génomes» (CNRS: Centre National de la Recherche Scientifique). Many thanks to Michel Page who carefully read a first version of the paper.

REFERENCES

Bolognesi, T., and Brinksma, E. (1987). Introduction to the ISO Specification Language LOTOS, *Computer networks and ISDN systems* 14(1):25-59

Collabra Software (1996). Internet collaboration starting with Collabra share, White paper, Collabra, Mountain View (CA US)
[http://www.collabra.com/articles/inetcom.htm]

Conklin, J. (1996). Designing organisational memory: preserving intellectual assets in knowledge economy, White paper, Corporate memory systems, Austin (TX US), 1996
[http://www.cmsi.com/business/info/pubs/desom/index.htm]

Crampé, I., and Euzenat, J. (1996). Révision interactive dans une base de connaissance à objets, Proc. 10th RFIA, Rennes (FR), pp615-623
[ftp://ftp.inrialpes.fr/pub/sherpa/publications/crampe96a.ps.gz]

Durstewitz, M. (1994). Newsletter on corporate memory, Internal memo, Eurisko, Toulouse (FR)
[http://www.delab.sintef.no/MNEMOS/external-info/cm-eurisko.txt]

Euzenat, J. (1995a). Building consensual knowledge bases: context and architecture, in Mars, N. (Ed.), Building and sharing large knowledge bases, pp143-155, Amsterdam (NL): IOS press
[ftp://ftp.inrialpes.fr/pub/sherpa/publications/euzenat95a.ps.gz]

Euzenat, J. (1995b). Building consensual knowledge bases: protocol, Internal report, INRIA Rhône-Alpes, Grenoble (FR)

Euzenat, J. (1996). Knowledge bases as Web page backbones, Proc. 5th WWW workshop on «artificial intelligence-based tools to help W3 users», Paris (FR)
[http://www.inrialpes.fr/sherpa/papers/euzenat96a.html]

Farquhar, A., Fikes, R., Pratt, W., and Rice, J. (1995). Collaborative ontology construction for information integration, Research report 63, Knowledge system laboratory, Stanford university, Stanford (CA US)
[ftp://ksl.stanford.edu/pub/KSL_Reports/KSL-95-63.ps]

Finin, T., Fritzson, R., MacKay, D., and MacEntire, R. (1994). KQML as an agent communication language, Technical report CS-94-02, University of Maryland, Baltimore (MD US) (rep. in proc. 3rd CIKM, Gaithersburg (ML US), 1994)
[ftp://ftp.cs.umbc.edu/pub/ARPA/kqml/papers/cikm.ps]

Fruchter, R., Clayton, M., Krawinkler, H., Kunz, J., and Teicholz, P. (1993). Interdisciplinary communication medium for collaborative conceptual building design, Proc. 2nd conference on the application of artificial intelligence techniques to civil engineering, Edinburgh (GB), pp7-16
[ftp://cdr.stanford.edu/pub/CDR/Publications/Reports/ICM.ps]

Gaines, B. (1990). Knowledge-support systems, *Knowledge based systems* 3(4):192-203

Gaines, B., and Shaw, M. (1995). WebMap: Concept Mapping on the Web, Proc. 4th WWW conference, Boston (MA US)
[http://ksi.cpsc.ucalgary.ca/articles/WWW/WWW4WM/]

Gruber, T., Tenenbaum, J., and Weber, J. (1992). Toward a knowledge medium for collaborative product development, in Gero, J. (Ed.). Proc. 2nd. international conference on artificial intelligence in design, Pittsburg (PA US), pp413-432
[ftp://ksl.stanford.edu/pub/knowledge-sharing/papers/shade.ps]

Huhns, M., Jacobs, N., Ksiezyk, T., Shen, M.-W., Singh, M., and Cannata, P. (1993). Integrating enterprise information models in CARNOT, Proc. 1st international conference on intelligent and cooperative information systems, Rotterdam (NL), pp32-42

Hüser, C., Reichenberger, K., Rostek, L., and Streitz, N. (1995). Knowledge based editing and visualization for hypermedia encyclopedias, *Communication of the ACM* 38(4):49-51

Karp, P., and Mavrovouniotis, M. (1994). Representing, analyzing and synthesizing biochemical pathways, *IEEE Expert* 9(2):11-22 [http://www.ai.sri.com/pubs/papers/Karp94-11:Representing/document.ps.Z]

Kuntz, W., and Rittel, H. (1970). Issues as elements of information systems, research report 131, Institute of urban and regional development, University of California, Berkeley (CA US), 1970

Mathews, G., and Jacob, B. (1996). Electronic management of the peer review process, *Computer networks and ISDN systems* 28(7-11):1523-1538

Médigue, C., Vermat, T., Bisson, G., Viari, A., and Danchin, A. (1995). Cooperative computer system for genome sequence analysis, Proc. 3rd ISMB, Cambridge (GB), pp249-258 [ftp://ftp.inrialpes.fr/pub/sherpa/publications/medigue95a.ps.gz]

Overton, C., Koile, K., and Pastor, J. (1990). GeneSys: a knowledge management system for molecular biology, in Bells, G., Marr, T. (Eds.). Computers and DNA, pp213-239, Reading (MA US): Addison-Wesley

Perrière, G., and Gautier, C. (1993). ColiGene: object-centered representation for the study of *E. coli* gene expressivity by sequence analysis, *Biochimie* 75(5):415-422

Peters, J. (1995). The hundred years war started today: an exploration of electronic peer review [http://www.mcb.co.uk/literati/articles/hundred.htm]

Rechenmann, F. (1993). Building and sharing large knowledge bases in molecular genetics, in Mars, N. (Ed.), Building and sharing large knowledge bases, pp291-301, Amsterdam (NL): IOS press [ftp://ftp.inrialpes.fr/pub/sherpa/publications/rechenmann93a.ps.gz]

Rees, O., Edwards, N., Madsen, M., Beasley, M., and McClenaghan, A. (1995). A Web of Distributed Objects, Proc. 4th WWW conference, Boston (MA US), pp75-87 [http://www.ansa.co.uk/ANSA/ISF/wdistobj/Overview.html]

Riva, A., and Romani, M. (1996). LispWeb: a specialized HTTP server for distributed AI applications, *Computer networks and ISDN systems* 28(7-11):953-961

Schreiber, A. , Wielinga, B., and Breuker, J. (Eds.) (1993). KADS: A principled approach to knowledge-based system development, London (GB): Academic Press

Sherpa project (1995). TROPES 1.0 reference manual, Internal report, INRIA Rhône-Alpes, Grenoble (FR) [ftp://ftp.inrialpes.fr/pub/sherpa/rapports/tropes-manual.ps.gz]

Stefik, M. (1986). The next knowledge medium, *AI magazine* 7(1):34-46

Stodolsky, D. (1990). Consensus journals: invitational journals based upon peer consensus, *Datalogiske Skrifter* 29

Weld, D. (Ed.) (1995). The role of intelligent systems in the national information infrastructure, Technical report, AAAI [http://www.aaai.org/Publications/TechReports/Papers/nii.html]

Winograd, T. (1987). A language/action perspective on the design of cooperative work, *Human-computer interaction* 3:3-30