



Construction collaborative de bases de connaissance et de documents pour la capitalisation

Jérôme Euzenat

► To cite this version:

Jérôme Euzenat. Construction collaborative de bases de connaissance et de documents pour la capitalisation. Manuel Zacklad, Michel Grundstein. Ingénierie et capitalisation des connaissances, Hermès Science publisher, pp.25-48, 2001, 2-7462-0234-4. hal-00822908

HAL Id: hal-00822908

<https://hal.inria.fr/hal-00822908>

Submitted on 15 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

2. Construction collaborative de bases de connaissance et de documents pour la capitalisation¹

Jérôme Euzenat*

L'activité de « mémoire technique » est destinée à recevoir la connaissance technique utilisée par les ingénieurs de l'entreprise. Ces mémoires techniques participent de la problématique de la gestion des connaissances (“knowledge management”) en ce qu'elles permettent d'accroître les capacités de capitalisation et de gestion de la connaissance et des expériences au sein des entreprises. Une telle mémoire se doit d'être vivante si elle doit être utilisée ou enrichie. Elle doit donc être cohérente et intelligible. À cette fin, trois principes sont mis ici en avant : la mémoire technique doit être autant que possible formalisée, elle doit être liée aux sources de connaissance informelle, elle doit exprimer le consensus d'une communauté.

Ces principes sont illustrés ici par leur mise en œuvre dans le prototype CO4 que nous avons développé.

Mots-clef : mémoire technique, mémoire d'entreprise, serveur de connaissance, consensus, éditeur de connaissance, world-wide web, TROEPS, CO₄, protocole de collaboration.

2.1. Motivations

Le développement de mémoires d'entreprises a plusieurs raisons bien connues : des collaborateurs quittent l'entreprise avec leurs savoir-faire, le partage du travail se développe, la traçabilité de l'information est exigée et alors que les professions sont de plus en plus spécialisées et le travail plus collaboratif, il devient difficile d'établir des conventions communes [DIENG& 2000, O'LEARY 1998]. Dans ce chapitre, la mémoire d'entreprise est considérée comme un conteneur de

¹ Chapitre 2 de l'ouvrage : Manuel Zacklad, Michel Grundstein (éds.), « Ingénierie et capitalisation des connaissances », pp25-48, Hermès, Paris (FR), 2001 (il s'agit de la version originale, non manipulée par l'éditeur).

* INRIA Rhône-Alpes, 355 avenue de l'Europe, 38330 Montbonnot Saint-Martin, France, Jerome.Euzenat@inrialpes.fr

connaissance et de savoir-faire d'un ensemble d'individus travaillant dans une entreprise particulière. Il réalise les fonctions de compilation et transformation de connaissance [WIIG 1997]. Cette vision est discutée plus loin (voir §2.4). La cible est la mémoire technique — ou mémoire de domaine [SIMON 1997] — rassemblant une connaissance technique sur les domaines d'activités manipulées.

Les grandes entreprises échangent déjà massivement des données avec leurs sous-traitants ou leurs distributeurs. Des consortium regroupant plusieurs laboratoires et entreprises échangent massivement de l'information. Celle-ci peut être utilisée afin de construire et de maintenir un serveur encyclopédique (dont le contenu peut être fait de texte et d'image par exemple) à propos d'un domaine particulier. À cette fin, le groupe peut être implémenté à l'aide d'agents logiciels dont le but est de collecter les données et de les distribuer à ceux qui sont habilités à les consulter (une telle description n'est pas éloignée des serveurs Web et Lotus Notes ne fonctionne pas autrement). Pour cela l'outil informatique peut apporter son concours sur plusieurs points qui transforment le phénomène de mémorisation en un phénomène de médiation :

- médiation entre les auteurs de la mémoire technique et les utilisateurs, bien entendu, qui peut faire plus que régurgiter la connaissance mémorisée ;
- médiation de la connaissance d'un auteur après l'avoir manipulée et en avoir détecté les incohérences ou les structures cachées ;
- médiation entre les différents auteurs de la mémoire technique pour leur permettre d'articuler leur savoir de manière cohérente.

Cette vision de l'ordinateur-médiateur ("computer as medium") peut être renforcée et étendue à la connaissance elle-même. Cependant, exprimer la connaissance sous forme simplement numérique n'est pas satisfaisante parce que cela n'encourage pas la communication entre les intervenants ni l'analyse de la connaissance. L'approche de la mémoire technique présentée ici est nourrie de notre expérience de la construction de bases de connaissance (la plupart des exemples sont tirés de la base KNIFE en génétique moléculaire [EUZENAT& 1997]). Trois principes la gouvernent [EUZENAT 1996] :

1. Le premier principe stipule que les mémoires techniques seront d'autant plus utiles qu'elles seront exprimées dans un langage formel. L'avantage d'exprimer la connaissance dans un langage doté d'une sémantique précise, respectée par l'ordinateur, est que celui-ci peut la manipuler et réaliser des vérifications. Ceci recouvre les tests de consistance et de complétude mais aussi de généralité, voire de proximité. Cependant, tout ne peut être formalisé à ce point et, même si cela est possible, les systèmes formels peuvent souffrir de limitations sérieuses (comme la complexité et l'incomplétude).

2. Le second principe exige que la colonne vertébrale de connaissance formelle puisse être le support de la connaissance informelle (texte, images, sons, animations...) qui lui est liée. Ainsi, la connaissance qui n'a pas atteint un état formalisé peut être présente, les commentaires à propos de la production de connaissance et les explications informelles peuvent être liées au corpus formel.
3. Le troisième et très fort principe est que les intervenants doivent être soutenus dans la discussion à propos de la connaissance introduite dans la mémoire. Dans cette perspective, exprimer, réutiliser, diffuser et maintenir la connaissance doit être une activité impliquant tous les acteurs (utilisateurs et producteurs de connaissance). La mémoire ne sera utilisée que si les utilisateurs comprennent sa signification et sont assurés de sa cohérence. Pour cela, le consensus entre les différents intervenants sera maintenu.

Le système CO₄ (pour Construction coopérative de bases de connaissance consensuelles [EUZENAT 1995]), qui est présenté ici, est une infrastructure permettant, à plusieurs agents, de construire des bases de connaissance représentant leur consensus à propos d'un domaine à modéliser. CO₄ cherche à supporter ces trois principes. Il est composé de deux composants indépendants : le serveur de bases de connaissance TROEPS et le protocole de CO₄.

Le système de représentation de connaissance TROEPS est utilisable en tant que serveur de connaissance à l'aide d'un client HTTP, mais permet également, aux personnes autorisées, d'éditer directement les bases de connaissance [SHERPA 1995]. Afin d'atteindre les objectifs d'intelligibilité, les bases de connaissance peuvent être reliées directement à d'autres sites Web, à des bases de données ou à un lexique. L'utilisabilité est encore renforcée par l'intégration au sein de TROEPS d'un mécanisme de révision aidant l'utilisateur à corriger ses erreurs et d'un système d'argumentation assisté par ordinateur permettant de discuter les révisions à plusieurs.

Le troisième principe nécessite le développement de recherches et d'outils nouveaux pour la construction coopérative de bases de connaissance qui permettent de renforcer la cohérence des mémoires ainsi mises en œuvre. Le protocole de CO₄ [EUZENAT 1997] est une bibliothèque logicielle qui peut être connectée à un système de représentation d'un artefact à concevoir (ici, à TROEPS en tant que système de représentation de connaissance). L'originalité principale du protocole est d'encourager la collaboration entre utilisateurs en imitant la soumission d'articles à une revue scientifique, c'est-à-dire en promouvant une évaluation par les pairs des modifications soumises à une base consensuelle. Afin de rendre ceci possible, les utilisateurs ne modifient pas directement la base de connaissance consensuelle, mais travaillent sur un espace de travail personnel et, lorsqu'ils considèrent avoir atteint un état publiable, soumettent leurs modifications à la base consensuelle à laquelle ils

adhèrent. Le protocole prend en charge cette soumission et organise un appel à commentaires entre l'ensemble des adhérents à la base consensuelle. Le résultat peut être l'un des suivants : soit tous les adhérents acceptent la modification et celle-ci est intégrée dans la base consensuelle, soit un des adhérents refuse la modification et celle-ci est rejetée, soit un des adhérents fait une contre-proposition en acceptant une partie et en remplaçant une autre partie, auquel cas un appel à commentaire modifié est émis.

Le but du troisième principe est d'encourager la discussion et le consensus autour de la connaissance exprimée pendant que les producteurs sont encore présents plutôt que de s'empresse de stocker la connaissance brute, non vérifiée et non partagée pour se rendre compte lors de son utilisation qu'elle est inutilisable. La contrainte d'unanimité est réellement coopérative, elle n'a pas pour but de donner de bonnes et de mauvaises notes mais de forcer les différents souscripteurs à réellement discuter et élucider leurs conflits plutôt qu'à trouver un moyen terme peu productif [BUCKINGHAM SHUM 1997]. Le fait que chaque utilisateur travaille dans une base de connaissance privée (et que les bases consensuelles peuvent, à leur tour, souscrire à d'autres bases consensuelles) permet de ne pas bloquer prématurément un processus d'élaboration (ou le ralentir du fait de la longueur possible du processus de vote).

Le but de ce chapitre est de donner une vision globale du système proposé et des difficultés rencontrées. Des présentations plus précises de chacun des éléments — TROPES [SHERPA 1995], les mécanismes de révision [CRAMPÉ& 1998] et le protocole de CO4 [EUZENAT 1997] — peuvent être trouvées ailleurs. Le point important d'un tel dispositif est qu'il implémente les principes mis en œuvre plus haut. Il ne doit pas être pris comme un système clef en main destiné à résoudre tous les problèmes soulevés par la mémoire technique : les autres chapitres sont là pour rappeler que le domaine est fort complexe.

Les modèles de représentation de connaissance utilisés par le système sont d'abord présentés (§2.2) avant de considérer l'aspect serveur et éditeur de connaissance de TROEPS (§2.3). Puis le protocole de collaboration de CO4 est présenté dans son utilisation à la construction de bases de connaissance (§2.4). Les solutions proposées sont comparées à d'autres systèmes et principes mis en œuvre dans la littérature.

2.2. Modèles de représentation

Le modèle de représentation utilisé par CO4 est la représentation de connaissance par objets TROEPS (§2.2.1). Il offre un formalisme de représentation de connaissance permettant d'implémenter le premier principe. La liaison entre les éléments du langage à un lexique et au World-wide web est ensuite brièvement présentée (§2.2.2).

2.2.1 Objets

TROEPS [MARIÑO& 1990, SHERPA 1995] est un système de représentation de connaissance par objets, c'est-à-dire un système de représentation de connaissance inspiré par les langages de frames et les langages de programmation par objets [EUZENAT 1998].

Un *objet* est un ensemble de couples attributs-valeurs associé à un identifiant. En général cette identification se fait à l'aide d'un nom (que celui-ci soit extérieur aux couples attributs-valeurs ou la valeur d'un attribut particulier). La valeur d'un attribut peut soit être un objet, soit être une valeur d'un type primitif du langage (par exemple, une chaîne de caractères ou un entier). Elle peut être connue ou pas. Souvent les attributs peuvent contenir une collection (par exemple, un ensemble) de valeurs. Les objets, en TROEPS, sont séparés en *concepts* étanches qui déterminent l'identité (par une clef) et la structure (par la définition des attributs et de leur type) des objets. Un objet ne peut changer ni de concept ni de clef.

Les objets peuvent être vus sous différents *points de vue* qui correspondent chacun à une taxonomie de classes différente (voir figure 2.3). Dans chaque point de vue, un objet peut être attaché à une classe. Les *classes*, à l'instar de celles des langages de programmation par objets, permettent de regrouper des traits communs aux objets qui leur sont attachés. Elles contraignent les valeurs des attributs à l'aide de descripteurs. Contrairement au modèle original des schémas ("frames"), seules les classes associent des facettes (ou descripteurs) aux attributs et seul un ensemble restreint de *descripteurs* est autorisé et clairement identifié. Les facettes servent principalement à deux objectifs : préciser les valeurs d'attributs admissibles dans une classe (*typage*) et déclarer des mécanismes capables de déduire la valeur d'un attribut manquant (*inférence*).

Les facettes de typage permettent de restreindre les valeurs possibles d'un attribut en précisant les valeurs admissibles par une énumération (*domaine*) ou une réunion d'intervalles (*intervalles*), introduisant des restrictions (*sauf*) ou restreignant la cardinalité dans le cas de collections (*cardinal*). Lorsque la valeur d'un attribut peut être un autre objet, les restrictions peuvent se faire en précisant les classes auxquelles l'objet doit appartenir (*type*). La Figure 2.1 présente deux définitions de classes.

```

<gène
  attributs = {
    <nom type = chaîne; constructeur = un;>,
    <synonymes type = chaîne; constructeur = ensemble;>,
    <protéine type = protéine; constructeur = un;>,
    <séquence type = séquence-adn; constructeur = ensemble;>,
    <taille-de-l-unité-de-transcription
      type = entier;
      constructeur = un;
      intervalles = { [10 +inf] };>
  };>

<gène-polarité-de-segment
  sorte-de = gène-structurel;
  attributs = {
    <protéine type = { protéine-structurelle };>;>

```

Figure 2.1: Une classe et une sous-classe. Une classe est décrite par les ensembles de valeurs que peuvent prendre ses attributs (ici nom, synonymes, protéine, séquence et taille-de-l-unité-de-transcription). Ces ensembles de valeurs sont spécifiés par un constructeur (un, liste ou ensemble) et le type de la valeur qui peut être un type de base (entier ou chaîne) ou une autre classe (protéine ou séquence-adn). Les valeurs admises peuvent être précisées comme étant prises dans un ensemble d'intervalles ($[10 +inf]$) ou comme appartenant à une sous-classe (protéine-structurelle).

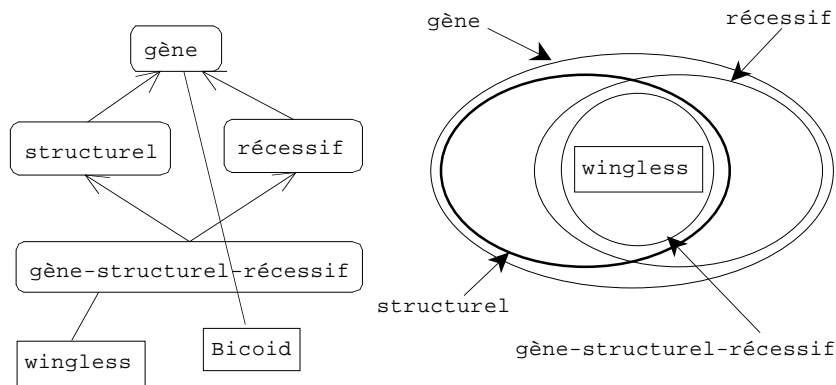


Figure 2.2 : Signification des relations entre classes. Les boîtes rectangulaires représentent les objets, les boîtes arrondies les classes. Les traits sans flèche représentent l'instanciation et les flèches vides la spécialisation. Le dessin de droite est une représentation ensembliste de la situation. Il faut noter que *Bicoid* n'y figure pas car s'il appartient à l'ensemble *gène*, rien n'est connu de son appartenance à un de ses sous-ensembles (il y a donc plusieurs interprétations possibles suivant les positions de *Bicoid* dans le diagramme).

Les classes sont organisées en une *taxonomie* par la relation de *spécialisation* (parfois nommée relation d'héritage ou de généralité). La représentation de connaissance par objets privilégie donc ce lien particulier entre les classes. La spécialisation est une relation d'ordre. Dans certains systèmes, le graphe de la réduction transitive de cette relation est restreint à un arbre, contraint à être connexe ou à disposer d'une unique source (racine dans le cas d'un arbre). La sémantique de cette relation est celle de l'inclusion ensembliste : les individus appartenant à l'interprétation d'une classe doivent appartenir à celle de ses super-classes. Les taxonomies peuvent donc être interprétées comme un ensemble de sous-ensembles imbriqués (voir figure 2.2).

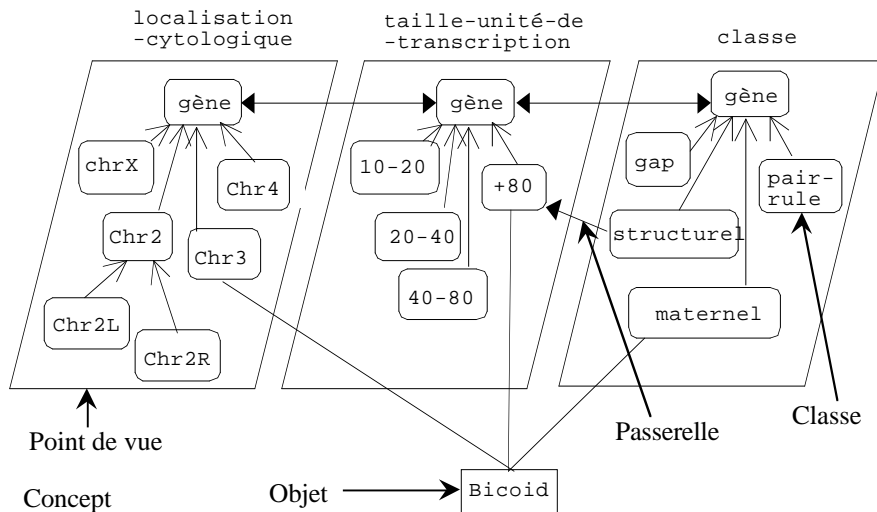


Figure 2.3 (d'après la base *KNIFE*) : Les différentes sortes d'entités de TROEPS. Le concept *gène* peut être vu sous les points de vue *localisation-cytologique*, *taille-unité-de-transcription* et *classe* (de *gène*). Chacun d'entre eux détermine une hiérarchie de classes dont la racine est nommée *gène*. Par exemple, sous le point de vue *localisation-cytologique*, l'ensemble des gènes est décomposé selon leur chromosome. L'objet *Bicoid* est attaché à la classe *chr3* sous le point de vue *localisation-cytologique*, à la classe *+80* sous le point de vue *taille-unité-de-transcription* et à la classe *maternel* sous le point de vue *classe*.

L'intérêt de la représentation de connaissance par objets est de permettre de nombreuses modalités de manipulation de connaissance parmi lesquelles :

— Les requêtes de type filtrage (qui retrouvent les objets d'un concept satisfaisant certaines contraintes sur ses attributs ou ses classes d'attachement) et

recherche d'objets similaires (en fonction des valeurs d'attributs ou des classes d'appartenance) impliquant une mesure de distance.

— L'inférence de valeurs d'attributs à l'aide de valeurs par défaut, d'attachement procédural (où un programme est invoqué pour calculer la valeur), de passage de valeur (où la valeur provient directement de la valeur d'un autre attribut) ou de filtrage (où la valeur est l'ensemble des objets d'une classe satisfaisant certaines contraintes). Celle-ci est secondée par l'héritage qui permet de propager ces méthodes d'inférence des classes générales aux classes plus spécifiques.

— L'inférence de position (classification et identification) où les places possibles d'un objet (figure 2.4) ou d'une classe sont cherchées dans une taxonomie préalablement établie. Ce type de fonction est prolongé dans TROEPS par l'inférence de taxonomies qui construit une taxonomie à partir d'un ensemble d'objets (figure 2.5).

2.2.2 Textes et lexiques

En plus de son modèle d'objets à proprement parler, TROEPS dispose d'une représentation de lexique permettant à chaque terme d'être décrit par sa définition privilégiée mais aussi par ses synonymes, antonymes, hyponymes et hyperonyme. Par ailleurs, les entités de TROEPS peuvent être automatiquement indexées par le lexique (en fonction des identificateurs qu'elles mettent en jeu). Ainsi, il est possible de naviguer des objets au lexique et du lexique aux objets (par exemple, si le terme retenu au sein d'une base particulière n'est pas connu, il est possible de le retrouver à l'aide des synonymes et d'accéder directement aux objets impliqués).

Nous avons récemment développé une interface avec Xterm, un extracteur de terminologie à partir de texte (ici des spécifications) [CERBAH& 2000]. Dans celui-ci, non seulement l'extracteur peut construire une terminologie mais il va aussi créer les classes correspondantes que l'utilisateur juge pertinent tout en maintenant le lien entre terminologie et objets ce qui permet d'assurer la traçabilité entre textes et objets engendrés par l'intermédiaire de la terminologie.

Enfin, les objets TROEPS (représentation formelle) peuvent être annotés informellement. C'est ainsi que les objets sont liés aux termes de la terminologie. Ces annotations peuvent être de n'importe quel type mais ce sont souvent des URLs permettant d'identifier des ressources distantes comme des textes à lier aux objets. Ces URL sont surtout utiles dans le contexte du Web, lorsque TROEPS est utilisé comme serveur de connaissance.

2.3. Serveur de connaissance

Comme souligné par différents auteurs [REES& 1995, GAINES& 1995] une base d'objets est déjà un graphe d'objets reliés entre eux. La similarité avec un ensemble

de pages HTML est donc immédiate et la liaison entre les deux est aisée. Le premier intérêt du Web est qu'il permet de lier de manière naturelle la connaissance formelle et la connaissance informelle (facilitant ainsi la mise en œuvre du second principe). La connaissance contenue dans une mémoire technique peut donc être diffusée sous une forme active (§2.3.1) mais elle peut aussi être éditée de manière distance moyennant la prise en compte de quelques problèmes (§2.3.2).

2.3.1. Diffusion de la connaissance

Les bases de connaissance peuvent être utilisées comme le squelette d'un site Web dont la chair est faite d'annotations, spécifiquement rédigées en HTML. Les entités TROEPS sont accessibles via un client HTTP en invoquant leur URL normalisé et ils s'affichent comme une page HTML classique (une interface XML pour exporter vers d'autres systèmes existe aussi). Des documents (HTML, gif, etc.) référencés par ces objets peuvent aussi être automatiquement engendrés par TROEPS.

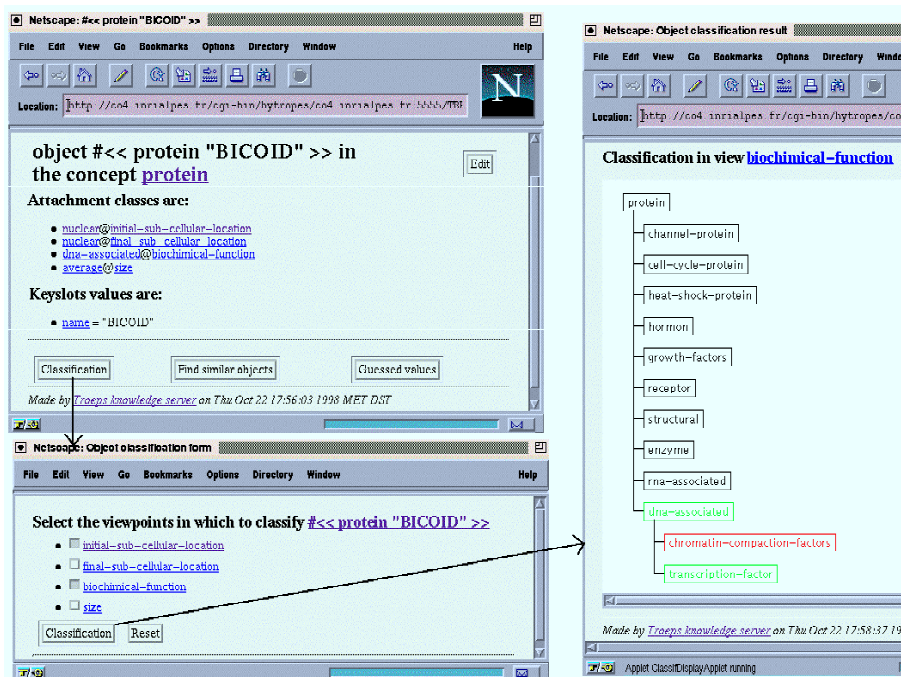


Figure 2.4: Requête de classification (KNIFE). La protéine BICOID est classée dans les points de vue initial-sub-cellular-location et biochemical-function. Le résultat est affiché sur les différentes taxonomies dans lesquelles les classes sont en rouge (chromatin-compaction-factor dans biochemical-function) lorsque la classe est impossible et en vert (transcription-factor) lorsqu'elle est possible pour l'objet.

La diffusion de la connaissance par le biais d'HTTP est déjà intéressant. Mais les bases de connaissance peuvent offrir plus. En effet, à partir d'un serveur de connaissance, il est possible de construire des requêtes structurées fondées sur le langage formel utilisé (voir figure 2.4). La réponse sera fondée sur une méthode sémantiquement fondée. Par exemple, un utilisateur recherchant une protéine dans la base de connaissance KNIFE peut sélectionner un formulaire de filtrage dans le concept protéine et remplir les domaine des champs correspondant aux critères (`transcription-factor` pour le champ `biochemical-function`) et sélectionner les classes dans lesquelles la protéine recherchée doit se trouver (la classe `average` du point de vue `size`). Il peut alors examiner les résultats de l'application du filtre et revenir à celui-ci pour étendre ou restreindre la recherche.

Un autre aspect bien connu des représentations par objets est leur capacité de classification d'un objet en fonction des valeurs de ses champs. TROEPS permet de classer un objet dans plusieurs points de vue simultanément. Le résultat de la classification est affiché dans une représentation graphique des taxonomies utilisant un code couleur pour distinguer les classes possibles des classes impossibles (voir figure 2.4).

2.3.2. Problèmes d'édition

De même qu'une base peut être consultée sur le Web, il est possible d'éditer une telle base à l'aide d'HTTP. Du point de vue de l'interface, cela dédouble les points d'entrée, du point de vue fonctionnalités, cela permet d'avoir accès à l'ensemble des primitives de manipulation des entités TROEPS mais aussi d'introduire de nouveaux services (ainsi, la classification présentée ci-dessus se transforme en un outil pour attacher les objets à des classes plus spécialisées; l'inférence de taxonomies, qui construit de nouveaux points de vue peut intervenir dans la base, voir figure 2.5).

Mais la possibilité d'éditer la base de connaissance soulève de nouveaux problèmes concernant le contrôle d'accès (qui a le droit de modifier la base), la détection des erreurs et la sécurité des données (la base étant accessible sur le Web il faut qu'elle le soit en permanence et que son contenu ne soit pas mis en péril) et la consistance de la base. Dans le cas de TROEPS, ces problèmes ont été résolus de la manière la plus simple possible.

Le contrôle d'accès est délégué au serveur HTTP. Il permet de contrôler l'accès à un site ou à un groupe de personnes via un mot de passe (ce qui est pratique pour créer un site public en consultation et privé en édition). Ceci est facilité par la structure des URL TROEPS qui séparent les fonctions d'édition des fonctions de consultations par des préfixes spécifiques.

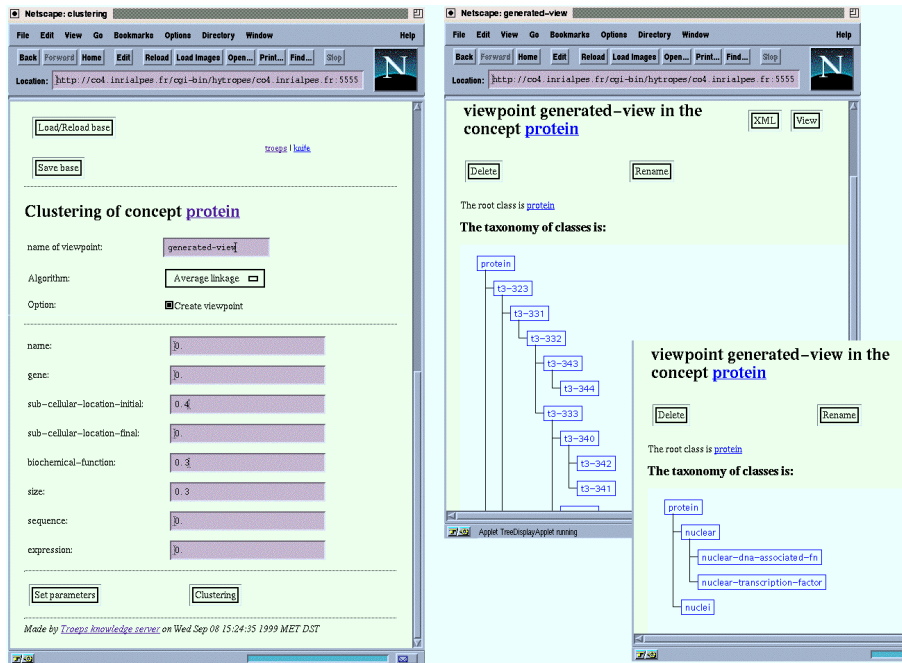


Figure 2.5: Inférence de taxonomies. L'algorithme est utilisé pour créer un nouveau point de vue qui organise les objets suivant leur similarité. Une fois ce point de vue accepté et modifié par l'utilisateur, il est intégré à la base.

La détection des erreurs est incorporée à TROEPS lui-même. Il empêche toute action menant à une inconsistante d'être exécutée. Les inconsistances peuvent être causées par un objet attaché à une classe dont il ne satisfait pas les contraintes ou une mauvaise orthographe d'un identificateur. Ces erreurs sont attrapées par le système et affichées dans le client HTTP. Un mécanisme de révision plus élaboré [CRAMPÉ& 1998] a été développé pour aider les utilisateur à corriger leurs requêtes lors d'une telle erreur.

D'autres erreurs peuvent être suscité par le simple usage du client HTTP. En effet, un utilisateur peut détruire un objet alors que son URL subsiste dans ses marque-pages ou dans le client d'un autre utilisateur. Si quelqu'un tente d'accéder à la page correspondant à un tel URL, la requête est encore une fois attrapé par le système.

Même si les problèmes évoqués ci-dessus sont résolus de manière simple, aucune politique de modification ou d'élaboration collaborative d'une telle base de connaissance n'a été présentée. À notre connaissance, il s'agit là d'un problème ouvert. La section suivante a pour but de présenter un moyen de le résoudre.

2.4. Protocole de coopération

Le but principal de CO4 est la construction d'une base consensuelle. Pour parvenir à cette fin, il propose d'organiser l'interaction entre les intervenants suivant une architecture particulière et de contrôler cette interaction à l'aide d'un protocole défini formellement. Ainsi, le troisième et dernier principe aura été mis en œuvre.

L'avantage de mettre en place une telle organisation est celle de décharger les utilisateurs de la vérification et de la maintenance du consensus et de toutes les tâches fastidieuses que cela entraîne. Le danger est d'enfermer les utilisateurs dans un schéma qui leur échappe.

Les principes qui sous-tendent le protocole sont dérivés de ceux mis en œuvre dans l'évaluation par les pairs des journaux : avant d'être introduit dans la base consensuelle, la connaissance doit être soumise et acceptée par la communauté. À l'issue du processus, la base de connaissance devrait être suffisamment contrôlée pour que les acteurs puissent l'utiliser en confiance.

Sont d'abord présentés, l'organisation des bases dans le but de contribuer à une base consensuelle et les principes du protocole pour l'intégration de connaissance (§2.4.1) et une illustration de son utilisation proposée (§2.4.2). Enfin, les propriétés satisfaites par ce protocole sont évoquées (§2.4.3).

2.4.1 Architecture

Afin de produire un artefact consensuel, les collaborateurs doivent pouvoir communiquer. L'architecture logicielle (voir figure 2.6) proposée permet de mettre en relation non pas les collaborateurs, mais leurs bases de connaissance. Les bases sont organisées en un arbre dont les feuilles sont les bases individuelles et les nœuds représentent le consensus entre leurs fils. Cette architecture ne préjuge pas de l'utilisateur des bases feuilles (ce peut être un programme ou un être humain) alors que les bases consensuelles ou bases de groupe fonctionnent sans intervention humaine. Chaque base de groupe est réputée représenter le consensus établi entre ses bases souscriptrices. L'architecture est récursive, si bien que le consensus peut être établi par exemple d'abord entre les membres d'un groupe avant d'être discuté plus largement. Des bases de connaissance indépendantes peuvent souscrire auprès des bases de groupe comme observateurs : elles recevront tout ce qui est incorporé dans les bases de groupe mais ne pourront intervenir dans la décision d'incorporer des éléments de connaissance.

L'intérêt de multiplier les bases de connaissance est d'une part d'éviter la modification entrant en conflit avec celle d'autres intervenants (comme dans ONTOLINGUA SERVER [FARQUHAR & 1997]) et d'autre part de permettre aux intervenants de poursuivre leur travail personnel sans avoir à le divulguer prématurément (implémentant ainsi les cercles de distribution privés

[SCHATZ 1992]). Ce dernier point permet de pallier certains inconvénients dus aux délais de prise de décision (voir un exemple au §2.4.3).

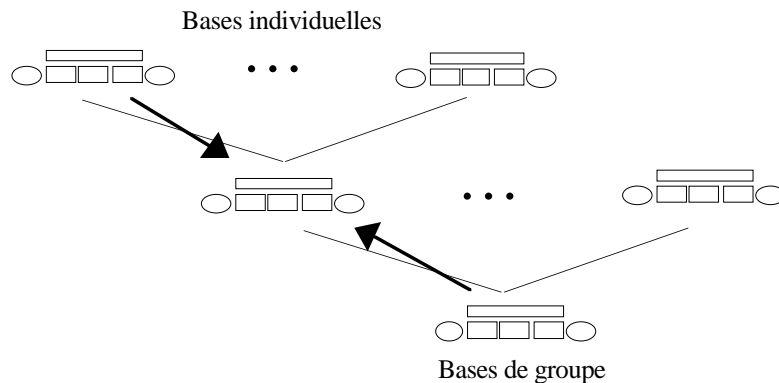


Figure 2.6 : Architecture hiérarchique et flot de messages (flèches sombres) de CO4. Les messages descendants comprennent la soumission d'une proposition, les réponses d'acceptation, de refus ou de contre-proposition. Les messages ascendants comprennent la diffusion des propositions acceptées et des appels à commentaires sur les soumissions.

Le dispositif rapidement tracé obéit à quelques principes simples qui sont résumés ci-dessous :

Uniformité : Chacun (base individuelle ou base de groupe) dispose du même outil pour représenter la connaissance. Ceci est bien entendu nécessaire à l'heure actuelle pour assurer l'intégration des contributions. À noter que le logiciel autorise l'intégration de connaissance sous forme d'objets mais aussi sous forme textuelle ou terminologique : le protocole lui-même est totalement indépendant de TROEPS.

Consistance : L'accent est mis sur la consistance entre les apports de chacun. Si dans un premier temps ceci peut être vu comme une contrainte trop forte, il faut considérer que c'est surtout une aide afin que les conflits soient explicités et discutés entre les différents intervenants et non ignorés (comme cela peut se passer dans les revues scientifiques textuelles). Même s'il semble naturel que des spécialistes aient une vision contradictoire (entre eux) d'un même domaine, il est inutile de la stocker telle quelle [EUZENAT 1996] : il vaut mieux qu'ils rendent compte des différences et si possible de leurs raisons. La conception d'une révolution reposant sur des principes complètement nouveaux [KUHN 1959] peut toujours être développée dans une nouvelle base qui sera ensuite confrontée à l'ancienne base. Enfin, le but est d'obtenir une vision cohérente du domaine considéré.

Bonne volonté : L'ensemble de cet édifice est basé sur la bonne volonté des intervenants (ils soumettent leur connaissance et leur opposition à certaines contributions) et leur bon usage (ils ne diffèrent pas des publications en ne répondant

pas aux appels à commentaires, il ne consomment pas les ressources inutilement, ils ne soumettent pas de propositions auxquelles ils n'adhèrent pas). Indépendamment de comment cette hypothèse de bonne volonté se traduit — et peut être relâchée — il est clair qu'un tel système ne peut fonctionner qu'entre des individus ayant un intérêt commun bien compris (celui de construire ensemble une base de connaissance)... comme c'est normalement le cas lorsque l'on participe conjointement à la progression de la connaissance.

Tout système permettant la construction d'un artefact doit établir une politique de modification de celui-ci. Le protocole de CO4 imite celle des comités éditoriaux : avant d'être intégrée à la base consensuelle, la connaissance doit être soumise et acceptée par la communauté. Le choix d'un tel protocole n'est pas innocent : il a montré son utilité au sein de la communauté scientifique et, dans sa version consensuelle plus que dans ses versions majoritaire ou intersective, il promeut le dialogue entre les intervenants.

Le protocole mis en œuvre diffère de celui des revues scientifiques dans le sens où la base de connaissance doit rester consistante (mais il est possible de soumettre un retrait de connaissance) et toute modification doit être acceptée par l'ensemble des parties (*consensus*). L'aspect formel de la représentation et l'exigence de consistance requièrent plus de rigueur que l'évaluation par les pairs (en effet, il n'est pas interdit dans un même journal — voire dans un même article — d'exprimer des assertions contradictoires). Ceci conduit à l'exigence de consensus dans laquelle la connaissance doit être acceptée par tous les intervenants.

2.4.2. Un exemple de soumission

Voyons plutôt un exemple concret de fonctionnement du système, lui-même accessible à l'aide d'un client HTTP (voir figure 2.7).

Un utilisateur peut souscrire à une base de groupe en émettant une requête vers cette base. Cette requête va devoir être acceptée par les autres souscripteurs dans les mêmes conditions que s'il avait soumis la connaissance. Une fois accepté comme souscripteur, il sera intégré à l'architecture de la figure 2.6, il recevra le contenu de la base de groupe (qu'il est réputé avoir accepté par sa souscription) et il sera intégré aux appels à commentaires en cours. Il sera bien entendu habilité à soumettre de la connaissance.

Lorsque le souscripteur a suffisamment confiance dans une partie de la connaissance exprimée dans sa base individuelle, il peut la soumettre à sa base de groupe. Ceci est réalisé en circonscrivant la partie à soumettre (en TROEPS une entité particulière et tout ce qu'elle contient) et en appelant la procédure de soumission (sur la partie gauche de la figure 2.7, en cliquant sur le bouton « Soumettre »). Afin de compléter la requête, le système fait la différence entre ce qui est à soumettre et ce qui est déjà dans la base de groupe et prend en compte les modifications de la

base de groupe que l'utilisateur n'a pas voulu intégrer dans sa base. Puis il transmet la requête à la base de groupe. Si la connaissance soumise est consistante avec le contenu de la base de groupe, la requête est rendue anonyme (suivant la configuration de la base de groupe) puis transmise aux autres souscripteurs sous la forme d'un appel à commentaires afin qu'ils évaluent la contribution.

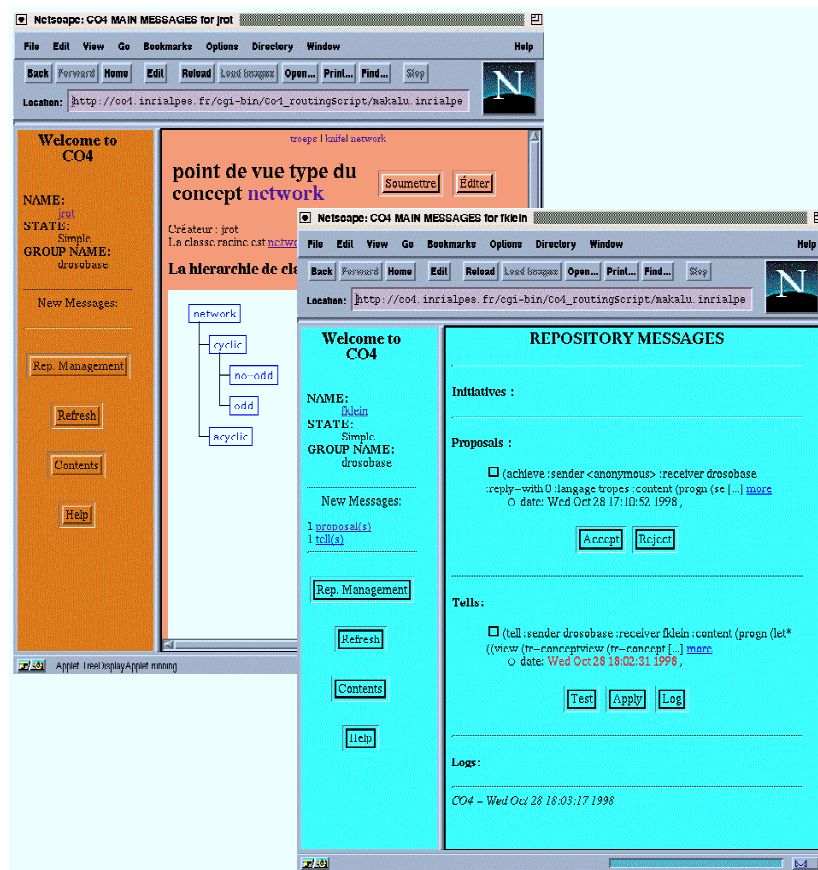


Figure 2.7 : Interface de la bibliothèque de CO4 connectée à TROEPS. La partie gauche présente le point de vue qu'une utilisatrice soumet à la base de groupe et la partie droite présente l'état des soumissions reçues par un autre souscripteur.

Cet appel à commentaires parvient à l'ensemble des souscripteurs (dans la partie droite de la figure 2.7, il apparaît sous l'en-tête "Proposals"). Les utilisateurs peuvent lire ces soumissions (en cliquant sur "More") ou les inclure dans leur propre base individuelle. Ceci peut conduire à une détection d'inconsistance ou de

redondance que l'utilisateur pourra utiliser dans sa réponse ou dans sa confection d'une contre-proposition. En réponse à cet appel à commentaire, l'utilisateur peut répondre par l'acceptation lorsqu'il estime que la proposition doit être intégrée à la base de groupe, le rejet s'il estime qu'elle ne doit pas l'être ou la contre-proposition lorsqu'il propose des modifications (boutons correspondants sur la partie droite de la figure 2.7).

Lorsque la base de groupe dispose de suffisamment de commentaires, elle intègre ou non la modification à son contenu. Quatre cas se présentent :

— Tous les souscripteurs acceptent la contribution, alors elle est intégrée et communiquée à l'ensemble des souscripteurs et observateurs.

— Un souscripteur rejette la proposition, alors l'appel à commentaire est annulé, la contribution est refusée et les commentaires sont transmis au soumissionnaire.

— Le soumissionnaire retire sa proposition durant le processus de vote ce qui conduit à la suppression de l'appel à commentaire.

— Un souscripteur soumet un compromis (où une partie de la soumission est acceptée, une partie refusée et une partie modifiée), alors tout se passe comme si la partie refusée était refusée, la partie acceptée était acceptée par ce souscripteur et la partie modifiée était une nouvelle soumission.

À noter que même si la soumission a été rendue anonyme pour le processus d'évaluation, elle est maintenant créditée à son auteur original.

Il peut cependant s'avérer que les recherches menées par l'un des souscripteurs entrent en conflit avec ce qu'il vient d'accepter. Aussi peut-il refuser d'intégrer le nouveau contenu dans sa base individuelle. Ce contenu est alors stocké dans le journal de la base individuelle (bouton "Log" sur la figure 2.7). Le fait que chacun puisse maintenir une base différente de la base de groupe permet l'exploration de voies divergentes. Cela permet aussi à la communication, la négociation et l'acceptation d'être asynchrone. En fait, cela reproduit le fonctionnement des comités de lecture : les évaluateurs peuvent prendre leur temps pour examiner une contribution avec attention parce que cela n'empêche pas les auteurs de continuer à travailler.

Ce protocole fonctionne récursivement sur la structure de l'arbre des bases décrit à la figure 2.6, mais la décision revient toujours aux bases feuilles, les bases de groupe ne faisant que relayer les requêtes de leurs bases de groupes et les réponses.

2.4.3 Propriétés

Le protocole de CO4 a été décrit formellement et certaines propriétés ont été établies [EUZENAT 1997]. Les propriétés intéressantes pour un tel protocole sont les suivantes :

- (0) que le protocole soit complètement spécifié ;

- (1) que les bases ne puissent être dans un état inconsistant ;
- (2) que n'importe quel utilisateur puisse soumettre sa connaissance n'importe quand ;
- (3) que l'acceptation (resp. le rejet) d'une proposition soit subordonnée à l'accord (resp. au non-accord) de tous les (resp. au moins un) souscripteur(s) ;
- (4) qu'une décision (accepté/refusé/retiré) pour toute proposition soit prise dans un délai fini ;
- (5) que le nombre de messages soit minimum.

La terminaison n'est pas forcément souhaitable puisque le système peut-être utilisé indéfiniment. À la place, (4) propose une propriété de terminaison locale. La propriété la plus simple à obtenir (0) est le fait qu'à chaque message corresponde une réponse.

Certains résultats concernant ce protocole dépendent de quatre hypothèses de travail. Elles sont toutes présentées en une seule fois.

HYPOTHÈSES. Le protocole est considéré ici sous les hypothèses suivantes :

- (a) il y a toujours, dans un délai fini, une réponse à une requête émise vers une base individuelle,
- (b) il n'y a pas un nombre infini de contre-propositions à une proposition,
- (c) il n'y a pas, dans un délai fini, un nombre infini de nouvelles souscriptions,
- (d) différentes propositions examinées simultanément sont indépendantes.

Chaque hypothèse sera évoquée dans le cadre de la propriété où elle est utile. Les hypothèses (b) et (c) sont des hypothèses raisonnables et ne seront pas discutées. L'hypothèse (a) est difficile à garantir dans la réalité mais elle est absolument nécessaire pour que la terminaison soit vérifiée. Ce n'est pas une propriété spécifique au protocole de CO₄ ; elle est partagée par toutes les tentatives de résoudre le "transaction commit problem" dans les systèmes de bases de données distribuées (même pour les votes à la majorité) [FISCHER& 1985]. Les temporisations ("time-out") sont les procédés habituellement utilisés pour résoudre ce problème.

L'hypothèse (d) est aussi très difficile à garantir mais nécessaire pour obtenir la consistance. Elle peut être garantie en détectant les dépendances et en retardant une requête tant qu'une décision n'a pas été prise sur toutes les autres requêtes dont elle dépend (affaiblissant ainsi la propriété d'équité). Le même problème se pose dans le cadre législatif où des amendements contradictoires peuvent être déposés sur un même texte. La procédure de vote permet de se prémunir de ce problème [ASSEMBLÉE NATIONALE 1959]. Ainsi, chaque texte de loi est examiné soit en bloc (et ne peut être amendé), soit article par article auquel cas les articles sont examinés amendements par amendements dans l'ordre d'importance décroissante (d'abord les

amendements de suppression de l'article, puis les amendements qui changent de moins en moins le texte initial et dans l'ordre de ce texte). Une disposition spéciale est par ailleurs utilisée pour des amendements contradictoires qui peuvent être examinés de concert mais toujours votés de manière ordonnée. Dès qu'un amendement est accepté sur un article, tous les amendements concurrents (ceux avec lesquels il y a inconsistance) sont considérés comme rejetés.

Les propositions suivantes ont pu ainsi être établies :

PROPOSITION (Consistance). Sous l'hypothèse (d), les bases de groupe sont toujours dans un état consistant.

PROPOSITION (Vivacité et équité). Tout souscripteur peut soumettre des propositions à sa base de groupe à tout moment.

PROPOSITION (Consensus). Toute soumission est acceptée si et seulement si tous les souscripteurs (au moment de son introduction dans la base) l'ont acceptée.

PROPOSITION (Terminaison). Sous les hypothèses (a-d) toute soumission atteint un état d'acceptée ou de rejetée au bout d'un délai fini.

Certaines propriétés ne se transfèrent pas facilement vers un protocole majoritaire ou intersectif. En effet, si l'intersection et le consensus sont des fonctions monotones décroissantes de l'ensemble des souscripteurs, ce n'est pas le cas de la majorité (ajouter des souscripteurs peut élargir ou réduire l'ensemble des propositions acceptées).

Le protocole de CO4 assure donc un schéma d'intégration du contenu des bases de type intersectif dans lequel chaque contribution fait progresser vers un modèle plus complet. En conclusion, la base consensuelle finira par être plus complète que toutes les bases individuelles initiales. Il permet donc, comme c'est le but de ce travail, un partage de la connaissance qui enrichit l'ensemble des intervenants.

2.5. Conclusions

2.5.1 Bilan

Le système CO4 présenté ici répond à trois problèmes particuliers de la construction d'une mémoire technique : promouvoir l'expression formelle de la connaissance et assurer sa consistance, lier connaissance formelle et connaissance informelle et imposer le consensus. Résoudre ces problèmes a pour but de faire des mémoires techniques d'utiles outils de référence et non des « tiroirs de la dernière chance ». Cependant un tel dispositif n'est pas une fin mais aura à faire ses preuves dans des expérimentations. Actuellement, nous n'avons pu réaliser que des

expériences partielles. Il est bien entendu que d'autres préoccupations, organisationnelles et sociales en particulier, interviennent dans la mise en œuvre de tels systèmes et que le présent chapitre est uniquement consacré aux aspects techniques.

Le système CO4 tel que décrit ici est complètement opérationnel. Ses principales originalités concernent :

- l'utilisation d'un système de représentation de connaissance formelle pour collecter la connaissance et les outils correspondants ;
- son accès à l'aide du Web et sa connexion à des sources d'information extérieures;
- son protocole formalisé pour l'intégration de la connaissance (formelle ou informelle).

Il va donc bien au-delà des outils de recherche d'information ou des systèmes de gestion de bases de données rapidement rebaptisés « outils de gestion des connaissances ». Le prototype présenté ici n'est pas un outil magique, mais nous sommes convaincu que les principes qui ont présidés à son développement feront leur chemin.

2.5.2 Travaux connexes

Les travaux considérés ici concernant principalement la gestion des connaissances (il y a d'autres travaux connexes dans les systèmes de gestion de bases de connaissance [EUZENAT 1999], les systèmes multi-agents ou le travail collaboratif [EUZENAT 1997]).

Le mot « mémoire d'entreprise » n'est pas très clairement défini [DIENG& 2000, BARTHES 1996, O'LEARY 1998] ; il a pris ici une acception précise : il ne s'agit pas de mémoire organisationnelle mais de mémoire technique, il ne s'agit pas de mémoire purement documentaire mais de mémoire formalisée. Ces deux précisions concernant le contenu sont à souligner.

En général, la mémoire d'entreprise est construite à l'aide de logiciels pour publier des données, documents et connaissance et de logiciels pour accéder ces publications [HUHNS& 1993]. Souvent la connaissance est modifiée par une unique personne autorisée [VAN HEIJST 1997, ERMINE& 1996], parfois la consistance est assurée [VAN HEIJST 1997, ALTHOFF& 1998], parfois l'inconsistance est préservée [DIENG& 2000]. La notion de conflit n'est alors pas mise en avant.

Cependant, la tendance de l'adoption massive des technologies de l'Internet, conduit à envisager des mécanismes de construction directe et collaborative [BUCKINGHAM SHUM 1997, LECHNER& 1999]. L'avantage de cette approche est d'impliquer tous les intervenants dans la construction de la mémoire et, par conséquent, d'assurer son évolution. À l'instar d'un site Web, la maintenance d'une

telle mémoire (et donc la vie de l'organisation qui devrait l'utiliser) est un des problèmes les plus importants.

Le modèle du système présenté pourrait être le "Worm Community System" [SCHATZ 1992] qui tente de résoudre les problèmes présentés ci-dessus (lien formel/informel, sources de données externes et politique de mise à jour de la base). La différence essentielle tient à la gestion de ces mises à jour qui est réalisée manuellement par un éditeur et non de manière collaborative. À notre connaissance, l'évaluation par les pairs [PETERS 1995] n'a jamais été utilisée pour la construction de bases de connaissance.

L'un des systèmes les plus proches des principes de CO4 est le projet SHADE [GRUBER& 1992] qui aide à la conception collaborative d'un artefact. Un tel système doit assurer la consistance et l'agrément des intervenants. La principale différence réside dans la différenciation des rôles de intervenants au sein de SHADE qui s'appuie sur une théorie de la conception ainsi que sur de nombreux outils préexistants (dans la philosophie du partage de connaissance).

Dans le sillage de SHADE, ONTOLINGUA SERVER [FARQUHAR& 1997] est un système de construction d'« ontologies » accessible par le Web. Il offre de nombreuses similarités avec CO4 (construction coopérative de mémoire consensuelle, représentation par objets dans le contexte hypermédia du Web). ONTOLINGUA SERVER est cependant déployé à une plus vaste échelle. La différence saillante entre les deux systèmes est le contrôle de la collaboration qui est quasi-inexistante dans ONTOLINGUA SERVER (il s'agit principalement de contrôle d'accès) alors qu'elle est un point fort de CO4. WEBONTO [DOMINGUE 1998] est assez proche d'ONTOLINGUA SERVER.

La méthode MKSM dans ses derniers développements [BESSE& 1999] cherche à modéliser finement le processus d'élaboration de « livre de connaissance » et s'appuie pour cela sur des processus d'évaluation par les pairs. Le système ICM [FRUCHTER& 1993] est au contraire un système implémentant un cycle de confrontation de la connaissance très précis : "proposer-interpréter-critiquer-expliquer". Il est dédié à la conception architecturale et fait collaborer les intervenants de différents métiers par le biais des plans. ICM est capable de raisonner sur les dessins (sous différents points de vue par exemple) et d'élaborer et de visualiser des critiques lorsqu'ils entrent en conflit. Par rapport à CO4, ce système produit lui-même les critiques parce qu'il est lié à une représentation formelle maîtrisée par le système et ne laisse que peu de place à la connaissance informelle.

2.5.3 Discussion

Le sujet principal de discussion à propos du protocole de CO4 est l'aspect consensuel (par rapport à majoritaire ou fondé sur des critères de décision économique). Les justifications de ce choix ont été considérées du point de vue des

principes (il doit mener à la discussion, §2.1) et du point de vue technique (il offre des propriétés formelles satisfaisantes, §2.4.4). Il faut rappeler que CO4 offre toujours la possibilité de développer des bases de connaissances alternatives et n'est donc pas un uniformisateur universel, mais ces bases ne pourront évidemment pas être qualifiées de consensuelles. Le point le plus délicat du consensus tel que définit ici, reste donc celui de son acceptabilité sociale.

L'exigence de consistance est aussi un point délicat. Cependant, la disponibilité sur le Web de gigantesque quantité d'information non consensuelle et non consistante a mis en évidence le besoin de contrôle de l'information mis à la disposition des collaborateurs et les approches telles que celle de CO4 sont une réponse à ce type de problèmes. Dans la perspective plus vaste du « web sémantique » [BERNERS-LEE 1998], un tel système pourrait être mis en œuvre

Le protocole de CO4 a sans doute d'autres limitations liées à la restriction des communications possible. Mais ces limitations sont directement héritées du protocole d'évaluation par les pairs qui n'empêche pas les scientifiques de soumettre, évaluer et lire les revues scientifiques. Il est sans doute souhaitable d'adopter une attitude plus souple et d'utiliser dans de tels systèmes plusieurs niveaux de contrôle : libre, modéré, coordonné et évalué [SCHATZ 1992, HARNAD 1998].

La difficulté principale avec CO4 provient du fait qu'il doit être étroitement lié aux besoins de ses utilisateurs qui sinon n'en tireront pas parti. Il sera donc sans doute nécessaire, non pas d'utiliser CO4 tel quel mais de l'adapter à des types de besoins spécifiques (TROEPS est déjà un système très adaptable). À cet égard, le prototype développé est surtout une preuve de la viabilité technique de ses principes.

Bibliographie

- [ALTHOFF& 1998] ALTHOFF K.-D. , BOMARIUS F., TAUTZ C., Using case-based reasoning technology to build learning software organizations, Actes ECAI workshop on « building, maintaining and using organizational memories », Brighton (UK), pp1-20, 1998
- [ASSEMBLÉE NATIONALE 1959] Assemblée nationale, Règlement de l'assemblée nationale, Petite loi 29, Assemblée nationale, Paris (FR), 21 juillet 1959 (tel qu'en vigueur depuis le 25 mars 1998) <http://www.assemblee-nationale.fr/5/5ab.htm>
- [BARTHES 1996] BARTHES J.-P., Mémoire et capitalisation des connaissances dans l'entreprise: problèmes, enjeux, contexte, <http://iiaa.utc.fr>, 1996
- [BERNERS-LEE 1998] BERNERS-LEE T., Semantic web roadmap, <http://www.w3.org/DesignIssues/Semantic.html>
- [BESSE& 1999] BESSE A., ERMINE J.-L., ROSENTHAL-SABROUX C., Modelling organisation, practices and procedures for knowledge book design, Actes 2nd conference on « practical applications of knowledge management », London (UK), 1999
- [BUCKINGHAM SHUM 1997] BUCKINGHAM SHUM S., Negotiating the construction and reconstruction of organisational memories, *Journal of universal computer science* 3(8):899-928, 1997 (<http://www.jucs.springer.com>)
- [CERBAH& 2000] CERBAH F., EUZENAT J., Integrating textual knowledge and formal knowledge for improving traceability, *Lecture notes in computer science 1937* (Olivier Corby, Rose Dieng (éds.), Actes 12th international conference on knowledge engineering and knowledge management (EKAW), Juan-les-Pins (FR), pp296-303, 2000
- [CRAMPÉ& 1998] CRAMPÉ I., EUZENAT J., Object knowledge base revision, Actes 13th ECAI, Brighton (UK), pp 3 - 7 , 1998 <ftp://ftp.inrialpes.fr/pub/sherpa/publications/crampe98a.ps.gz>
- [DIENG& 2000] DIENG R., CORBY O., GIBOIN A., GOLEBIOWSKA J., MATTA N., RIBIÈRE M., Méthodes et outils pour la gestion des connaissances, Dunod, Paris (FR), 2000
- [DOMINGUE 1998] DOMINGUE J., WebOnto 1.0 user guide, 1999 <http://kmi.open.ac.uk/projects/webonto/user-guide-html/>

[ERMINE& 1996] ERMINE J.-L., CHAILLOT M., BIGEON P., CHARRETON B., MALAVIELLE D., MKSM : méthode pour la gestion des connaissances, *Ingénierie des systèmes d'information* 4(4):541-575, 1996 <http://www.mcxapc.org/ateliers/8/mksm.htm>

[EUZENAT 1995] EUZENAT J., Building consensual knowledge bases: context and architecture, dans MARS N. (éd.), *Towards very large knowledge bases*, IOS press, Amsterdam (NL), pp 143 - 155, 1995 <ftp://ftp.inrialpes.fr/pub/sherpa/publications/euzenat95a.ps.gz>

[EUZENAT 1996] EUZENAT J., Corporate memory through cooperative creation of knowledge bases and hyper-documents, Actes 10th knowledge acquisition workshop, Banff (CA), pp(36)1-18, 1996 <http://www.inrialpes.fr/sherpa/papers/euzenat96b.html> <ftp://ftp.inrialpes.fr/pub/sherpa/publications/euzenat96b.ps.gz>

[EUZENAT 1997] EUZENAT J., A protocol for building consensual and consistent repositories, Rapport de recherche 3260, INRIA Rhône-Alpes, Grenoble (FR), septembre 1997, 46p. <ftp://ftp.inrialpes.fr/pub/sherpa/rapports/rr-inria-3260.ps.gz>

[EUZENAT 1998] EUZENAT J., Représentation de connaissance par objets, dans Roland Ducournau, Jérôme Euzenat, Gérard Masini, Amedeo Napoli (éds.), *Langages et modèles à objets : état des recherches et perspectives*, INRIA, Rocquencourt (FR), pp293-319, 1998 <http://exmo.inrialpes.fr/papers/lmobook/>

[EUZENAT 1999] EUZENAT J., La représentation de connaissance est-elle soluble dans le Web ?, *Document numérique* 3(3-4):151-167, 1999

[EUZENAT& 1997] EUZENAT J., CHEMLA C., JACQ B., A knowledge base for *D. melanogaster* gene interactions involved in pattern formation, Actes 5th international conference on intelligent systems for molecular biology, Halkidiki (GR), pp108-119, 1997 <ftp://ftp.inrialpes.fr/pub/sherpa/publications/euzenat97b.ps.gz>

[FARQUHAR& 1997] FARQUHAR A., FIKES R., RICE J., The Ontolingua server: a tool for collaborative ontology construction, *International journal of human-computer studies* 46(6):707-727, 1997

[FISCHER& 1985] FISCHER M., LYNCH N., PATERSON M., Impossibility of distributed consensus with one faulty process, *Journal of the ACM* 32(2):374-382, 1985

[FRUCHTER& 1993] FRUCHTER R., CLAYTON M., KRAWINKLER H., KUNZ J., TEICHOLZ P. Interdisciplinary communication medium for collaborative conceptual building design, Actes 2nd conference on the application of artificial intelligence techniques to civil engineering, Edinburgh (UK), pp 7 - 16, 1993 <ftp://cdr.stanford.edu/pub/CDR/Publications/Reports/ICM.ps>

[GAINES& 1995] GAINES B., SHAW M. WebMap: Concept Mapping on the Web, Actes 4th WWW conference, Boston (MA US), 1995 <http://ksi.cpsc.ucalgary.ca/articles/www/www4wm/>

[GRUBER& 1992] GRUBER T., TENENBAUM J., and WEBER J. Toward a knowledge medium for collaborative product development, dans Gero, J. (éd.). Actes 2nd. international conference on artificial intelligence in design, Pittsburg (PA US), pp413-432, 1992

[HARNAD 1998] HARNAD S., The invisible hand of peer review, *Nature web matters*, 1998 <http://helix.nature.com/webmatters/invisible/invisible.html>

- [HUHNS& 1993] HUHNS M., JACOBS N., KSIEZYK T., SHEN M.-W., SINGH M., CANNATA P. Integrating enterprise information models in CARNOT, Actes 1st international conference on intelligent and cooperative information systems, Rotterdam (NL), pp32-42, 1993
- [KUHN 1959] KUHN T., The essential tension: tradition and innovation in scientific research, dans TAYLOR C. (éd.), Actes 3rd University of Utah research conference on the identification of creative scientific talent, Salt Lake City (UT US), pp162-174 (trad. fr. JACOB P., La tension essentielle : tradition et innovation dans la recherche scientifique, dans JACOB P. (éd.), De Vienne à Cambridge : l'héritage du positivisme de 1950 à nos jours, Gallimard, Paris (FR), 1980), 1959
- [LECHNER& 1999] LECHNER U., SCHMID B., Logic for media : the computational media metaphor, Actes 32th international conference on system sciences (HICSS), Honolulu (HA US), 1999 à paraître
- [MARIÑO& 1990] MARIÑO O., RECHENMANN F., UVIETTA P., Multiple perspectives and classification mechanism in object-oriented representation, Actes 9th ECAI, Stockholm (SE), pp425-430, 1990
- [O'LEARY 1998] O'LEARY D., Enterprise knowledge management, *IEEE Computer* 31(3):54-61, 1998
- [PETERS 1995] PETERS J., The hundred years war started today: an exploration of electronic peer review, 1995 <http://www.mcb.co.uk/literati/articles/hundred.htm>
- [REES& 1995] REES O., EDWARDS N., MADSEN M., BEASLEY M., MCCLENAGHAN A. A Web of Distributed Objects, Actes 4th WWW conference, Boston (MA US), pp75-87, 1995 <http://www.ansa.co.uk/ANSA/ISF/wdistobj/Overview.html>
- [SCHATZ 1992] SCHATZ B., Building electronic community system, *Journal of management information systems* 8(3):87-107, 1992
- [SHERPA 1995] Projet SHERPA, CO4 protocol 1.0 reference manual, Rapport interne, INRIA Rhône-Alpes, Grenoble (FR), juin 1998, 41p. <http://co4.inrialpes.fr/docs/co4-manual.html> <ftp://ftp.inrialpes.fr/pub/sherpa/rapports/co4-manual.ps.gz>
- [SIMON 1997] SIMON G., Modèles et méthodes pour la conception des mémoires d'entreprises, Thèse d'informatique, université Henri Poincaré, Nancy (FR), 1997
- [VAN HEIJST& 1997] VAN HEIJST G., VAN DER SPEK R., KRUIZINGA E., Corporate memories as a tool for knowledge management, *Expert systems with applications*, 13(1):41-54, 1997
- [WIIG 1997] WIIG K., Roles of knowledge-based systems in support of knowledge management, dans Liebowitz J., Wilcox L. (éds.), Knowledge management and its integrative elements, CRC press, Boca Raton (FL US), pp69-87, 1997