Opérationnalisation d'une ontologie: une méthode et un outil

Frederic Furst

Opérationalisation des ontologies : une méthode et un outil

Frédéric Fürst

Laboratoire d'Informatique de Nantes Atlantique (CNRS-FRE 2729)
2 rue de la Houssinière - BP 92208 - 44322 Nantes CEDEX 03
Frederic.Furst@lina.univ-nantes.fr

Résumé : Le travail présenté dans cet article relève de l'intégration des ontologies de domaine au sein de Systèmes à Base de Connaissances (SBC). La mise en œuvre d'une ontologie opérationnelle de la géométrie nous a permis de dégager des règles d'*opérationalisation* d'une ontologie dans le modèle des Graphes Conceptuels. Ces règles ont été implémentées dans le cadre d'un outil d'édition et d'opérationalisation d'ontologie baptisé TooCoM.

Mots-clés : Ontologies, Opérationalisation, Graphes Conceptuels.

1 Introduction

Les ontologies sont apparues dans le domaine de l'Ingénierie des Connaissances (IC) afin de permettre la représentation de connaissances au niveau conceptuel, indépendamment des usages auxquels pouvaient être destinés les systèmes intégrant ces connaissances. Cette approche tranche avec celle des systèmes experts à base de règles où la forme sous laquelle les connaissances sont représentées est déterminée par l'usage, essentiellement la résolution de problèmes. Cette indépendance entre la représentation des connaissances et leur utilisation opérationnelle permet ainsi l'intégration dans les Systèmes à Base de Connaissances (SBC) de connaissances indépendantes des mécanismes opératoires pouvant être mis en œuvre sur ces dernières.

Cependant, les ontologies de domaine (limitée à un champ précis de la connaissance) sont conçues pour supporter ou rendre possible le traitement de requêtes, l'association entre documents, la communication entre agents logiciels sur le Web ou le raisonnement au sein de Web Services, comme le décrivent par exemple les spécifications de l'Ontology Web Language (OWL, 2002). Puisque les ontologies doivent être utilisées comme ressource pour le raisonnement, elles doivent pouvoir offrir des formes opérationnelles adaptés aux différents usages opérationnels, tout en restant des représentations de connaissance au niveau conceptuel (Charlet, 2003). L'utilisation opérationnelle d'une ontologie nécessite donc, pour

chaque application différente, son opérationalisation, c'est-à-dire sa transcription dans un langage opérationnel de représentation de connaissance (i.e. un langage doté d'une sémantique opérationnelle et susceptible d'être exécuté sur machine) selon un scénario d'usage précis.

Le problème est alors de définir les mécanismes d'opérationalisation permettant cette transcription. Dans cet article, nous proposons une méthode d'opérationalisation d'ontologie testée lors d'une expérience d'opérationalisation d'une ontologie de la géométrie (Fürst *et al.*, 2003) dans le cadre du modèle des Graphes Conceptuels (Sowa, 1984). Cette expérience a conduit à la définition de règles d'opérationalisation d'une ontologie de domaine et au développement d'un outil support à la mise en oeuvre de ces règles d'opérationalisation.

La suite de cet article est structurée comme suit. Après avoir introduit la notion d'ontologie et avoir souligné l'importance d'un de ses composants essentiels, à savoir les axiomes (cf. section 2), la nécessité d'une étape d'opérationalisation des ontologies est affirmée, et les différents scénarii d'usage d'une ontologie précisés (cf. section 3). La section 4 expose une méthode formelle d'opérationalisation dans le cadre du modèle des Graphes Conceptuels, implémentée dans l'outil TooCoM d'édition et d'opérationalisation d'ontologie. Finalement, l'application des mécanismes d'opérationalisation proposés à la validation des ontologies est explicitée.

2 Les éléments constitutifs d'une ontologie de domaine

Une ontologie de domaine est traditionellement définie comme une *spécification d'une conceptualisation partagée* (Gruber, 1993). Elle décrit les connaissances d'un domaine à travers un paradigme conceptuel non opérationnel. Toujours liée à un domaine particulier de connaissances (i.e. un ensemble borné et cohérent de connaissances doté d'une sémantique consensuelle), une ontologie contient tout d'abord les primitives terminologiques du domaine, c'est-à-dire le vocabulaire conceptuel. Dans le cadre du paradigme Entité-Relation (Chen, 1976) utilisé dans nos travaux, le vocabulaire conceptuel est structuré en un ensemble de **concepts**, représentant les objets du domaine, et un ensemble de **relations** existant entre ces concepts.

A ce niveau terminologique doit s'ajouter un ensemble d'**axiomes** exprimant la sémantique du domaine. Les **axiomes** représentent les connaissances n'ayant pas un caractère strictement terminologique et spécifient la façon dont les primitives terminologiques du domaine (i.e. les concepts et relations) peuvent être utilisées (Staab & Maedche, 2000). Bien que peu intégrés aux ontologies jusqu'à présent, les axiomes sont essentiels à la spécification de connaissances ontologiques et distinguent les ontologies des thesaurus, qui ne représentent que des terminologies alors que les ontologies intègrent des connaissances au sens large (Charlet *et al.*, 2003). Ce sont les axiomes qui permettent véritablement l'utilisation opérationnelle des ontologies car « *un système ne peut exploiter un concept*

qu'en fonction des opérations ou règles qu'il peut lui associer » (Bachimont, 2000).

Les axiomes peuvent représenter des propriétés communes liées aux concepts et aux relations, c'est-à-dire des propriétés vérifiées par de nombreuses primitives conceptuelles dans de nombreux domaines de connaissances. Ces propriétés, que nous proposons d'appeler *schémas d'axiomes*, peuvent être les propriétés algébriques d'une relation (symétrie, réflexivité, transitivité), la propriété de *subsomption* entre concepts et/ou entre relations, la généricité d'un concept[1], la signature et la cardinalité d'une relation, l'exclusivité et l'incompatibilité entre primitives (l'incompatibilité entre deux primitives P_1 et P_2 ets formalisée par la formule $\neg(P_1 \wedge P_2)$, l'exclusivité par la formule $P_1 \Rightarrow \neg P_2$). Par exemple, dans le domaine de la géométrie, l'axiome 1.3.1 de HILBERT « *Sur une droite il y a au moins deux points* » est une propriété de cardinalité : la relation d'appartenance d'un point à une droite porte une propriété de cardinalité minimum de 2 par rapport à la droite. Certains schémas d'axiomes sont intégrées dans les formalismes de représentation de connaissances utilisés pour décrire des ontologies. Par exemple, la relation *sorte-de* apparaît aussi bien dans les langages utilisant le paradigme Entité-Relation (par exemple les Graphes Conceptuels) que dans ceux qui utilisent le paradigme des Frames (par exemple OIL) (Gomez-Perez *et al.*, 1998).

Des propriétés propres au domaine de connaissances considéré peuvent également apparaître et être incluses dans l'ontologie. Ainsi, l'axiome 2.1.1 « *Si un point B est entre un point A et un point C, les points A, B et C appartiennent à une même droite* » ne relève pas d'un schéma d'axiome classique. Cependant, il doit être inclus dans l'ontologie car il participe à la définition de la sémantique du domaine.

Ainsi, les axiomes spécifient la sémantique des primitives conceptuelles, c'est-à-dire la façon dont les primitives sont utilisées pour exprimer des connaissances dans le domaine. Mais pour utiliser les axiomes dans un SBC, la sémantique de manipulation de ceux-ci doit aussi être précisée, c'est-à-dire la façon dont les axiomes sont utilisés dans le cadre de l'application envisagée. Puisque cette sémantique dépend de l'objectif opérationnel de l'application, elle ne peut être incluse dans l'ontologie, qui doit demeurer indépendante de tels objectifs. La spécification de cette sémantique conduit, à travers un processus d'*opérationalisation*, à une *ontologie opérationnelle*, aussi dénommée *ontologie computationnelle* dans certains travaux (Bachimont, 2000).

3 L'opérationalisation d'une ontologie de domaine

Une ontologie n'est qu'une représentation conceptuelle des connaissances d'un domaine, indépendante des différentes utilisations opérationnelles possibles. La sémantique formelle dont est dotée l'ontologie sert à exprimer la sémantique de

1. un concept générique, ou abstrait, n'admet pas d'instance.

manipulation des primitives conceptuelles, sémantique qui est liée au domaine. Afin d'intégrer une ontologie au sein d'un SBC, il convient donc de transcrire celle-ci dans une forme adaptée à l'usage auquel est destiné le SBC. Cette **opérationalisation** consiste donc, d'une part, à choisir un langage de représentation de connaissances offrant des mécanismes de manipulation adaptés à l'objectif opérationnel considéré et, d'autre part, à adapter la représentation de l'ontologie à cet objectif en précisant la sémantique de manipulation des axiomes, sémantique qui est liée à l'application envisagée et non au domaine considéré. Opérationaliser une ontologie dans un SBC, c'est donc la plonger dans un langage opérationnel de représentation de connaissances suivant un scénario d'usage décrivant l'objectif opérationnel du SBC.

L'opérationalisation d'une ontologie ne se conçoit que pour une utilisation opérationnelle bien définie, caractérisée par un *scénario d'usage* précis (Gruninger & Fox, 1995). Un scénario d'usage décrit à quelles fins vont être utilisées les connaissances spécifiées dans l'ontologie, c'est-à-dire essentiellement à quoi vont servir les axiomes de l'ontologie. En effet, la représentation des connaissances terminologiques du domaine ne dépend pas des multiples contextes applicatifs possibles, la représentation d'un concept ou d'une relation étant la même dans le cas d'un système dédié à la validation de connaissances ou d'un système dédié à la production de connaissances. Seules les représentations opérationnelles des axiomes doivent être adaptées à l'objectif de l'application envisagée.

Ainsi, un axiome peut être utilisé, soit pour **inférer** de nouvelles connaissances, soit pour **valider** l'adéquation d'une connaissance par rapport à la sémantique du domaine considéré. Par exemple, l'axiome 1.6 de HILBERT « *Si deux points A et B d'une droite a appartiennent à un plan α, alors tous les points de la droite a appartiennent à α* » peut être utilisé pour inférer l'appartenance de points à un plan. Il peut également permettre de déterminer qu'un énoncé n'est pas conforme à la sémantique de la géométrie si, dans un contexte où deux points appartiennent à une droite et à un plan, un troisième point de la droite n'appartient pas au plan.

D'autre part, un axiome peut n'être utilisé qu'à la demande de l'utilisateur du SBC, ou être appliqué automatiquement par le système. Le premier usage est qualifié d'**explicite**, le second d'**implicite**. L'axiome 1.3.1 « *Sur une droite il y a au moins deux points* » peut ainsi être utilisé de façon implicite pour ne pas obliger l'utilisateur à appliquer systématiquement cet axiome avant de pouvoir considérer des points sur une droite, ou au contraire de façon explicite si on désire qu'il y recoure systématiquement avant de pouvoir manipuler les points en question, par exemple dans un but pédagogique.

L'opérationalisation d'une ontologie nécessite donc le choix, pour chaque axiome, d'un *contexte d'usage* qui précise à quoi va servir l'axiome et de quelle façon il va être mis en œuvre. Les différents contextes d'usage que nous proposons de considérer sont :

- le contexte d'usage **inférentiel et explicite** où l'utilisateur déclenche l'application de l'axiome sur une base de faits pour produire de nouvelles assertions ;

- le contexte d'usage **inférentiel et implicite** où l'axiome est appliqué par le système sur une base de faits pour produire de nouvelles assertions ;
- le contexte d'usage **de validation explicite** où l'application de l'axiome est déclenchée par l'utilisateur pour contrôler la conformité sémantique des faits d'une base par rapport au domaine ;
- le contexte d'usage **de validation implicite** où l'axiome est appliqué par le système pour contrôler la conformité sémantique des faits d'une base par rapport au domaine.

Un scénario d'usage consiste ainsi en un ensemble de contextes d'usage choisis pour chaque axiome présent dans l'ontologie. Dans le cas des schémas d'axiomes, il est possible de fixer par défaut le même contexte d'usage pour tous les axiomes de l'ontologie relevant de ce schéma. Par exemple, les axiomes exprimant une symétrie de relation peuvent être tous utilisés pour inférer de nouvelles relations.

De manière générale, une forme opérationnelle d'une ontologie combine des mécanismes inférentiels et des mécanismes de validation permettant de manipuler les connaissances de façon plus ou moins automatique. Par exemple, un scénario correspondant à une application d'enseignement assisté par ordinateur devra offrir à l'utilisateur la possibilité d'appliquer les connaissances du domaine pour en déduire de nouvelles ou pour valider son travail, et prévoir des inférences et des contrôles automatiques en fonction, entre autres, du niveau de l'apprenant. Deux cas extrêmes de scénario d'usage peuvent être distingués : les scénarii de pure validation, où les connaissances ontologiques sont utilisées pour valider une base de faits par rapport à la sémantique d'un domaine, et les scénarii inférentiels implicites où les connaissances ontologiques sont utilisées pour produire de nouvelles connaissances sur un cas donné, sans intervention de l'utilisateur. Dans ce dernier cas, qui est celui des systèmes experts, les inférences automatiques sont censées produire des connaissances conformes à la sémantique du domaine et aucune validation n'est nécessaire.

La figure 1 présente le schéma général d'application des formes opérationnelles des axiomes. Ce schéma repose sur une étape où l'utilisateur ajoute des faits (1), une étape d'inférences guidées par l'utilisateur (2), une étape d'inférences automatiques (3) et une étape de validation semi-automatique ou automatique en fonction de l'usage ou non d'axiomes de contexte de validation explicite (4).

4 L'opérationalisation des axiomes à l'aide du modèle des Graphes Conceptuels

Dans cette section, nous proposons une méthode d'opérationalisation fondée sur les principes énoncés précédemment mais dans le cadre particulier du modèle des Graphes Conceptuels (GCs). Cette méthode est inspirée d'une expérience pratique d'opérationalisation d'une ontologie de la géométrie à l'aide d'une extension du modèle des GCs, la SG-family (Baget & Mugnier, 2002). La SG-family

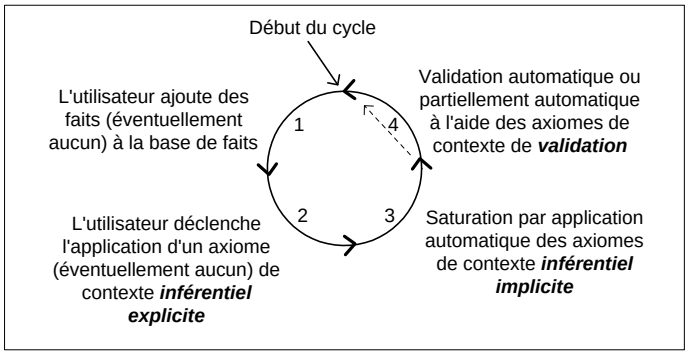

FIG. 1 – *Le cycle de raisonnement dans un SBC utilisant une ontologie opérationnelle.*

est un formalisme opérationnel de représentation de connaissances qui permet, d'une part, de représenter des primitives conceptuelles sous forme de concepts et de relations et, d'autre part, de représenter les axiomes exprimant la sémantique de manipulation de ces primitives à l'aide de règles et de contraintes de Graphes Conceptuels. Cette méthode a été implémentée dans un outil d'édition et d'opérationalisation d'ontologie baptisé TooCoM, a Tool to Operationalize an Ontology with the Conceptual Graph Model (Fürst, 2003). Cet outil permet l'édition graphique d'une ontologie dans le cadre du paradigme Entité-Relation et son opérationalisation dans le modèle des GCs après spécification du scénario d'usage choisi. Un moteur d'inférence basé sur la librairie de manipulation de graphes conceptuels CoGITaNT (Genest & Salvat, 1998) permet de mettre en œuvre le cycle de raisonnement présenté en figure 1.

La définition du vocabulaire conceptuel doit précéder la spécification des axiomes du domaine. Le vocabulaire conceptuel est structuré selon le paradigme Entité-Relation en *types de concepts* et *types de relations*, inclus respectivement dans une hiérarchie de types de concepts et une hiérarchie de types de relations structurés par une relation *sorte-de* (les termes *concept* et *relation* désignent ici des instances des types de concepts et types de relations). Des instances des types de concepts peuvent être définies dans une ontologie, mais ces instances doivent être ontologiques, c'est-à-dire qu'elles jouent un rôle dans la définition de la sémantique du domaine. Par exemple, π est une instance ontologique du type de concept nombre dans l'ontologie de la trigonométrie.

Les concepts et relations peuvent être utilisés pour écrire des graphes exprimant des faits où les concepts sont des noeuds du graphe et les relations des arêtes. Un couple de graphes permet de représenter un axiome, avec un graphe hypothèse et un graphe conclusion. Les propriétés de subsomption entre primitives et les signatures des relations sont des axiomes directement intégrés dans le modèle. Ils permettent de comparer les concepts et relations et, de manière générale, les graphes par une opération de *projection* (Mugnier & Chein, 1996). Toutes les autres propriétés doivent être décrites par un couple de graphes conceptuels,

un graphe hypothèse et un graphe conclusion, les deux graphes étant liés par des sommets concepts communs. La figure 2 présente un axiome du domaine de la géométrie. Il est important de noter que la représentation de cet axiome demeure au niveau ontologique puisque sa forme ne conditionne pas l'usage qui en sera fait.

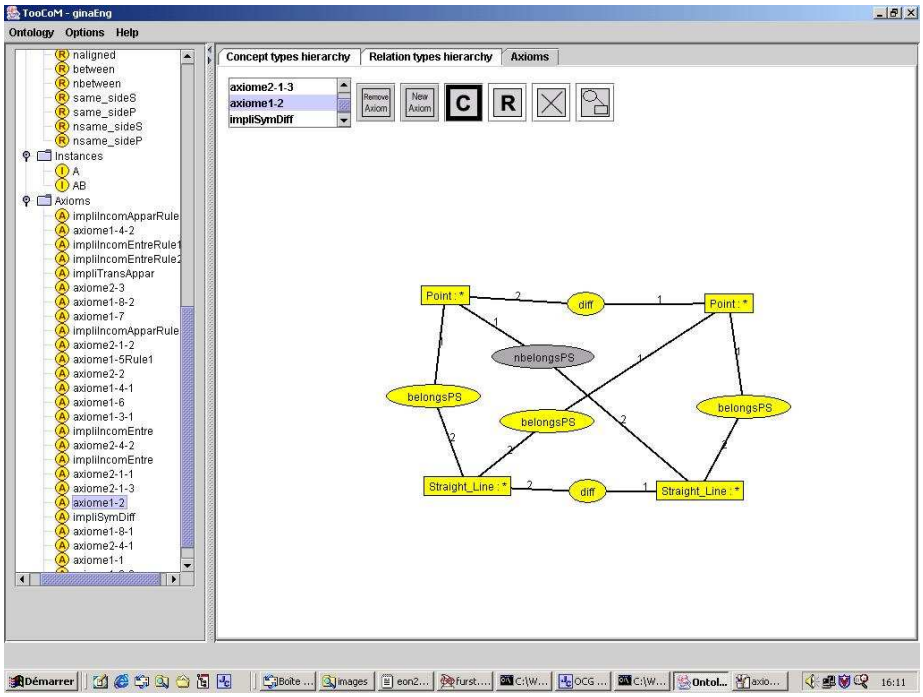

FIG. 2 – *Spécification d'un axiome à l'aide de TooCoM. Les concepts et relations apparaissant sur fond clair représentent l'antécédent de l'axiome, les concepts et relations apparaissant sur fond sombre représentent le conséquent. La sémantique de cet axiome est la suivante : s'il existe deux points différents et deux droites différentes tels que le premier point appartient à chacune des droites et que le deuxième point appartient à la première droite, le deuxième point n'appartient pas à la deuxième droite.*

Les axiomes exprimant les propriétés algébriques des types de relations peuvent être spécifiés simplement en indiquant la propriété en question dans la boîte de propriété du type de relation. L'axiome correspondant sera alors généré automatiquement et ajouté dans le panneau des axiomes. Les axiomes ne correspondant pas à un schéma prédéfini peuvent être spécifiés dans le panneau des axiomes. Pour plus de détails sur l'outil TooCoM, nous renvoyons le lecteur à (Fürst, 2003).

La représentation opérationnelle des axiomes repose sur les trois types de primitives de raisonnement offertes par la SG-family : (1) les **contraintes posi-**

tives, avec un graphe hypothèse et un graphe conclusion dont la sémantique est *si l'hypothèse est présente dans un graphe G, la conclusion doit également être présente dans G* (sinon la contrainte n'est pas respectée et G n'est pas valide dans le domaine) ; (2) les **contraintes négatives**, avec un graphe hypothèse et un graphe conclusion dont la sémantique est *si l'hypothèse est présente dans un graphe G, la conclusion doit être absente de G* (sinon la contrainte n'est pas respectée et G n'est pas valide dans le domaine) ; (3) les **règles** de Graphes Conceptuels (règle GC), avec un graphe hypothèse et un graphe conclusion dont la sémantique est *si l'hypothèse est présente dans un graphe G, la conclusion peut être ajoutée à G.*

Une règle GC peut être utilisée implicitement par le système (c'est-à-dire appliquée partout où apparaît l'hypothèse de la règle) ou explicitement par l'utilisateur (sur un élément donné de la base considérée). Une contrainte (positive ou négative) peut être utilisée automatiquement par le système (c'est-à-dire vérifiée partout dans la base considérée) ou vérifiée explicitement à la demande de l'utilisateur.

Une fois l'ontologie construite par spécification du vocabulaire conceptuel et des axiomes exprimant la sémantique de manipulation des types de concepts et de relations, l'opérationalisation permet de générer un ensemble de règles et de contraintes (implicites ou explicites) pouvant être ensuite utilisées par le moteur d'inférences.

La génération des règles et contraintes est guidée par le choix d'un scénario d'usage, l'utilisateur choisissant pour chaque axiome un contexte d'usage parmi ceux décrits en section 3. Les formes des axiomes qui ont été prises en compte sont celles rencontrées lors de l'opérationalisation de l'ontologie de la géométrie. De manière générale, les axiomes de la géométrie, y compris ceux relevant d'un des schémas d'axiomes cités en section 2, ont pour forme ontologique « *si antécédent alors conséquent* » et plus précisément : « *s'il existe des instances* $\{x_i\}_{i=1..p}$ *telles que* $\{T_i(x_i)\}_{i=1..p}$ *et* $\{r_j(x_{n_j}, x_{m_j})\}_{j=1..q}$, *avec* $\{T_i\}_{i=1..p}$ *des types de concepts et* $\{r_j\}_{j=1..q}$ *des types de relations et* n_j *et* m_j *dans* $[1..p]$, *alors il existe des instances* $\{y_k\}_{k=1..r}$, *telles que* $\{T_k(y_k)\}_{k=1..r}$ *et* $\{r_l(z_{u_l}, z_{v_l})\}_{l=1..s}$, *avec* $\{T_k\}_{k=1..r}$ *des types de concepts et* $\{r_l\}_{l=1..s}$ *des types de relations et* z_{u_l} *et* z_{v_l} *dans* $\{x_i\}_{i=1..p} \cup \{y_k\}_{k=1..r}$ ». Les instances x_i et y_k peuvent être des instances indéfinies ou des instances ontologiques.

En fonction des différentes formes dérivées de cette forme générale des axiomes, les représentations opérationnelles des axiomes vont varier. Dans chaque cas, et pour chaque contexte d'usage, l'opérationalisation conduit à générer un ensemble de règles GC et de contraintes correspondant à la sémantique opérationnelle de l'axiome dans le contexte d'usage choisi.

Considérons le cas des axiomes dont le conséquent contient uniquement des relations et ont pour forme générale *s'il existe des instances* $\{x_i\}_{i=1..p}$ *telles que* $\{T_i(x_i)\}_{i=1..p}$ *et* $\{r_j(x_{n_j}, x_{m_j})\}_{j=1..q}$, *avec* $\{T_i\}_{i=1..p}$ *des types de concepts et* $\{r_j\}_{j=1..q}$ *des types de relations et* n_j *et* m_j *dans* $[1..p]$, *alors* $\{r_l(z_{u_l}, z_{v_l})\}_{l=1..s}$,

avec $\{r_l\}_{l=1..s}$ *des types de relations et* z_{u_l} *et* z_{v_l} *dans* $\{x_i\}_{i=1..p}$ [2]. Par exemple, l'axiome 2.1.2 de l'axiomatique hilbertienne de la géométrie, « *Si un point B est entre un point A et un point C, B est aussi entre C et A* », a pour représentation ontologique : *s'il existe des instances* x_1 *et* x_2 *et* x_3 *telles que* $Point(x_1)$ *et* $Point(x_2)$ *et* $Point(x_3)$ *et* $entre(x_2, x_1, x_3)$, *alors* $entre(x_2, x_3, x_1)$.

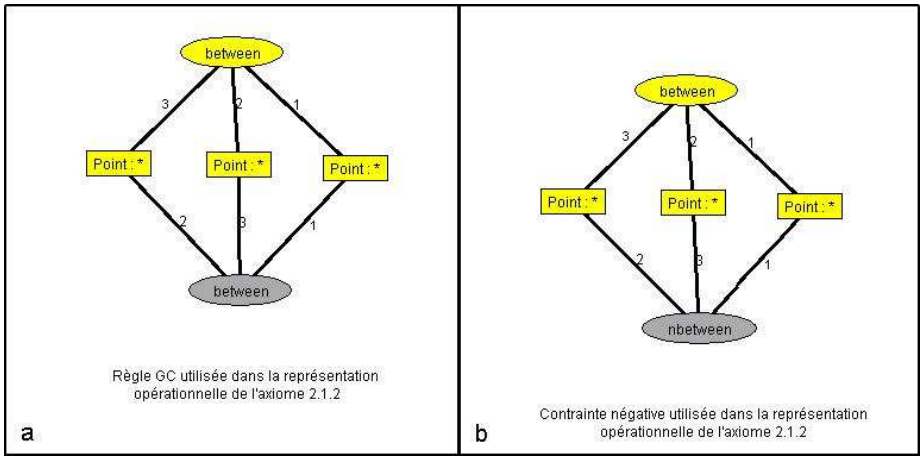

FIG. 3 – *Représentation opérationnelle de l'axiome 2.1.2. La règle (a) permet, si l'hypothèse est présente, d'inférer l'existence d'une relation d'appartenance. La contrainte (b) interdit, sous la même condition, l'existence d'une relation de non appartenance (les parties hypothèse de la règle et de la contrainte sont en clair, les parties conclusion en sombre).*

Dans le cas d'un **contexte inférentiel implicite**, un tel axiome est opérationalisé sous la forme d'une unique règle GC implicite permettant d'inférer les relations du conséquent de l'axiome si son antécédent est rencontré. L'axiome 2.1.2 sera ainsi opérationalisé en contexte inférentiel implicite par la règle GC présentée figure 3a. Dans le cas d'un **contexte inférentiel explicite**, l'objectif est de laisser l'utilisateur libre d'utiliser ou pas l'axiome pour générer de nouveaux faits, tout en contrôlant qu'il n'ajoute pas de fait incompatible avec l'axiome. La forme opérationnelle de l'axiome consiste donc en une règle GC explicite identique à la précédente plus s contraintes négatives interdisant que, pour chacune des relations $\{r_l\}_{l=1..s}$, si l'antécédent de l'axiome est rencontré, on puisse avoir en même temps la relation exclusive de r_l. Si aucune relation exclusive de r_l n'existe dans l'ontologie, et si p relations incompatibles avec r_l existent, la contrainte portant sur la relation r_l est représentée par p contraintes négatives interdisant que, si l'antécédent de l'axiome est présent, on puisse avoir en même temps une des relations incompatibles avec r_l. Par exemple, l'axiome 2.1.2 est

2. Notons que dans cet article, nous ne proposons qu'un exemple de règle d'opérationalisation de façon à apporter une compréhension intuitive au lecteur qui, pour plus de détails, se reportera à (Fürst, 2003).

opérationalisé en contexte inférentiel explicite par la règle GC de la figure 3a plus la contrainte de la figure 3b. Dans le cas d'un **contexte de validation**, seules les contraintes sont prises en compte. L'axiome 2.1.2 sera opérationalisé dans ce contexte par la contrainte de la figure 3b.

5 Opérationalisation et validation d'ontologie

L'opérationalisation permet d'obtenir une forme opérationnelle d'une ontologie adaptée à un type d'application donné. Ce mécanisme est donc très utile pour mener sur l'ontologie des opérations de validation, opérations qui consistent à tester sa fidélité à la sémantique du domaine de connaissances considéré (Gomez-Perez *et al.*, 1998). La validation permet de tester la **complétude de l'ontologie** et la **cohérence de l'ontologie** par rapport au domaine. Cette cohérence est relative au domaine car on suppose que le domaine de connaissances est lui-même cohérent. Un défaut de cohérence de l'ontologie trahit donc une modélisation non fidèle au domaine. La validation d'une ontologie ne peut être menée qu'à travers son utilisation opérationnelle puisque tester formellement la sémantique d'une ontologie (au sens de tester la forme de l'ontologie) suppose de pouvoir se référer à un modèle formel pré-existant des connaissances du domaine. Or bâtir une ontologie a justement pour but de créer ce modèle formel, à partir d'un corpus (le plus souvent informel) de connaissances. On ne peut donc valider une ontologie à travers sa forme mais uniquement à travers son utilisation opérationnelle.

La validation s'appuie de manière consensuelle sur des **questions de compétence** (Gruninger & Fox, 1995) auxquelles l'ontologie doit permettre de répondre. L'impossibilité de répondre à une telle question en utilisant les connaissances exprimées dans l'ontologie implique que certaines connaissances du domaine indispensables à la résolution de la question manquent. La validation des ontologies repose ainsi sur des spécifications externes à l'ontologie (questions de compétences) et des spécifications internes constituées par les axiomes décrivant la sémantique du domaine.

Tester la complétude de l'ontologie par rapport au domaine à l'aide de questions de compétences peut être automatisé par les mécanismes d'opérationalisation proposés : une fois les questions de compétences formalisées dans le langage opérationnel choisi, l'ontologie est opérationalisée suivant un scénario d'usage inférentiel, puis le moteur d'inférence est utilisé sur le fait initial de chaque question de compétence pour produire la conclusion. L'impossibilité de produire la conclusion traduit l'incomplétude de l'ontologie.

Ainsi, dans le cas de l'ontologie de la géométrie, les théorèmes de base du domaine constituent un ensemble de questions de compétences : l'impossibilité de démontrer automatiquement ces théorèmes assure de l'incomplétude de l'ontologie. Dans le cadre de nos travaux, nous avons testé l'ontologie de la géométrie dans un contexte de démonstration automatique de théorèmes (Leclere *et al.*, 2002). L'impossibilité de démontrer certains théorèmes nous a conduit à identifier des connaissances utilisées dans le domaine mais non explicitées (par exemple,

la symétrie de la relation *entre(Point,Point,Point)*) (Fürst *et al.*, 2002).

L'opérationalisation d'une ontologie permet également de tester sa cohérence. L'opérationalisation d'un axiome dans un contexte de validation permet de générer les contraintes induites par l'axiome. Ces contraintes peuvent ensuite être utilisées pour tester si, pour chacun des autres axiomes, l'antécédent ou le conséquent de l'axiome viole une des contraintes. Dans les deux cas, cela signifie qu'un des deux axiomes (celui qui est opérationalisé dans un contexte de validation ou celui qui est testé) ne correspond pas à la sémantique du domaine, ce qui constitue un défaut de modélisation.

6 Conclusion

La plupart des outils d'édition d'ontologie permettent à l'utilisateur de spécifier des axiomes au niveau opérationnel, et non conceptuel, et ces outils ne permettent pas d'opérationaliser les ontologies pour différents types d'usage. Par exemple, dans OntoEdit, la spécification des axiomes ne correspondant pas à un type prédéfini utilise la syntaxe de la F-Logic et fixe ainsi la sémantique opérationnelle des axiomes, qui sont tous utilisés dans le moteur d'inférence comme des règles. Dans Protégé, le langage PAL permet de définir des contraintes là aussi sous forme de règles, qui sont utilisées dans le moteur d'inférence uniquement pour tester la cohérence de l'ontologie.

L'aspect le plus innovant de notre approche est d'autoriser la spécification des axiomes au niveau conceptuel, dans une syntaxe graphique qui n'en contraint pas l'usage opérationnel. L'opérationalisation de l'ontologie dans un langage opérationnel donné et pour différents scénarios opérationnels précis est alors possible, et même automatisable, et conduit à la génération d'ontologies opérationnelles adaptées aux scénarios. Ainsi, la réutilisabilité des ontologies est renforcée, et leur mise en oeuvre facilitée, en particulier à des fins de validation des ontologies.

La méthode proposée demande à être étendue à d'autres langages opérationnels de représentation d'ontologie, en particulier les langages du Web sémantique, dont le développement nécessite de pouvoir disposer d'ontologies opérationnelles permettant le raisonnement et susceptibles d'être échangées sur le Web. La définition de règles d'opérationalisation pour une extension du OWL (Ontology Web Language) à la représentation de règles et de contraintes, rendra ainsi possible la production d'ontologies opérationnelles dans une syntaxe XML standardisée pour les différents contextes applicatifs du Web sémantique.

Références

Bachimont B. (2000). *Engagement sémantique et engagement ontologique : conception et réalisation d'ontologies en ingénierie des connaissances*, In J. Charlet, M. Zacklad, G. Kassel & D. Bourigault, Eds., *Ingénierie des connaissances : évolutions récentes et nouveaux défis*, p. 305–323. Eyrolles.

Baget J. & Mugnier M. (2002). Extensions of simple conceptual graphs: the complexity of rules and constraints. *Journal of Artificial Intelligence Research*, **16**, 425–465.

Charlet J. (2003). L'ingénierie des connaissances : développements, résultats et perspectives pour la gestion des connaissances médicales. *Mémoire d'Habilitation à diriger des recherches, Université Pierre et Marie Curie.*

J. Charlet, P. Laublet & C. Reynaud, Eds. (2003). *Rapport Web sémantique de l'Action Spécifique 32 CNRS/STIC.*

Chen P.-P. (1976). The Entity-relationship Model - Toward a unified view of data. *ACM Trans. Database Syst.*, **1**(1), 9–36.

Fürst F. (2003). TooCoM: a Tool to Operationalize an Ontology with the Conceptual Graph Model. In *Proceedings of the second Workshop on Evaluation of Ontology-Based Tools (EON'2003) at the International Semantic Web Conference (ISWC'2003)*, p. 57–70.

Fürst F., Leclère M. & Trichet F. (2002). Construction d'une ontologie opérationnelle : un retour d'expérience. *Extraction des Connaissances et Apprentissage (actes de la conférence Extraction et Gestion de Connaissances EGC'2002), Hermès*, **1**(4), 227–232.

Fürst F., Leclère M. & Trichet F. (2003). Ontology engineering and mathematical knowledge management: a formalization of projective geometry. *Numéro spécial de la revue internationale Annals of Mathematics and Artificial Intelligence, Kluwer Academic Publishers ISSN:1012-2443*, **38**(1), 65–89.

Genest D. & Salvat E. (1998). A platform allowing typed nested graphs : how CoGITo became CoGITaNT. In *Proceedings of the International Conference on Conceptual Structures (ICCS'98)*, volume 1453, p. 154–161: Springer-Verlag LNAI.

Gomez-Perez A., Fernandez-Lopez M. & Corcho G. (1998). *Ontological Engineering.* Advanced Information and Knowledge Processing - Springer-Verlag.

Gruber T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, **5**(2), 199–220.

Gruninger M. & Fox M. S. (1995). Methodology for the design and evaluation of ontologies. In *Proceedings of the Workshop on Basic Ontological Issues on Knowledge Sharing, IJCAI'95.*

Leclere M., Trichet F. & Fürst F. (2002). Operationalising domain ontologies: towards an ontological level for the sg family. In *Fondations and Applications of Conceptual Structure, contributions to the International Conference on Conceptual Structures (ICCS'02)*: Bulgarian Academy of Sciences.

Mugnier M. & Chein M. (1996). Représenter des connaissances et raisonner avec des graphes. *Revue d'Intelligence Artificielle*, **10**, 7–56.

OWL (2002). Ontology web language guide. In *http://www.w3.org/TR/2002/WD-owl-guide-20021104/.*

Sowa J. (1984). *Conceptual Structures : information processing in mind and machine.* Addison-Wesley.

Staab S. & Maedche A. (2000). *Axioms are objects too: Ontology Engineering beyong the modeling of concepts and relations.* Research report 399, Institute AIFB, Karlsruhe.