

Navigation hypertexte dans une ontologie multi-points de vue

Gilles Falquet, Claire-Lise Mottaz Jiang

Centre Universitaire d'Informatique, Université de Genève, Rue du Général-Dufour 24, 1211 Genève 4, Suisse
{Gilles.Falquet, Claire-Lise.Mottaz}@cui.unige.ch

RESUME : Une ontologie multi-points de vue est une ontologie dans laquelle chaque concept peut avoir plusieurs définitions différentes, chacune d'entre elles représentant un point de vue différent (selon le point de vue un concept peut également être situé à plusieurs emplacements différents dans la hiérarchie). Si beaucoup de travaux ont porté sur l'utilisation d'ontologies dans diverses applications, relativement peu a été écrit sur la construction manuelle d'ontologies et sur leur utilisation comme outil de référence (à la manière d'un dictionnaire) par des utilisateurs humains. Dans cet article, nous étudierons la création d'interfaces hypertextes qui permettent de naviguer dans une ontologie. La navigation est un moyen intéressant pour se familiariser avec un domaine, à la fois pour les utilisateurs finaux et pour les personnes participant à la construction ou à la mise à jour de l'ontologie elle-même. Nous chercherons entre autres à créer de nouveaux types de liens entre les concepts (c'est-à-dire des liens autres que les liens générique/spécifique ou que les rôles de la logique descriptive utilisée pour spécifier l'ontologie), par exemple des liens "calculés". On peut par exemple calculer des distances entre concepts; distances qui seront ensuite utilisées pour créer des liens entre concepts proches.

MOTS CLES : ingénierie des connaissances, ontologie, hypertexte

1. Introduction

Les ontologies constituent l'un des ingrédients essentiels de la gestion des connaissances. Elles sont utilisées dans un grand nombre d'applications informatiques pour faciliter la diffusion, le partage, la réutilisation et la conservation des connaissances. Les ontologies sont parfois aussi utilisées dans des systèmes de recherche d'information ou pour la fabrication de livres électroniques.

Il y a une grande similarité entre ontologies et terminologies: toutes deux présentent un panorama d'un domaine de connaissance. Cette représentation du domaine est constituée d'une liste des concepts pertinents, des termes qui sont utilisés pour les désigner, de leur définition et de liens (explicites ou implicites) qui permettent de les positionner les uns par rapport aux autres. La principale différence entre ontologie et terminologie réside dans le fait que dans une terminologie, les définitions des concepts sont rédigées en langue naturelle, tandis que dans une ontologie, les définitions sont le plus souvent formalisées, de manière à permettre un certain nombre de traitements automatiques. Il faut cependant noter que les formalismes utilisés pour spécifier les ontologies (par exemple les langages à base de frames, Cyc, KIF, OIL...) ne sont pas faciles à maîtriser pour des non-spécialistes de l'ingénierie des connaissances.

Il existe désormais de nombreux logiciels et systèmes permettant de créer et de manipuler des ontologies, parmi lesquels on peut citer : OntoEdit [ONT], OntoTerm [MOR 00], OilEd [OIL], WebOnto, Tadzebao [DOM

98], Ontosaurus [SWA 96] et Ontolingua [FAR 96]. Cependant, bien que ces logiciels proposent des interfaces graphiques pour faciliter le travail de l'utilisateur, ce dernier doit malgré tout bien connaître le formalisme de représentation des connaissances pour pouvoir manipuler efficacement l'ontologie. En effet, les interfaces de ces logiciels ne masquent généralement pas la représentation interne et n'offrent souvent qu'une vue unique des informations, par exemple sous forme d'un graphe ou d'un arbre de concepts.

La définition de Studer et al. [STU 98] « An ontology is a formal, explicit specification of a shared conceptualization » implique qu'une ontologie est le résultat d'un consensus entre les personnes ayant participé à sa construction. La plupart des modèles de représentation des ontologies font écho à cette définition : chaque concept ne doit avoir qu'une seule définition (dans les systèmes CO4 [EUZ 96] et APECKS [TEN 98], il est possible d'avoir des définitions différentes, mais elles sont dispersées dans des ontologies séparées, ce qui rend plus difficile leur consultation simultanée). Ces modèles sont donc plus prévus pour stocker le *résultat* de la construction de l'ontologie plutôt que pour supporter le *processus* de construction.

Le but de notre recherche est double. Nous essayons tout d'abord de fournir un formalisme de représentation d'ontologies et des outils de manipulation qui soient adaptés aux besoins de non-spécialistes de l'ingénierie des connaissances. Pour la construction des outils, nous utilisons le paradigme de l'hypertexte, qui, de par sa simplicité, s'est révélé être un des facteurs-clés de succès

du Web. Nous espérons ainsi pouvoir donner la possibilité de construire des ontologies formalisées aux spécialistes de domaines et aux terminologues. En même temps, nous ajoutons un aspect *multi-points de vue* à notre modèle afin de le rendre plus apte à supporter une approche évolutive.

Dans cet article, nous clarifierons tout d'abord la notion de point de vue; nous présenterons ensuite notre modèle de connaissances multi-points de vue, avec un accent particulier sur la définition des concepts; puis nous discuterons de la création d'une interface hypertextuelle pour explorer et manipuler une ontologie; enfin nous expliquerons la notion de distance entre concepts, qui est utilisée pour créer des liens de navigation supplémentaires, par rapport à ceux qui existent explicitement dans la logique descriptive.

2. La notion de point de vue

Alors qu'un domaine correspond à une unité thématique, un point de vue est plutôt lié à un type de personne (métier, âge, niveau de formation, etc.) ou d'utilisation (une même personne pourra avoir un point de vue différent en fonction de la tâche qu'elle cherche à accomplir). Un point de vue est un ensemble de définitions, chacune d'entre elles étant reliée à un concept différent. On peut envisager plusieurs types de relations entre points de vue. Par exemple :

points de vue par « proximité » / niveau

Selon le niveau de connaissance ou de formation *par rapport à un domaine*, on peut être plus ou moins sensible à certains détails. On peut donc avoir plusieurs hiérarchies de concepts avec une forme similaire, mais avec plus ou moins de niveaux. Si l'on prend par exemple le concept « cheval » dans le domaine de la zoologie, on peut avoir dans le point de vue « enfant », la classification : cheval → mammifère → vertébré → animal et dans le point de vue « zoologue » cheval → onguligrade avec nombre de doigts impair → onguligrade → ... Le point de vue « enfant » représente donc un sous-ensemble du point de vue « zoologue ».

point de vue = vue partielle

Chaque type d'utilisateur va s'intéresser à un ensemble restreint de caractéristiques. Par exemple, pour le domaine des voitures, dans le point de vue « mécanicien » les véhicules vont être définis selon le type de moteur, le diamètre des roues, etc. tandis que dans le point de vue « vendeur », les définitions vont plutôt contenir des informations sur le type de véhicule (tourisme, utilitaire, transport de groupes), la taille, la catégorie de prix, etc. Ce type de points de vue peut être envisagé comme une projection d'un espace vers un espace de plus petite dimension.

points de vue avec chevauchement

Des personnes travaillant dans des domaines voisins peuvent avoir des points de vue qui se recoupent. Par exemple si l'on considère le domaine des immeubles, on peut supposer qu'une partie des définitions de concepts sera identique pour un architecte et pour un ingénieur en génie civil, alors que certains autres concepts seront définis différemment (on peut aussi penser qu'une partie des concepts n'existeront que dans l'un ou l'autre des points de vue).

points de vue avec ordres de classification différents

Deux points de vue sur un domaine contiennent exactement les mêmes concepts, avec les mêmes définitions, mais l'ordre de classification est différent. Prenons par exemple le domaine des boissons. On peut avoir dans un point de vue une première distinction « alcoolisée/non alcoolisée » puis une deuxième « gazeuse/non gazeuse », puis « à base de fruits/à base de lait/à base d'eau », etc. Dans un autre point de vue, on peut prendre les mêmes distinctions, mais dans un ordre différent.

Il existe un certain nombre de modèles qui sont capables de traiter une approche multi-points de vue (KRL, LOOPS, ROME, VIEW, TROPES [MAR 93]), mais ils sont basés sur l'hypothèse qu'un point de vue est une représentation partielle d'un ensemble cohérent d'objets. Dans le modèle que nous proposons, chacune des situations décrites ci-dessus peuvent être gérées. Nous avons donc une vision beaucoup plus souple du point de vue.

3. Modèle de connaissance multi-points de vue

Dans cet article, nous envisageons une ontologie comme une représentation des concepts d'un *domaine* (contrairement, par exemple au projet CYC, qui cherche à réaliser une ontologie générale du monde). Un domaine correspond à une unité thématique.

De plus, nous travaillons uniquement avec des domaines « connus », c'est-à-dire qui ont une certaine maturité et par là même une certaine stabilité (par opposition avec des domaines nouveaux qui correspondent à des théories en cours d'élaboration et dont la terminologie n'est pas fixée). Si plusieurs écoles de pensées / conceptions s'affrontent à l'intérieur du domaine, nous pensons qu'il est préférable de les traiter comme des domaines séparés : le but n'est pas de forcer un consensus artificiel sur les définitions des concepts. Le rôle de l'ontologie n'est pas de normaliser un domaine, mais de donner une représentation de l'existant.

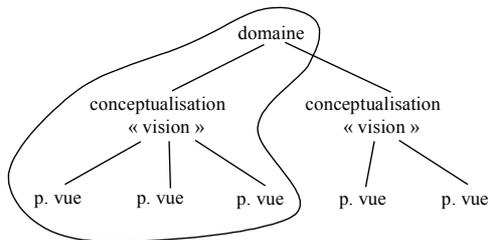


Fig. 1 : domaines, visions et points de vue

Nous utilisons ici le terme « domaine » pour désigner une *vision d'un domaine*. A l'intérieur d'une vision, on peut avoir plusieurs points de vue. En prenant ces points de vue deux à deux, il ne doit pas y avoir de contradictions/incompatibilités. Si deux définitions sont incompatibles, cela signifie qu'il n'existe pas d'objet qui puisse les satisfaire toutes les deux.

3.1. Points de vue et conflits

Notre modèle est inspiré des modèles ConcepTerm [BER 94] (pour les définitions) et IBIS [KUN 70] (pour la partie points de vue). Le modèle IBIS permet d'explorer les solutions possibles à une question ouverte. Chacun des participants peut proposer une réponse qui sera ensuite argumentée (positivement ou négativement). Dans le cas d'une ontologie, la question qu'on se pose est : « Quelle est la définition du concept x ? » L'idée est que chacun puisse exprimer son avis (en proposant une définition qui est rattachée au concept) et qu'ensuite, on essaie de se diriger progressivement vers un consensus, par exemple en rattachant chaque définition au point de vue adéquat. *Le but est d'obtenir une ontologie où, pour chaque concept, on a au maximum une définition par point de vue; les différentes définitions d'un concept devant être compatibles.*

Si le fait d'autoriser que plusieurs définitions soient rattachées à un même concept est intéressant, il ne faut pas oublier que certaines définitions peuvent remettre en cause la position d'un concept dans la hiérarchie (les concepts sont organisés en une hiérarchie générique-spécifique, avec héritage simple, donc avec une structure arborescente). Pour cette raison, la possibilité d'avoir plusieurs concepts désignés par le même terme, à différents emplacements dans la hiérarchie, a été rajoutée.

Même dans un modèle multi-points de vue, il peut arriver qu'il y ait des conflits d'opinions. On se trouve dans une situation conflictuelle si, à l'intérieur d'un point de vue, on a : 1) deux ou plusieurs définitions pour un même concept, 2) deux ou plusieurs concepts désignés par le même terme. Dans [FAL 00] on trouvera une méthode d'analyse et de résolution de ces conflits. Cette méthode énumère entre autres un certain nombre de règles qui permettent de tester la compatibilité de deux définitions. Par ailleurs, notre modèle inclut un mécanisme de

versions qui permet de gérer les modifications des définitions.

3.2. Modèle de concepts

Le langage formel de définition de concepts fait partie de la famille des logiques descriptives (ou logiques terminologiques). L'intérêt de ce formalisme est qu'il est d'une part suffisamment simple pour que des non spécialistes puissent l'utiliser (comme l'a montré notre expérience avec des terminologues dans le projet ConcepTerm) et, d'autre part, qu'il est plus expressif que les modèles à objets ou "sémantiques" du type entité-association comme UML ou RDF [LAS 98].

Un concept peut être défini de trois manières :

- par conjonction ou disjonction de deux concepts
- par négation d'un concept
- par un lien, appelé rôle, vers un autre concept

Par exemple, l'énoncé

partie: "pointe"

définit le concept recouvrant toutes les choses qui ont une partie qui est une pointe. Le rôle peut être quantifié universellement pour signifier l'exclusivité. Ainsi l'énoncé

all usage: "écriture"

définit le concept formé de toutes les choses qui n'ont pas d'autre usage que l'écriture.

On peut également ajouter une contrainte d'inclusion entre concepts, qui correspond à la relation générique-spécifique. Ceci permet d'écrire des définitions de concept telles que :

concept quaternion

generic nombre

(partie (1,1) : "nombre réel"

and partie (1,1): "vecteur"

and operation: "addition"

and operation: "multiplication"

)

"Un quaternion est un nombre formé d'au moins et d'au plus (indication (1,1)) une partie réelle et d'au moins et d'au plus une partie vectorielle. On peut appliquer les opérations d'addition et de multiplication sur un quaternion". Si l'on voulait être plus précis et indiquer qu'un quaternion n'a pas d'autres parties il faudrait ajouter le rôle :

and all partie: ("nombre réel" **or** "vecteur")

La sémantique formelle de ce langage peut être définie exactement de la même manière que celle des logiques terminologiques de type ALCNR [BUC 93], on la trouvera dans [FAL 00]. Le principe de l'interprétation est que l'interprétation d'un concept est un ensemble d'individus et l'interprétation d'un rôle est une relation

entre individus. Ces interprétations doivent satisfaire les définitions des concepts (par exemple, pour chaque individu *c* appartenant au concept *quaternion* on doit avoir un et un seul individu *n* du concept *nombre réel* tel que (*c, n*) appartient à la relation correspondant au rôle *partie*).

Afin de créer des bases de connaissances terminologiques exploitables, il est nécessaire de réifier les définitions de concepts, ce qu'on peut faire sous forme de graphes orientés étiquetés, de la manière suivante :

- Un nœud du graphe représente un concept défini par conjonction, disjonction ou négation. Il est étiqueté par le nom du concept (s'il en a un) et par l'opérateur logique utilisé. Le nœud est lié par des arcs de type "composition" aux concepts qui le composent et par un arc de type "généralisation" à son concept générique.
- Un concept défini par un rôle est représenté par un arc de type "role" qui porte le nom du rôle et pointe vers un autre concept. L'arc peut être étiqueté par le quantificateur all et par une contrainte de nombre d'occurrences minimum et maximum. La définition de quaternion ainsi représentée devient :

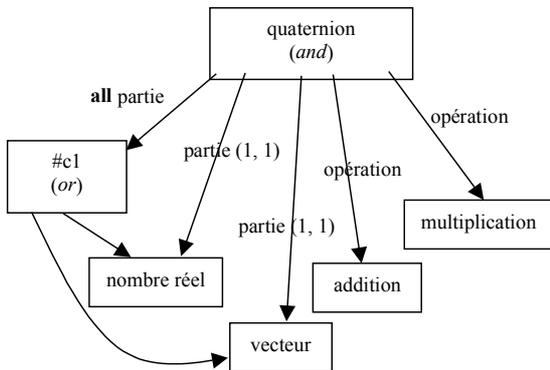


Fig. 2 : exemple de définition

Cette réification permet :

- de stocker les définitions de concepts dans une base de données (relationnelle ou semi-structurée)
- de faire référence aux définitions en tant qu'objets et donc de les associer à d'autres objets représentant les termes, les points de vue, les auteurs, etc.
- d'appliquer des traitements de type graphe (parcours de chemins, recherches de chemins, etc.) pour implémenter diverses fonctions sur la base de concepts (déductions, calculs de distances sémantiques et syntaxiques, réorganisation, génération de vues, etc.)

Le diagramme ci-dessous montre l'insertion des objets Concept et LienSémantique dans le modèle orienté-objet.

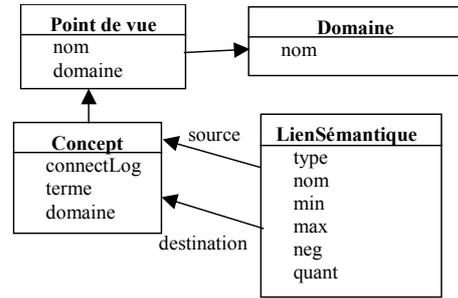


Fig. 3 : diagramme de classes

La définition de concepts dans un domaine particulier fait toujours usage de concepts généraux, ou d'un autre domaine, que l'on considère comme univoquement définis. Par exemple, la définition du concept *armoire* dans le domaine des meubles fait référence aux concepts *haut*, *ranger*, *linge*, *vêtement*, qui n'appartiennent pas au domaine des meubles. On peut raisonnablement supposer que le travail terminologique dans un domaine particulier n'est possible que s'il y a accord sur les concepts hors du domaine. C'est pourquoi nous considérerons ceux-ci comme atomiques. La base de concept est donc formée d'une base de travail, celle que l'on est en train de construire ou de modifier, et d'une ou plusieurs autres bases auxiliaires considérées comme fixes (il peut s'agir, par exemple, d'une ontologie générale telle que WordNet).

Ceci étant donné, on dira que l'*arbre de définition* d'un concept est le sous-graphe maximum que l'on obtient en partant du nœud représentant ce concept et en parcourant le graphe jusqu'à atteindre des concepts atomiques. La définition étendue est donc un arbre dont les feuilles sont des concepts atomiques. Il s'agit bien d'un arbre si l'on exclut les définitions autoréférentes.

4. Construction de l'interface

Le travail sur l'ontologie nécessite une interface sophistiquée qui aille bien au delà du simple affichage de la hiérarchie des concepts. Il s'agit en effet d'offrir plusieurs vues de la base, adaptées aux différentes tâches et actions de l'utilisateur. La construction et la maintenance de l'ontologie nécessitent en particulier de pouvoir comparer des définitions de concepts, de rechercher des définitions similaires ou encore de retrouver le terme correspondant à un concept dont on connaît (partiellement) la définition (recherche dite onomasiologique par opposition à la recherche sémasiologique habituelle qui va du terme à sa définition).

La réalisation de vues de l'ontologie qui "supportent" ce type d'actions met en jeu essentiellement trois mécanismes :

1. le parcours de l'espace des concepts en suivant les liens de généralisation, de composition ou de rôles

2. la comparaison de concepts
3. le calcul de distances sémantiques entre concepts (et l'inférence de liens basés sur les distances)

Plutôt que de définir une interface ad hoc et figée, nous proposons de construire l'interface de l'ontologie comme une vue hypertextuelle (un hypertexte virtuel) qui peut être augmentée ou modifiée en fonction des besoins propres à un domaine ou en fonction de nouvelles idées de présentation et de navigation.

Pour réaliser cet objectif de flexibilité, nous utilisons une approche déclarative [ATZ 97], [FER 98], [FAL 99], qui consiste à définir la vue hypertextuelle par un ensemble de schémas de nœuds hypertextes. Un schéma de nœud comprend essentiellement :

- une expression de sélection qui fournit un ensemble d'objets de l'ontologie (concepts, liens sémantiques, etc.)
- des expressions de construction de contenus pour créer des éléments (de type XML) à partir des objets sélectionnés et de constantes
- des expressions de création de liens vers d'autres nœuds (liens de référence, d'inclusion immédiate ou d'inclusion différée)
- des paramètres

La vue hypertextuelle est composée d'instances de nœuds, qui sont obtenues à partir de schémas en fixant des valeurs de paramètres. L'exemple suivant montre un schéma de nœud, exprimés dans le langage Lazy [FAL 99] et une instance de nœud.

exemple : visualisation des définitions

Les relations (tables) suivantes sont utilisées pour le stockage des définitions :

relation **concept** (concept_id, terme, connecteurLogique, pointDeVue)

relation **lienSem** (source, destination, type, min, max, negation, all)

relation **distance** (concept_id1, concept_id2, distance)

Un nœud de type Concept affiche le terme correspondant au concept, le connecteur logique employé et tous les liens (rôles ou « compositions ») servant à définir le concept.

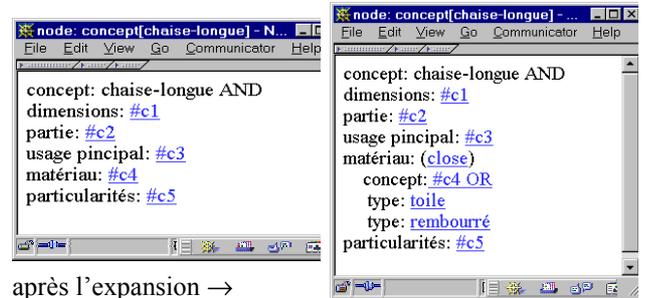
```
node Concept[c]
  "concept: ", terme, connecteurLogique,
  {include Lien[concept_id]}
  from concept selected by concept_id=c
```

Un nœud de type Role affiche tous les liens sémantiques issus du concept posé en paramètre.

```
node Role[c]
  negation, all, "<", min, ",", max, ">", type, expand href
  Concept[destination] (concept_id)
```

from lienSem, concept selected by source=c and concept_id=c

Un lien « expand » correspond à une inclusion différée.



après l'expansion →

Fig. 4 : nœud pour l'affichage d'une définition

Il faut noter que pour obtenir une interface efficace, il n'est pas suffisant de créer des vues hypertextuelles qui reproduisent simplement la structure du graphe des concepts, sinon on n'obtiendrait rien d'autre qu'un outil d'exploration « pas à pas » d'une structure de données. Les exemples qui suivent montrent comment créer des vues variées sur l'ontologie.

exemple : visualisation des concepts génériques

L'inclusion récursive de nœuds permet des parcours d'arbres impossibles avec les langages de requêtes relationnels (sauf extension adhoc). Le nœud Generique permet de remonter dans la hiérarchie de concepts, jusqu'à la racine, en suivant les liens de type « générique ».

```
node Generique[c]
  terme, "-> "
  include Generique[destination]
  from lienSem, concept selected by source=c and
  type="generic" and concept_id=destination
```



Fig. 5 : nœud pour l'affichage récursif des concepts génériques.

exemple : création de nœuds hypertextes complexes

Après avoir constitué une bibliothèque de nœuds de base (tels que ceux qui ont été présentés ci-dessus), il est possible de les combiner afin de créer des vues plus compactes. Pour ce faire, on utilisera l'inclusion de nœuds. L'exemple ci-dessous montre un nœud qui inclut plusieurs nœuds simples; ces derniers pouvant à leur tour utiliser des inclusions immédiates ou différées. Le nœud ci-dessous donne de nombreuses possibilités pour l'exploration de l'ontologie : on utilise tout d'abord des nœuds qui permettent de situer le concept dans la

hiérarchie en affichant les concepts génériques et spécifiques et en cherchant les concepts proches (voir section suivante pour le calcul des distances). D'autres nœuds ont été ensuite ajoutés pour offrir des fonctionnalités supplémentaires.

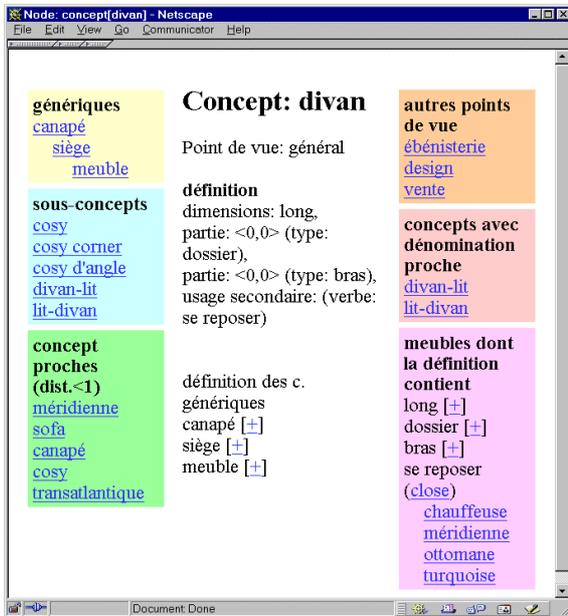


Fig. 6 : nœud complexe

5. Support à la navigation sémantique

Si les liens sémantiques entre concepts forment le premier support de la navigation dans l'ontologie, ceux-ci sont insuffisants pour offrir une navigation basée sur la proximité sémantique ou définitoire entre concepts. Afin de pouvoir déterminer cette proximité, il est nécessaire de pouvoir effectuer des comparaisons de définitions et de définir des distances.

Plusieurs auteurs ont proposés des notions de distance dans les structures conceptuelles (on trouvera dans [CHA 00] une revue des principales approches dans ce domaine). La plupart de ces définitions sont basées, au moins en partie, sur la position des concepts dans un arbre ou treillis de généralisation/spécialisation. Ces distances s'appliquent donc bien dans le cas où la hiérarchie des concepts est bien formée, c'est-à-dire dans une ontologie ou base de concepts stabilisée. Or, dans une phase de construction ou de restructuration d'une ontologie, la structure hiérarchique des concepts peut être incomplète ou incohérente. C'est pourquoi ce type de distance sémantique n'est pas directement applicable et doit être remplacé par une distance basée sur la comparaison de la structure même des définitions de concepts, indépendamment de leur position dans la hiérarchie de généralisation.

La notion de distance que nous utilisons est décrite formellement dans [FAL 00], elle est basé sur la

comparaison des arbres de définitions. Le résultat d'une comparaison est un arbre étiqueté qui représente les opérations d'ajout (+), de suppression (-) et de remplacement (ρ) à effectuer pour passer d'une définition à l'autre. Par exemple : si l'on veut obtenir la définition

concept quaternion2

generic nombre

```
(
  partie (4,4) : "nombre réel"
  and operation: "addition vectorielle"
  and all partie: "nombre réel"
)
```

à partir de la définition de quaternion donnée précédemment (cf. section 3.2), on pourra appliquer la modification.

modification

```
(
  partie [ $\rho$ ](1,1) --> (4, 4): "nombre réel"
  and [-] partie (1,1): "vecteur"
  and operation: [ $\rho$ ] "addition" --> "addition vectorielle"
  and all partie: ("nombre réel" or [-] "vecteur")
)
```

Chaque modification a un poids qui dépend de la hauteur des opérations de modification dans l'arbre syntaxique de la définition (voir figure 2), plus une opération est proche de la racine, plus elle est importante. La distance entre deux concepts est définie comme le poids de la plus légère des modifications d'un concept à l'autre. Dans ce cas on a considéré que les concepts *nombre réel*, *addition vectorielle* et *addition* sont atomiques et qu'ils sont donc définis dans une ontologie auxiliaire existante (par exemple WordNet). Le poids du remplacement de l'un de ces concepts par un autre sera donné par leur distance sémantique dans leur ontologie (par exemple la distance de Sowa modifiée). Si les deux concepts à comparer n'ont pas le même générique, on étend leur définition par conjonction avec les génériques, jusqu'au plus petit générique commun.

Cette opération de comparaison permet à la fois de voir où se situent les différences entre concepts et de définir une distance, et donc une navigation parmi les définitions. Cette opération permet également de rechercher une définition à partir d'une définition (partielle) donnée comme modèle, par exemple pour voir s'il existe déjà une définition proche de celle qu'on désire ajouter dans la base.

6. Conclusions

Dans cet article, nous avons tout d'abord présenté un modèle de représentation de connaissances multi-points de vue basé sur une logique descriptive. En utilisant un outil de publication de bases de données (Lazy), nous avons ensuite montré qu'il est possible de créer diverses vues hypertextuelles en combinant un ensemble de

schémas de nœuds de base. Pour créer des liens entre les nœuds représentant les concepts, on utilise à la fois les liens faisant partie intégrante de la logique descriptive, mais aussi des liens calculés, par exemple des liens de proximité sémantique, obtenus par comparaison des définitions. Nous testons actuellement notre approche avec un groupe d'étudiants, pour nous assurer que des utilisateurs ni spécialistes de l'ingénierie des connaissances, ni terminologues sont réellement à mêmes de manipuler des ontologies à l'aide de vues hypertextuelles.

travaux futurs

Le langage de spécification de vues hypertextuelles permet de remédier à certaines limitations des langages de requêtes relationnels sur les structures d'arbres et de graphes, essentiellement à l'aide d'inclusions de nœuds. Pour simplifier la spécification de l'interface nous envisageons cependant de remplacer les prédicats de sélection de type SQL que nous utilisons par des expressions de type XQuery mieux adaptées au parcours dans les graphes de concepts et par des constructions spécifiques aux structures conceptuelles.

7. Références

- [ATZ 97] P. ATZENI, G. MECCA, P. MERIALDO, « To Weave the Web », in Procs VLDB97, Athens, 1997
- [BER 94] D. BERTHET, M. BONJOUR, B. DE BESSE, G. FALQUET, M. LEONARD, J. SINDAYAMAZE, « ConcepTerm "Construction de dictionnaires encyclopédiques multilingues et informatisés" », rapport technique, CUI, Université de Genève, 1994
- [BUC 93] M. BUCHHEIT, F. DONINI, A. SCHAEFER, « Decidable Reasoning in Terminological Knowledge Representation Systems », in Journal of Artificial Intelligence Research, 1993
- [CHA 00] S. CHABERT-RANWEZ, « Composition Automatique de Documents Hypermédia Adaptatifs à partir d'Ontologies et de Requêtes Intentionnelles de l'Utilisateur. », Thèse de doctorat, Université Montpellier II, décembre 2000.
- [DOM 98] J. DOMINGUE, « Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. », in Procs. 11th Knowledge Acquisition Workshop (KAW98), Banff, Canada, 1998
- [EUZ 96] J. EUZENAT, « Corporate memory through cooperative creation of knowledge bases and hyperdocuments », in Procs KAW 96, Banff, Canada
- [FAL 00] G. FALQUET, C.-L. MOTTAZ JIANG, « Conflict Resolution in the Collaborative Design of Terminological Knowledge Bases », In Proc. EKAW2000, Juan-les-Pins, France, 2000, LNAI 1937, Springer Verlag
- [FAL 99] G. FALQUET, L. NERIMA, J. GUYOT, « Languages and Tools to Specify Hypertext Views on Databases. », In The World Wide Web and Databases, P. Atzeni, A. Mendelzon, G. Mecca (eds.), LNCS vol 1590, Springer 1999
- [FAR 96] A. FARQUHAR, R. FIKES, J. RICE, « The Ontolingua Server: a Tool for Collaborative Ontology Construction », in Procs. 10th Knowledge Acquisition Workshop (KAW96), Banff, Canada, 1996
- [FER 98] M. FERNANDEZ, D. FLORESCU, J. KANG, A. LEVY, D. SUCIU, « Catching the Boat with Strudel : Experiences with a Web-Site Management System », in Procs ACM SIGMOD Conf. Seattle, pp 414-425, 1998
- [FOO 92] N. FOO, B. J. GARNER, A. RAO, E. TSUI, « Semantic Distance in Conceptual Graphs », in Current Directions in Conceptual Structure Research, ed. L. Gerhotz, pp. 149-54, Ellis Horwood, 1992.
- [KUN 70] W. KUNTZ, H. RITTEL, « Issues as elements of information systems », Working Paper No 131, Institute of Urban and Regional Development, University of California at Berkeley, 1970
- [LAS 98] O. LASSILA, R. SWICK, « Resource Description Framework (RDF) Model and Syntax Specification », W3C Working Draft, 08 October 1998
- [MAR 93] O. MARIÑO DREW, « Raisonnement classificatoire dans une représentation à objets multi-points de vue », thèse de doctorat, 1993, Université Joseph Fourier, Grenoble
- [MOR 00] A. MORENO ORTIZ, « Managing conceptual and terminological information in a user-friendly environment », In Procs. OntoLex 2000, Sept. 2000, Sozopol, Bulgaria
- [OIL] OILED, <http://img.cs.man.ac.uk/oil/>
- [ONT] ONTOEDIT, <http://ontoserver.aifb.uni-karlsruhe.de/ontoedit/>
- [STU 98] R. STUDER, V. BENJAMINS, D. FENSEL. « Knowledge Engineering: Principles and Methods. » in Data and Knowledge Engineering. 25, 1998 161- 197
- [SWA 96] B. SWARTOUT, R. PATIL, K. KNIGHT, T. RUSS, « Toward Distributed Use of Large-Scale Ontologies », in Procs KAW96, Banff, Canada, 1996
- [TEN 98] J. TENNISON, N. R. SHADBOLT, « APECKS: a Tool to Support Living Ontologies », in Procs KAW98, Banff, Canada.