

Ontologie et terminologie : le modèle *OK*

Les ontologies, comprises ici comme des vocabulaires communs sur des termes et leurs significations, constituent un axe de recherche primordial pour les sociétés de l'information. Dans ce cadre, elles doivent répondre à un double critère de consensus et de cohérence, condition *sine qua non* d'un modèle computationnel réellement exploitable. Cet article décrit un modèle d'ontologies par différenciation spécifique, héritier direct des arbres de Porphyre, et le langage de description associé ainsi que sa mise en œuvre à travers un environnement logiciel.

Termes-clés : terminologie ; ontologie ; différence spécifique ; langage de description ; représentation.

1 Problématique

Cette introduction a pour but, non pas tant de montrer l'intérêt des ontologies pour la représentation des signifiés, que de présenter les conditions qui ont présidé à une telle approche. Si l'on considère que les usages qu'une communauté fait de ses signes en déterminent le sens, il est important d'en expliciter les cadres d'utilisation. Ces cadres justifient les choix qui seront effectués. Ainsi, la notion d'inter-opérabilité, pris au sens de communication d'agents logiciels hétérogènes, qui n'a de sens que pour les sociétés de l'information, plaide ici pour une modélisation formelle des significations et de leurs relations.

1.1 Cadre général

Notre problématique se situe dans le cadre des sociétés de l'information, où un ensemble d'acteurs, qu'ils représentent des personnes, des organisations, des logiciels ou des machines, sont amenés à communiquer par l'intermédiaire de moyens informatiques.

Le paradigme *agent* permet la communication d'acteurs hétérogènes par leur encapsulation sous une forme normalisée. Les systèmes multi-agents distribués sur des réseaux informatiques communiquant par envoi de messages, partageant et échangeant des connaissances et non plus seulement des données, constituent un modèle privilégié des

sociétés de l'information. L'ingénierie simultanée basée sur une approche multi-agents (agent-based concurrent engineering), où chaque agent coopère et collabore à la conception et la fabrication d'un produit, qu'il soit logiciel ou manufacturé, est une bonne illustration de cette approche (Roche 1998). Citons à titre d'exemples les projets Shade (*SHARED Dependency Engineering*) (McGuire 1993) et PACT (Palo Alto Collaborative Testbed) (Cutkosky 1993) pour l'ingénierie simultanée ; ainsi que les projets Tove (*Toronto Virtual Enterprise*) (Fox 1992) et *Enterprise Project* (Stader 1996) pour la modélisation et l'intégration d'entreprise.

1.2 Les problèmes posés : communication, partage et échange de connaissances

Une société de l'information n'existe que s'il y a communication, partage et échange de connaissances entre ses acteurs. Elle peut faire sienne le leitmotiv de l'ingénierie collaborante : «*Product development is a knowledge and communication intensive process*» (Gruber 1992).

Cela nécessite l'emploi d'un langage de communication compréhensible pour tous les agents. L'utilisation d'un même langage de communication, dénommé *ACL* pour *Agent Communication Language*, permet de résoudre le problème syntaxique. Ces langages, tels que *KQML* et ses dérivés (*Knowledge Query Manipulation Language* (Labrou 1997)) sont basés sur les

actes de langages au sens des projets Darpa.

Le problème sémantique est repoussé à la charge des agents. Seul a été considéré le problème de la sémantique des termes échangés lors des communications: une communication ne pourra être prise en compte que si l'agent receveur en «comprend» les termes. C'est pourquoi la structure des phrases d'un ACL permet de préciser l'ontologie utilisée par l'émetteur d'un message.

1.3 Objectifs et limitations

Les premières réalisations d'architecture multi-agents ont mis en évidence l'importance du problème de la sémantique des termes échangés: il ne peut y avoir communication, et *a fortiori* collaboration et coopération, sans compréhension des mots qui composent le message. Les ontologies, comprises ici comme des vocabulaires communs sur des termes et leurs significations (Gruber 1992), constituent donc un axe de recherche primordial. Notons qu'en toute rigueur une ontologie n'est pas une terminologie, mais une représentation et une structuration particulières de connaissances conceptuelles. Cet abus de langage, en usage dans cette problématique et que l'on retrouvera dans cet article, peut s'expliquer par une présence des connaissances conceptuelles et une acceptation de l'arbitraire du signe dans le domaine technique.

Notre travail porte sur la représentation de la signification des termes dénotant des connaissances conceptuelles. Nous nous limitons aux domaines techniques où le lexique est propre à un groupe social défini par une activité spécifique et où l'élaboration du sens d'un mot s'appuie sur l'élaboration d'une idée. S'écartant des problèmes que posent la langue naturelle, en espérant de pas

tomber dans ceux d'une novlangue, nous avons résolument adopté une approche aristotélicienne de la définition de la signification des termes qui nous permet de garantir une cohérence forte de nos ontologies, condition *sine qua non* d'un modèle computationnel réellement exploitable. Le modèle *OK* obtenu est un héritier direct des arbres de Porphyre (Porphyre, Eco 1988, Rastier 1987).

Ce travail a abouti à une première version d'un environnement logiciel, la *OK Station*, pour *Ontological Knowledge Station*, dédié à l'acquisition, la représentation et l'exploitation de connaissances ontologiques.

2 Le Modèle *OK*

S'il existe d'ores et déjà plusieurs ontologies tant générales que spécialisées, certaines parfois considérables en nombre de termes définis: *Cyc*, *Mikrokosmos* (Mahesh), *Generalized Upper Model*, *Sowa's ontology*, *Tove* (Fox 1992), *Enterprise Ontology* (Stader 1996)... il est cependant difficile de trouver des informations sur leur réutilisabilité et leur compatibilité. Alors que les ontologies ont une visée normative, il est surprenant d'en constater les divergences conceptuelles.

Ainsi, dans le cadre des ontologies d'entreprise, comment concilier les définitions de Tove et celles d'*Enterprise Ontology*? Prenons pour exemple le concept d'activité. *Enterprise Ontology* définit une activité comme étant décomposable en sous activités, réalisée par un exécutant et nécessitant des ressources. Elle hérite de la classe 'Activity-Or-Spec' définie en Ontolingua par la fonction: (Define-Class Activity-Or-Spec (?X) «The union of Activity and Activity-Spec»

```
:Iff-Def (And (Eo-Entity?X) (Or
(Activity?X) (Activity-Spec?X)))
:Axiom-Def (Partition Activity-Or-Spec
(Setof Activity Activity-Spec));
alors que pour Tove une activité est l'opération élémentaire de changement d'état. Elle correspond à un graphe (activity cluster) liant un état initial, dans lequel doit se trouver le système pour que l'activité soit applicable, à un état final. Les activités peuvent être structurées pour définir des activités plus complexes. Ainsi, un plan d'action sera défini par l'instruction:
```

```
(define-class plan_action (?a):def
(forall (?alpha?f?s)
(=> (holds (agent_constraint
?alpha (fluent_goal?f))?)s)
(forall (?ap?s1?s2)
(=> (and (subaction?ap?a)
(leq?s1?s2) (Do?ap?s1?s2)
(intended?s2))
(holds?f?s2))))))
```

De même, que peut-on attendre des définitions de Mikrokosmos, quand la présentation de ce projet débute par ces avertissements: «*In this ontology, you should not expect to find: any kind of guarantees, warranties, or liability for correctness or precision, formally clean or theoretically «pure» concepts, complete consistency; guaranteed absence of contradictions; etc.*».

Les principes épistémologiques (au sens de la théorie de la connaissance) de certaines ontologies, de par leur imprécision, peuvent expliquer ces problèmes. Si le calcul des prédicats offre un formalisme rigoureux pour la définition du sens d'un terme, il ne permet pas néanmoins de différencier entre concept et ensemble si l'on considère que le premier porte sur l'essence des objets qu'il subsume et le second sur leur état (un ensemble regroupe les objets, éventuellement de nature différente, vérifiant une propriété logique portant sur les valeurs de leurs attributs définissant leur état). Tove illustre ce type de problème. De

même, la confusion entre substance et qualité ou quantité, dans le cas de *Mikrokosmos*, peut être source de problèmes : dans une telle ontologie le mercure serait-il à la fois un métal et un liquide ou un métal dont l'état, sous certaines conditions, est liquide ? Le problème de la langue naturelle, miroir incontournable mais déformant de la réalité, devra être pris en compte lors de l'acquisition des connaissances ontologiques.

2.1 Principes d'OK

Afin d'éviter de tels écueils, le modèle *OK* repose sur des principes épistémologiques forts, voire contraignants, qui, s'ils ne permettent pas d'aborder tous les problèmes de la sémantique lexicale, permettent d'obtenir des définitions consensuelles et cohérentes souvent suffisantes dans le cadre de domaines techniques.

Notre objectif est la définition de la signification de termes dénotant des connaissances conceptuelles (concepts et ensembles), c'est-à-dire portant sur une pluralité de choses, et ce à partir de quoi elles sont définies (différences et attributs). L'ensemble de ces termes est structuré en quatre vocabulaires : celui des concepts, des ensembles, des différences et des attributs. L'association d'une signification à chacun de ces termes constitue une ontologie *OK*.

Le modèle *OK* s'appuie sur le classique triangle sémiotique : [«signifiant» – <signifié> – référent] où le sens d'un «mot» est le <concept> qu'il dénote. Savoir ce que signifie un mot se ramène à connaître l'idée dont il est le signe. Cette approche relève donc davantage d'une théorie des idées que de la linguistique proprement dite. Pour la suite de l'article, la notation «m» désignera le terme et <m> sa signification.

• Une distribution du sens

L'approche adoptée par *OK* n'impose pas une ontologie commune, acceptée et partagée par tous les agents. Le propre de l'ingénierie simultanée est d'avoir pris conscience de cette distribution du sens en autant d'ontologies locales aux agents que nécessaire. Néanmoins l'existence d'une ontologie minimale partagée par tous, l'ontologie invariante, est indispensable à la communication entre agents, tout comme l'est la compatibilité des ontologies locales avec l'ontologie commune.

• Une définition de la définition

Le modèle *OK* repose sur une définition précise des signifiés conceptuels (concepts) : le signifié d'un terme désignant un concept est défini à partir d'un autre signifié conceptuel (concept) en précisant sa **différence spécifique**.

De cette définition découlent les propriétés du modèle *OK*. Ce modèle repose sur une sémantique référentielle puisque le sens d'un terme est l'idée dont il est le signe, et différentielle dans la mesure où ce sens est défini à partir d'un signifié existant par différence spécifique.

OK postule une classification précise des connaissances mises en jeu pour la définition des signifiés, distinguant d'une part les concepts des ensembles, et d'autre part les différences des attributs. Les notions de concept et de différence sont centrales dans *OK*, c'est pourquoi nous nous limiterons à la présentation de ces deux seules notions.

• Les différences

La différence est l'unité sémique élémentaire à partir de laquelle se construit par différence la sémantique des concepts. Elle traduit une qualité essentielle des objets subsumés par le concept qui ne peut être évaluée (si *mortel* est une différence, *âge* est un attribut). Nous retrouvons ici l'approche des arbres de Porphyre (Porphyre).

De par la définition de la construction du signifié d'un concept, la différence est une unité divisive et constitutive : divisive au sens où elle sépare les concepts en ceux qui la possèdent et ceux qui ne la possèdent pas ; constitutive au sens où elle participe à leur l'essence. Une différence ajoutée à un concept <C> permet donc d'engendrer deux nouveaux concepts <C1> et <C2>, le premier la possédant, le deuxième ne pouvant la posséder. Une telle différence est dite différence spécifique pour le concept <C1>. Une différence divise l'ensemble des <concepts> subsumés en deux sous-ensembles de <concepts>, ceux qui la possèdent et ceux qui ne peuvent la posséder. Il en découle que l'ensemble des signifiés conceptuels se structure sous la forme d'un arbre binaire (arbre de Porphyre). La figure n°1 en est un exemple.

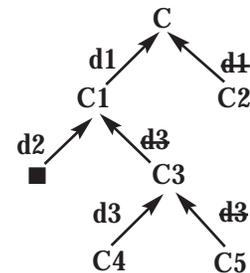


Figure 1

Les différences sont définies par paires de différences opposées (par exemple d1 et d1 (avec un trait horizontal) de la figure n°1) correspondant à des antonymes complémentaires. En tant qu'unité sémique élémentaire, aucun signifié n'est associé à une différence qui ne prend de sens que dans la mesure où elle participe à la définition des <concepts>. C'est pourquoi la détermination de ces sèmes élémentaires doit être consensuelle. Bien que consensuelle, la détermination de ces différences reste arbitraire et subjective. Elle est

fonction d'une application donnée et d'une communauté particulière.

• Les concepts

Le concept s'intéresse à l'essence et à la structure des choses de même nature, concepts ou objets, indépendamment de l'état des objets qu'il subsume. Il se distingue de l'ensemble qui permet de regrouper des objets pouvant être de nature différente mais satisfaisant des propriétés communes pouvant porter sur les valeurs de leurs attributs (état de l'objet). Ainsi, si *Homme* est un concept, *Adolescent* est un ensemble défini en intention par l'expression logique: $\text{Homme}(x) \wedge \text{âge}(x) < 18$.

Par définition les signifiés conceptuels sont structurés sous la forme d'un arbre binaire et héritent des différences spécifiques des <concepts> qui le subsument.

• Les catégories

Il existe des concepts qui par essence n'entretiennent aucune relation quant à leur définition (*êtres animés, usinages, temps...*). Leurs signifiés sont alors structurés en autant d'arbres binaires différents correspondant à autant de catégories OK. Les concepts racines de ces catégories ne sont pas définis par différenciation spécifique, ce sont des concepts primitifs dont le choix détermine une modélisation particulière du monde.

• Propriétés

La définition d'un concept par différenciation spécifique à partir d'un signifié existant basée sur l'utilisation de couples de différences correspondant à des antonymes complémentaires offre de multiples avantages qui ont justifié ce choix:

- Cette approche de la définition, pour rigide qu'elle puisse paraître, a l'avantage d'être admise par tous;
- Elle se concentre sur l'essence des choses sans la confondre avec la notion d'état soumis aux changements;
- Elle évacue le problème de la hiérarchie multiple par définition

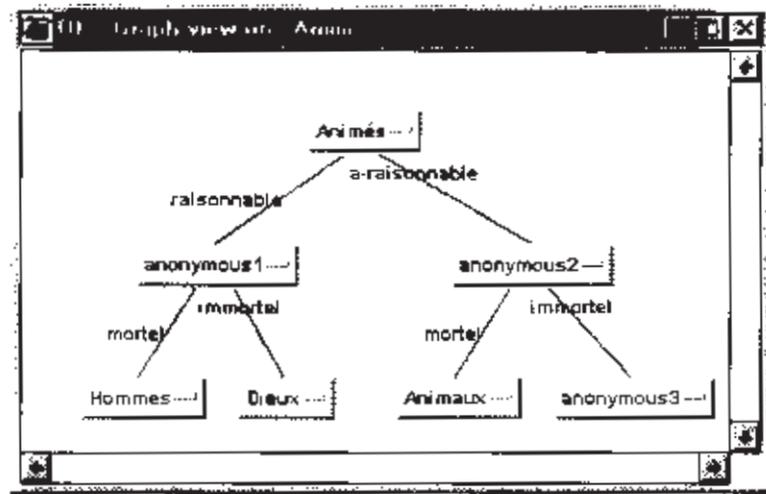


Figure 2

même des concepts. De plus, les différences étant définies par paires correspondant à des antonymes complémentaires, un concept ne peut hériter de deux concepts différents, nécessairement opposés par une paire d'antonymes (sachant qu'un concept OK ne peut appartenir, par définition, qu'à une seule catégorie);

- Elle assure aux modèle OK des propriétés logiques intéressantes, garantes de la cohérence des ontologies. Ces propriétés sont exploitées tant pour la définition ou l'acquisition de nouveaux concepts que pour le classement;
- Elle permet de définir des équivalences entre arbres du fait que seules importent les différences et ce quel que soit leur ordre,
- Elle réduit le problème des définitions consensuelles à la seule détermination des différences. Ce consensus est relativement facile à obtenir dans la mesure où les différences sont des unités de sèmes élémentaires. Il l'est d'autant plus que le domaine est technique et que les différences désignent des faits concrets: «enlèvement de matière», «rotation de la pièce», «traitement thermique»...

2.2 le langage Lok

La gestion des «signifiants» et des «signifiés» se fait à l'aide d'un langage dédié, le langage *Lok* pour *Language for Ontological Knowledge*. Ce langage comporte plus de 150 instructions, avec une syntaxe «à la Lisp» (notation fonctionnelle préfixée parenthésée). Ces instructions se répartissent en deux ensembles.

• Les instructions de définition d'ontologies

Un premier ensemble d'instructions *Lok* permet la définition et la modification des associations «signifiant» – «signifié». Prenons pour exemple l'arbre des concepts de la figure n°2.

Les instructions *Lok* suivantes permettent de définir de nouvelles paires de différences antonymes. Lorsqu'une seule différence est définie, la différence opposée est automatiquement créée à partir de cette différence en la préfixant du «a-» privatif.
 (defineDifference raisonnable')
 returns ('raisonnable a-raisonnable')
 (defineDifference mortel immortel')
 returns ('mortel immortel')

La définition d'un nouveau signifié conceptuel s'effectue à partir d'un signifié existant en précisant la différence spécifique.

```
(defineConceptFrom Animés
(leftConcept ?
(specificDifference raisonnable'))
returns ('anonymous1 anonymous2'))
(defineConceptFrom Animés
raisonnable
(leftConcept Hommes
(specificDifference mortel'))
(rightConcept Dieux
(specificDifference immortel')))
returns ('Homme Dieux')
```

Il est à noter que *Lok* permet de définir des signifiés sans signifiant («?») qui seront dénotés par une désignation, c'est-à-dire par un terme dénotant un concept et une suite de termes dénotant des différences: *Animés raisonnable* désignant le concept anonyme *anonymous1* de la figure n°2.

Toutes les instructions de ce premier ensemble modifient l'ontologie. C'est pourquoi une ontologie *OK* sera définie par une suite d'instructions *Lok* appartenant à ce premier ensemble.

• Les instructions de manipulation d'ontologies

Le deuxième ensemble d'instructions est utilisé pour l'exploitation des ontologies: interrogation et recherche selon différents critères, tant sur les termes que sur leurs significations.

L'évaluation de l'instruction suivante retourne la définition d'un concept sous la forme d'une chaîne de caractères:

```
(prettyDefinitionOf Homme)
returns concept 'Homme is Animé
raisonnable with
specific difference
mortel category
Animé
```

Ces instructions ne modifient pas l'ontologie et ne participent donc pas à sa définition.

2.3 Représentation computationnelle

Une ontologie *OK* est représentée de façon déclarative par un fichier d'instructions *Lok*. Le résultat de la compilation d'un tel fichier est une représentation computationnelle de l'ontologie qui permet de gérer cinq vocabulaires de termes, celui des concepts, des différences, des attributs, des relations et des ensembles, ainsi que leurs significations (représentation et organisation des concepts).

2.4 Lok: langage pivot

Un des buts des ontologies est de permettre le partage et l'échange de connaissances (termes et significations). C'est pourquoi il est

nécessaire de pouvoir fournir des ontologies écrites dans un formalisme plus commun que *Lok*. En fait, chaque acteur peut utiliser son propre formalisme de représentation de connaissances: réseau sémantique, graphe conceptuel, schéma, calcul des prédicats, base de données relationnelles, etc. dont la diversité illustre l'ampleur du problème. Plutôt que de traduire une représentation *Lok* dans chacun de ces formalismes, ce qui impliquerait l'écriture d'autant de traducteurs, nous avons choisi la solution des formats d'interchange qui permet de réécrire une ontologie *OK* dans un formalisme accepté par tous (le terme d'interchange que l'on préférera à celui d'échange, tout peut être échangé, insiste d'avantage sur le caractère commun du formalisme). C'est pourquoi les ontologies *Lok* peuvent être traduites en *Kif* (*Knowledge Interchange Format*) ou en graphes conceptuels. Ces fichiers sont plus conséquents en taille et moins lisibles du fait que ces formalismes sont plus généraux que *Lok* et qu'il est dès lors nécessaire de traduire également la sémantique du modèle *OK*.

2.5 Architecture de la OK Station

La *OK Station* est un environnement logiciel qui permet à

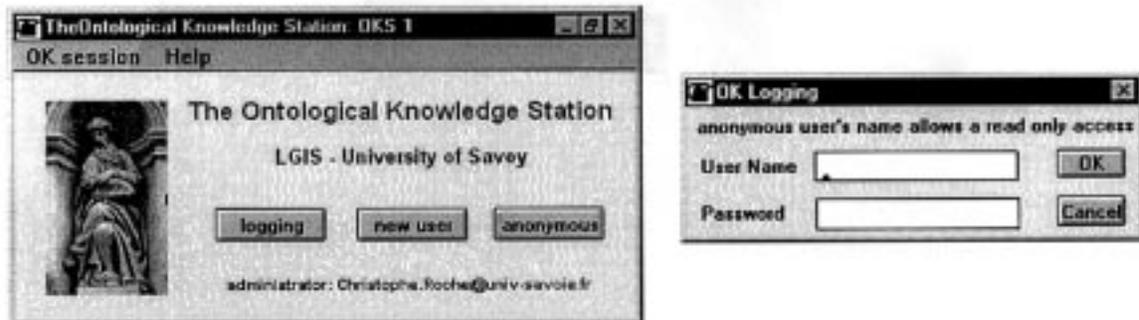


Figure 3

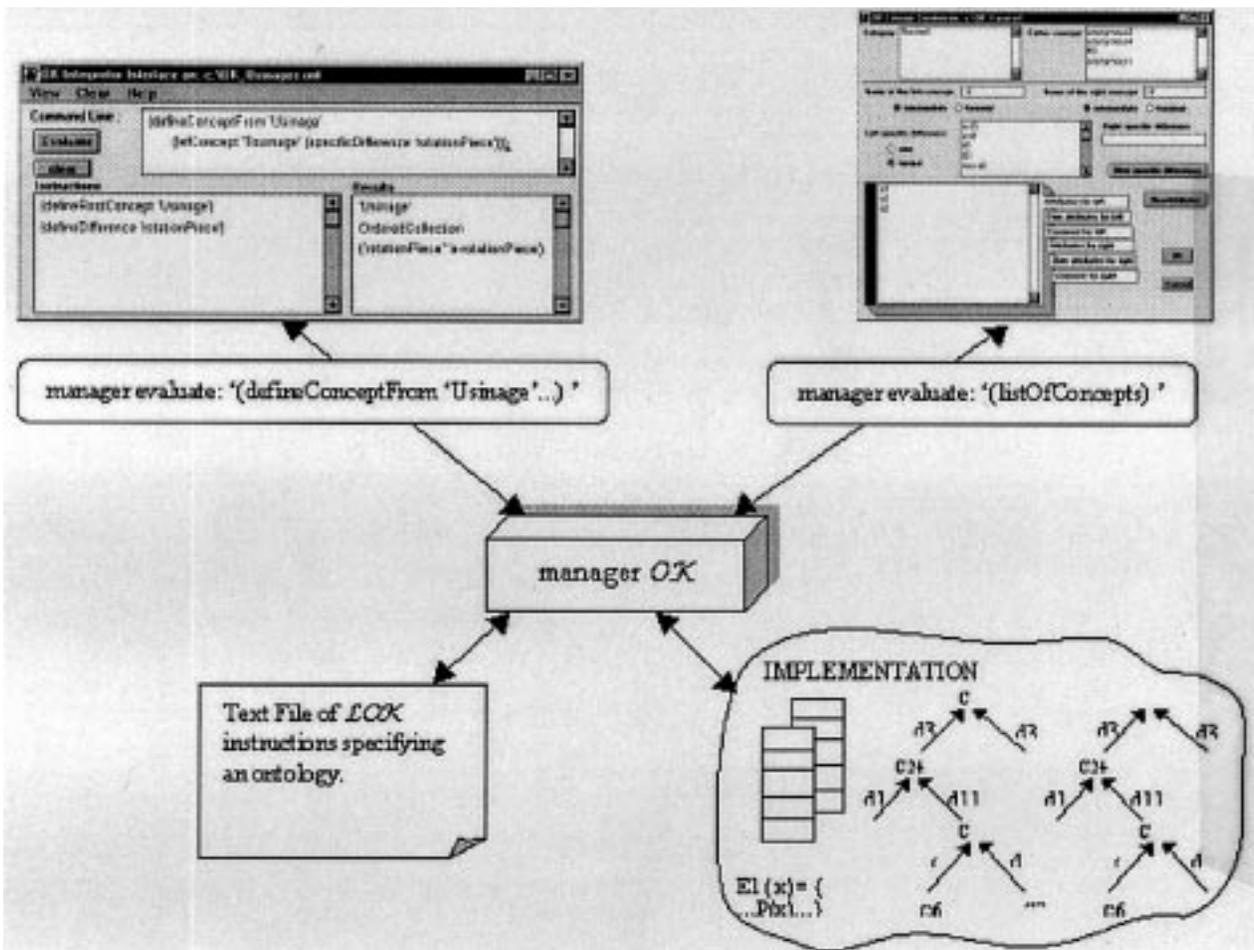


Figure 4

un ensemble d'utilisateurs d'acquérir, de définir, d'exploiter et de gérer des ontologies.

L'architecture de la *OK Station* repose sur trois modules: un module de gestion des utilisateurs, un module de gestion des ontologies comprenant un ensemble d'outils pour la définition et l'exploitation des termes qui les composent; et un module d'acquisition décrit dans le troisième paragraphe.

• Les utilisateurs

La *OK Station* permet à un ensemble d'utilisateurs de gérer (définir et exploiter) leurs ontologies en offrant les sécurités classiques d'un

tel système: gestion des utilisateurs, mot de passe, droits d'accès, gestion des ressources critiques. La figure n°3 présente la bannière d'accueil de la *OK Station* et la fenêtre de connexion des utilisateurs.

• Les «managers»

L'utilisation d'une ontologie définie par un fichier d'instructions *Lok* se fait par l'intermédiaire de sa représentation computationnelle et de son manager associé. La représentation informatique n'étant pas accessible, le manager *OK* est le passage obligé pour l'exploitation des termes et de leurs significations. Ainsi, pour une ontologie donnée, la

création d'un nouveau concept, la recherche de la définition d'un terme, sont autant de requêtes écrites en *LOK* que le manager associé à l'ontologie devra interpréter. La figure n°4 décrit les rapports entre ces différents composants.

• Le «launcher»

La gestion des ontologies d'un utilisateur s'effectue à l'aide d'un *launcher OK* qui lui est associé lors de sa connexion à la *OK Station*. Ce *launcher* lui permet d'une part de gérer les ontologies en tant que fichier: suppression, création, édition, traduction en format *Kif* (*Knowledge Interchange Format*) ou en graphes

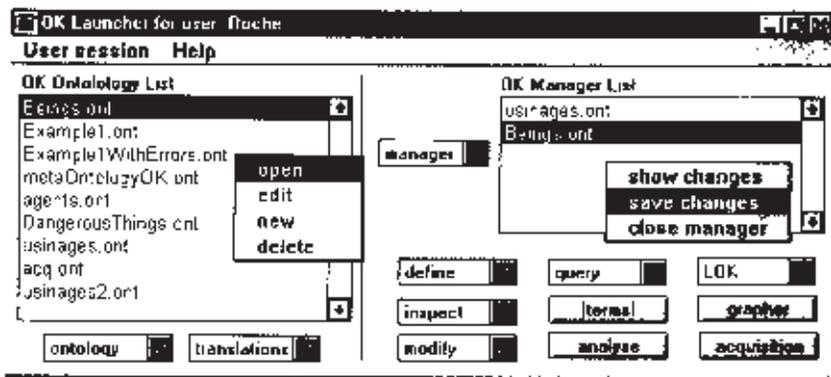


Figure 5

conceptuels; et d'autre part de gérer leurs représentations computationnelles par l'intermédiaire des managers associés aux ontologies lors de leur ouverture: gestion des termes et de leur signification: définition, modification et recherche. La figure n°5 présente l'ensemble de ces fonctionnalités.

• Les outils

Enfin, la *OK Station* offre un ensemble d'outils pour la manipulation des termes et de leur signification: éditeur graphique interactif, interpréteur d'instructions *Lok*, fouineurs de définition, de suppression, d'édition, de recherche... Tous ces outils, interfacés graphiquement, sont accessibles pour un utilisateur donné à partir de son *launcher* et portent sur l'ontologie gérée par le manager sélectionné.

• Applications

La *OK Station* est utilisée dans le cadre d'un projet Eurêka pour la définition des termes utilisés au sein de l'équipe de développement (ontologies agents) et pour la définition d'ontologies d'entreprise (usines, modélisation de l'entreprise).

3 Acquisition

Le chapitre 3 présente des travaux menés en parallèle avec ceux décrits dans les deux chapitres

précédents. Il présente en particulier comment certaines techniques d'acquisition peuvent être particularisées pour le modèle *OK*.

Le travail de modélisation de la connaissance peut être couplé avec l'acquisition de connaissances. L'acquisition de connaissance consiste à extraire et à analyser les connaissances d'un domaine et à les décrire d'une manière à la fois compréhensible et rigoureuse. Ce n'est pas une simple activité de collecte, mais un réel travail de modélisation. Ainsi, avec notre approche ontologique, nous voyons apparaître la nécessité d'adapter les méthodes d'acquisition choisies à notre modèle de représentation: Les concepts fondamentaux de la modélisation doivent transparaître à travers le guide offert par les différentes méthodes d'acquisition.

Le processus d'acquisition de connaissance est un processus incrémental. (Marty 1991). Chaque itération du processus permet d'acquérir de nouvelles connaissances et de raffiner ou d'enrichir la version courante de l'ontologie. L'intégration de plusieurs méthodes d'acquisition permettra de mieux tirer parti des avantages de chaque méthode. Le processus d'acquisition étant incrémental, on peut donc fort bien considérer que chaque cycle est en fait l'application d'une technique

d'acquisition (la plus adéquate à cette étape) afin de raffiner l'ontologie.

Chaque technique est caractérisée par une stratégie de recueil particulière, un mode de représentation des connaissances spécifique et des moyens d'analyse associés pour valider la modélisation obtenue. L'utilisation de l'éventail des techniques proposées permet ainsi de gagner en productivité, en qualité et en précision:

- En productivité car nous incluons des techniques d'extraction qui permettent de structurer et d'analyser les connaissances directement au cours d'une interview structurée; la phase de transcription en texte libre est ainsi supprimée. La productivité est aussi améliorée car les principes de base de ces techniques, issus de la psychologie cognitive, facilitent l'expression des experts.
- En qualité puisque les techniques reposent sur des stratégies systématiques (moins de risque d'oubli, de digression...) et aident à détecter les ambiguïtés, les contradictions ou les manques.
- En précision grâce à la synergie entre techniques: l'ingénieur de la connaissance peut appliquer à chaque étape la technique de modélisation la plus appropriée. La connaissance ainsi acquise pourra être ultérieurement utilisée par d'autres outils d'acquisition.

Chacune de ces activités contribue à l'extraction, à la précision ou à la réorganisation de la connaissance nécessaire pour résoudre un problème dans un domaine donné. On peut donc imaginer autour de notre noyau de modélisation ontologique, un atelier de génie cognitif composé de différents outils supportant des méthodes variées et complémentaires d'acquisition de connaissance. Cette plate-forme devra être ouverte afin de pouvoir intégrer de nouvelles techniques si besoin est. Nous avons déjà identifié au moins les besoins suivants:

• Détection et corrélation des éléments du domaine

Une des activités du cogniticien consiste à analyser les données brutes du domaine (documentations, transcriptions d'interviews) en identifiant et en corrélant les concepts du domaine, les attributs, les relations. Cette activité structure principalement la connaissance textuelle. Elle produit un vocabulaire des termes les plus importants du domaine ainsi que des liens sur leurs parties explicatives. Elle permet également d'annoter les différentes informations et de les classer les unes par rapport aux autres, procurant ainsi une première hiérarchie taxinomique des informations du domaine.

• Hiérarchie taxinomique des concepts: création et raffinement

Les concepts significatifs identifiés dans le domaine doivent être regroupés dans une hiérarchie entre concepts. Les attributs significatifs à prendre en compte pour ces concepts doivent être fournis. Des règles de classification peuvent émerger de cette activité.

Bien souvent, cette classification s'effectuera lors d'une séance d'interview supplémentaire avec l'expert du domaine. Une méthode adéquate pour cette activité est la méthode du tri par cartes (Gamack 1987) qui aide souvent à identifier des groupes de concepts liés entre eux.

La méthode de grille répertoire (Bannister 1977) permet au cogniticien de raffiner la hiérarchie des concepts en construisant une grille contenant d'une part des concepts du domaine et d'autre part des attributs de ces concepts qui ont deux extrêmes.

Nous avons particularisé ces méthodes classiques afin que le cogniticien se rapproche le plus possible de la formalisation en phase d'acquisition et qu'il prenne en compte les éléments majeurs de notre modèle *OK*, comme par exemple raisonner par attribut différenciateur.

En ce qui concerne la détection et la corrélation des éléments du domaine, la première itération peut consister en une interview libre. Comme notre système est ouvert, on pourrait imaginer ici l'intégration de techniques d'extraction statistiques telles que celles utilisées dans *Lexter* (Bourigault 1995). Actuellement, cette première élicitation du vocabulaire du domaine se fait en collaboration avec un expert, qui énonce les principaux termes du domaine et donne une définition pour chacun de ces termes. La définition donnée par l'expert fait apparaître de nouveaux termes, qui sont rajoutés à la liste, et dont l'expert doit également donner une définition. On s'arrête lorsque tous les termes apparaissant dans les définitions ont été eux-mêmes définis. On estime alors que l'on a obtenu une liste de termes suffisamment complète pour passer à l'étape suivante. Bien entendu, on peut revenir à cette étape dans une itération ultérieure, si une étape d'acquisition suivante fait apparaître de nouveaux termes.

Ensuite, on réalise un premier tri par cartes sur ces termes du domaine. L'objet de ce tri par cartes est de catégoriser le domaine (au sens des catégories *OK*). Ici, c'est l'expert qui détermine ces différentes catégories par le choix des tas. Ce tri permet également de simplifier les étapes suivantes de l'acquisition, en fournissant de plus petits groupes de termes.

L'étape suivante de notre processus incrémental consiste à élaborer la hiérarchie taxinomique des concepts. Nous nous rapprochons alors de notre modèle de représentation des connaissances, et les techniques d'acquisition vont devenir directives pour respecter ce modèle. Ainsi, on utilise de nouveau un tri par cartes, dédié à la terminologie *OK*, dont les tas correspondent aux types de termes. On obtient ainsi un tas de concepts,

un tas de différences spécifiques, un tas d'attributs, etc.

Il reste alors à déterminer l'organisation des concepts entre eux, à l'aide des différences spécifiques. On utilise ici la technique des grilles répertoires, particularisée pour *OK*. Dans notre cas, au lieu d'utiliser des attributs pour différencier les concepts, on utilise les différences spécifiques. Celles-ci n'étant par définition pas valuées, on ne remplit pas les cases de la grille avec des notes, comme on le fait habituellement pour les attributs dans la technique de grille répertoire. On remplit plutôt les cases de la grille en déterminant, pour chaque différence et chaque concept, si le concept possède ou non cette différence, s'il peut la posséder, ou si celle-ci n'a pas de sens pour le concept. Lorsque la grille est remplie, on l'analyse et on la complète avec l'aide de l'expert.

Cette étape d'analyse de la grille, habituellement utilisée pour rapprocher les concepts ou les attributs du domaine, sera dans notre cas utilisée pour déterminer si la grille est complète et cohérente. Par exemple, si deux concepts ont les mêmes valeurs pour toutes les différences, cela signifie soit qu'ils sont synonymes, soit qu'il manque un élément dans leur définition. S'il s'agit du deuxième cas, l'expert les différencie en rajoutant à la grille une nouvelle différence spécifique. De même, le système détecte les différences spécifiques susceptibles d'être synonymes ou opposées. Lorsque la grille est complète, on construit la hiérarchie des concepts d'après leurs différences.

Notre processus incrémental peut alors entrer dans une étape de raffinement. Dans certains cas, la génération de hiérarchies de concepts fournit une forêt et non un arbre. Dans ce cas, l'expert peut valider les résultats obtenus, auquel cas il s'agissait de plusieurs catégories différentes. Il peut au contraire

compléter la grille de manière à relier ces hiérarchies de concepts, en ajoutant par exemple une différence spécifique qui permettrait de les opposer.

Conclusion

Une société de l'information n'existe que s'il y a communication, partage et échange de connaissances entre ses acteurs. Or il ne peut y avoir communication, et *a fortiori* collaboration et coopération, sans compréhension des mots qui composent le message. C'est pourquoi les ontologies, comprises ici comme des vocabulaires communs sur des termes et leurs significations, constituent un axe de recherche primordial. Dans ce cadre particulier d'applications, où l'on est confronté à des problèmes qui lui sont propres tels que la notion d'inter-opérabilité, pris au sens de communication d'agents logiciels hétérogènes, les ontologies doivent répondre à un double critère de consensus et de cohérence, condition *sine qua non* d'un modèle computationnel réellement exploitable. Pour cela, elles doivent reposer sur des fondements épistémologiques « clairs » qui ne peuvent puiser à une seule discipline mais tenir compte des enseignements de la linguistique, des sciences cognitives, de la sémantique, de la théorie de la connaissance et des modèles computationnels.

Une approche atomiste de la signification basée sur la notion de différence, unité sémique élémentaire à partir de laquelle se construit la sémantique des termes désignant des concepts, directement issue des arbres de Porphyre, permet de répondre de façon satisfaisante à ces exigences en ce qui concerne les domaines techniques. Le modèle *OK*, pour *Ontological Knowledge*, présenté dans cet article, en est une illustration. La définition d'un concept par

différenciation spécifique basée sur l'utilisation de couples de différences correspondant à des antonymes complémentaires offre de multiples avantages qui ont justifié ce choix. Ainsi, l'accent est mis sur l'essence des choses sans la confondre avec la notion d'état soumis aux changements; le problème de la hiérarchie multiple est évacuée par définition même; le modèle offre des propriétés logiques intéressantes et garantes de la cohérence des ontologies; il est possible de définir des « équivalences » entre arbres des significés du fait que seules importent les différences et ce quel que soit leur ordre; et enfin le problème des définitions consensuelles est réduit à la seule détermination des différences.

Une ontologie trouve sa justification dans la pleine exploitation des relations entre les signifiés. Pour cela nous avons défini un langage déclaratif de représentation ontologique, le langage *LOK*, et un modèle computationnel associé. Le résultat en est un environnement logiciel, la *OK Station*, qui permet à un ensemble d'utilisateurs d'acquérir, de définir, d'exploiter et de gérer des connaissances ontologiques.

De nombreux problèmes restent en suspens tels que la distribution du sens à travers des ontologies locales, la réutilisabilité d'ontologies existantes et l'évolution des significations. Ici aussi, nous pensons qu'une approche analytique, même si elle est réductrice par certains aspects, peut apporter des solutions intéressantes dans le cadre des sociétés de l'information.

*Christophe Roche,
Jean-Charles Marty,
Stéphanie Lacroix,
LGIS – Université de Savoie.*

Bibliographie

Bourigault (D.), Lépine (P.), 1994: «Extraction et structuration automatiques de terminologie pour l'aide à l'acquisition des connaissances à partir de textes », dans *Actes du 9^e Congrès AFCET RFIA'94*, janvier 1994.

Bachimont (B.), 1996: *Herméneutique matérielle et artefacture: des machines qui pensent aux machines qui donnent à penser*, Doctorat de l'École Polytechnique, 1996

La banque des mots, 1995: *Terminologie et intelligence artificielle*, numéro spécial 7-1995, revue de terminologie française.

Cutkosky, Engelmores, Fikes, Gruber, Genesereth, Mark, Tenenbaum et Weber, 1993: «PACT: An experiment in integrating concurrent engineering systems», dans *IEEE Computer*, vol. 26, n°1, January 1993.

Eco (U.), 1988: *Sémiotique et philosophie du langage*, PUF, Paris, 1988

Fox (M.S.), 1992: «The TOVE Project: Towards a Common Sense Model of the Enterprise », dans *Enterprise Integration*, C. Petrie (Ed.), 1992, Cambridge MA: MIT Press.

Gamack (J.G.), 1987: «Different Techniques and Different Aspects of Declarative Knowledge», dans Kidd (A.L.) editor, *Knowledge Acquisition for Expert Systems: A Practical handbook*, Plenus Press, New York, 1987.

Fransella, D. Bannister, 1977: *A Manual for Repertory Grid Technique*, Academic Press, 1977

Gruber (T. R.), J.M. Tenenbaum and J.C. Weber, 1992: «Towards a knowledge Medium for Collaborative Product Development», dans *Proceedings of the Second International Conference on Artificial Intelligence in Design*, (Pittsburgh, Ill., USA, Jun. 22-25, 1992), Kluwer Academic Publishers.

Gruber (T.R.): «Toward Principles for the Design of Ontologies Used for Knowledge Sharing», dans *Formal Ontology in Conceptual Analysis and Knowledge Representation*, edited by Nicola Guarino and Roberto Poli, Academic Publishers.

- McGuire, Kuokka, Weber, Tenenbaum, Gruber et Olsen, 1993: «SHADE: Technology for Knowledge-Based Collaborative Engineering», dans *Concurrent Engineering: Research & Applications*, Volume 1, n° 3, September 1993
- Labrou (Y.), Finin (T.), 1997: *A proposal for a new KQML Specification*, Internal Report TR CS-97-03, Computer Science and Electrical Engineering Department (CSEE), University of Maryland Baltimore County (UMBC).
- Mahesh (K.): *Mikrokosmos*, <http://crl.nmsu.edu/Research/Projects/mikro/htmls/ontology-html/onto.index>
- Marty (J.C), Ramparany (F), Doize (M.S), Jullien (C.), 1991: «ACKnowledge: An Intelligent Workbench for the Knowledge Engineer», dans *Actes The World Congress on Expert Systems*, December 1991.
- Porphyre: *Isagoge*, traduction J. Tricot, Vrin 1984
- Rey (A.), 1992: *La Terminologie, noms et notions*, Paris, PUF.
- Rastier (F.), 1987: *Sémantique interprétative*, Paris, PUF.
- Rastier (F.), 1991: *Sémantique et recherches cognitives*, Paris, PUF.
- Roche (C.), Dumond (Y.), 1998: «From Concurrent Engineering to Collaborative Engineering: An Agent-Oriented Approach», dans *ECEC'98, 5th European Concurrent Engineering Conference (SCS: Society for Computer Simulation)*, Erlangen-Nurember, Germany, April 26-29 1998
- Roche (C.), 1999: «Ontology: A Key Point for Concurrent Engineering», dans *Actes CAPE'99: the 15th International Conference on Computer-Aided Production Engineering*, Durham, UK, April 19-21 1999
- Stader (J.), 1996: «Results of the Enterprise Project», dans *Actes Expert Systems 96, the 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems*; Cambridge, UK; December 1996