

Une nouvelle méthode graphique pour interroger et vérifier des diagrammes de classes UML

Thomas Raimbault

LERIA, Université d'Angers, 2 boulevard Lavoisier 49045 ANGERS Cedex 01
thomas.raimbault@info.univ-angers.fr

Résumé. UML est le langage graphique de référence dans l'industrie pour la modélisation objet. Cependant UML reste un langage, et ne fournit aucun moyen de vérification ou d'interrogation de ses schémas. Il existe aujourd'hui des outils de vérification, mais ils se comportent comme des boîtes noires où l'utilisateur ne peut accéder. Nous proposons une méthode graphique de vérification et d'interrogation de diagrammes de classes UML. L'aspect intuitif et dessinable de notre méthode offre à l'utilisateur la possibilité d'interroger le contenu de diagrammes de classes, ainsi que de définir et d'adapter ses propres critères de vérification. Le modèle calculatoire de notre approche est celui des graphes conceptuels.

1 Introduction

UML, Unified Modeling Language (Booch et al. 1998), est le langage graphique de référence dans l'industrie pour la modélisation objet. Cependant UML reste un langage, et ne fournit aucun moyen de vérification ou d'interrogation de ses schémas. Il existe des outils commerciaux de vérification, tels que Rational Software Rose (IBM 2004) ou Borland Together (Borland 2004). Mais les vérifications proposées sont uniquement *standards*, vérifiant la cohérence des diagrammes par rapport aux spécifications objet. De plus, la méthode de vérification est dans une boîte noire : les traitements sont de bas niveau et non accessibles à l'utilisateur. Enfin, l'interrogation de diagrammes n'est pas totalement libre mais limitée à un cadre pré-formaté de questions.

Pour répondre aux exigences de qualité et d'interaction en modélisation, nous proposons une méthode graphique d'interrogation et de vérification de diagrammes de classes UML. L'aspect intuitif et dessinable de cette méthode offre à l'utilisateur la possibilité de définir et d'adapter ses propres critères de vérification, ainsi que d'interroger librement le contenu de diagrammes de classes UML. Le travail présenté dans cet article est issu de (Raimbault 2004), et est traité pour l'atelier EGC 2005 "*Modélisation des connaissances*" de façon intuitive au travers un exemple. Concrètement, notre méthode utilise pour les calculs le modèle des graphes conceptuels (Sowa 1984).

Cet article est structuré comme suit : la section 2 traite de notre méthode graphique d'interrogation et de vérification de diagrammes de classes UML. En section 3, nous abordons l'aspect calculatoire de notre méthode qui utilise la modèle des graphes conceptuels. La section 4 discute des résultats et des perspectives de notre méthode.

2 Interroger et Vérifier un diagramme de classes

Nous indiquons d'une part comment formuler une *requête* pour interroger le contenu d'un diagramme de classes UML, d'autre part, comment définir les *critères de validité*

Interroger et vérifier graphiquement des diagrammes de classes UML

de diagrammes de classes. Ces vérifications peuvent être des standards que l'on retrouve en modélisation objets ou des vérifications dites de métier qui sont spécifiques aux besoins de l'utilisateur. L'aspect dessinable et intuitif des requêtes ou des critères de validité les rend accessibles et manipulables par l'utilisateur. De plus, contrairement à des outils commerciaux, les vérifications des diagrammes de classes ne se comportent pas comme des boîtes noires.

2.1 Interrogation

Pour interroger un diagramme de classes, il suffit de formuler la requête sous forme d'un diagramme de classes. La réponse à une requête, si elle existe, est une sous partie du diagramme de classes à interroger dans laquelle le diagramme requête se projette. Supposons que nous voulons interroger le diagramme de classes Figure 1 par la requête suivante : “la classe ‘4x4’ a-t-elle une classe mère?”. Cette requête est formulée par le diagramme requête présenté à gauche de la Figure 2 où la classe ‘4x4’ a pour généralisation une classe générique, notée *. Cette requête admet une réponse, via une projection du diagramme requête sur le diagramme Figure 1, qui est : “Oui, la classe ‘4x4’ possède bien une classe mère ; de plus, cette classe mère est la classe ‘Voiture’”.

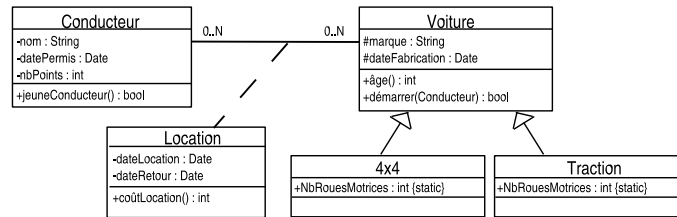


FIG. 1 – Exemple de diagramme de classes UML.

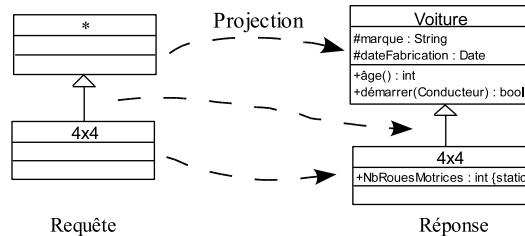


FIG. 2 – Exemple de diagramme requête et sa réponse.

Des requêtes plus riches peuvent être modélisées. Par exemple, “obtenir les attributs statiques des sous-classes d’une classe donnée” ou bien “quelles sont les classes publiques associées avec une multiplicité de 1 à une classe donnée”.

2.2 Vérification

Nous disposons de deux types de critères de validité de diagrammes de classes UML. Un critère positif est de la forme “si *condition*, il faut que *obligation*”, et un critère négatif de la forme “si *condition*, il ne faut pas *interdiction*”.

Un critère de validité, positif ou négatif, est modélisé sous la forme d’un diagramme de classes bicolore où la condition est sur fond blanc et l’obligation ou l’interdiction sur

fond noir. Un critère de validité positif est repéré par le symbole $\boxed{+}$, et un négatif par $\boxed{-}$. La Figure 3 présente trois critères de validité positifs, C_1 , C_2 et C_3 , et deux critères de validité négatifs, C_4 et C_5 . Le critère C_1 a pour condition une classe, notée * car non spécifiée, qui possède une opération abstraite non spécifiée elle aussi. L'obligation de ce critère est que la classe doit être abstraite. En d'autres termes, C_1 exprime le fait que "Si une classe possède une opération abstraite, alors cette classe doit être déclarée abstraite". C_2 stipule que "Si une classe A possède des opérations abstraites, alors une sous-classe B de A ne peut être non abstraite qu'à la condition que les opérations abstraites de A soient redéfinies comme non abstraites dans B ". C_3 indique que "toute classe doit être associée à une autre classe". C_4 interdit tout cycle d'héritage. La condition de C_5 est une classe possédant une opération finale notée $\$x$, et l'interdiction est que cette même opération $\$x$ soit présente dans une sous-classe de cette classe. Donc, "une opération finale, d'une classe A , ne peut être redéfinie dans une classe B qui est sous-classe de A ".

Remarquons d'une part que pour les critères de validité présentés Figure 3, le lien de généralisation est considéré comme transitif. D'autre part, ces mêmes critères de validité, à l'exception de C_3 , sont des critères *standard* de validité en modélisation objet. C_3 est un exemple de critère de validité qui peut être spécifique aux besoins d'un utilisateur, nous le nommons *critère métier*.

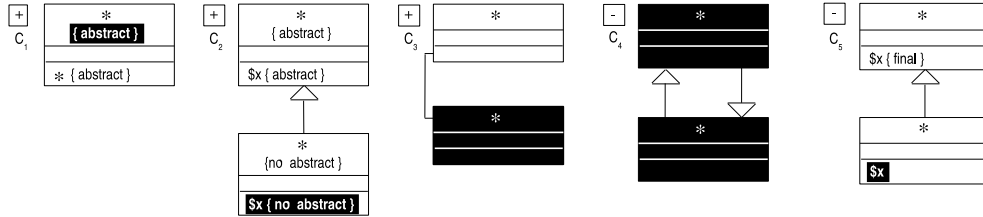


FIG. 3 – Exemple de critères de validité positifs et négatifs.

3 Graphes conceptuels, Règles et Contraintes

Les *graphes conceptuels* (GCs) constituent un modèle formel de représentation des connaissances, de la famille des réseaux sémantiques (Lehmann 1992). (Sowa 1984) introduit un modèle simple des GCs muni d'une sémantique logique du premier ordre. Les GCs permettent une interprétation du dessin le représentant en termes de relations entre des concepts.

Dans la pratique, une connaissance donnée est modélisée par un GC. Il s'agit d'un graphe biparti étiqueté, où les étiquettes d'une catégorie de sommets correspondent à des noms de *concepts* et celles de l'autre catégorie à des noms de *relations*. Ce GC est défini sur un *support* qui spécifie le vocabulaire de base pour représenter les connaissances sur le domaine modélisé. Chaque élément du modèle des GCs est interprétable en logique du premier ordre par un opérateur qui traduit un GC en une formule logique.

Notre méthode graphique d'interrogation et de vérification de diagrammes de classes UML est développée en utilisant le modèle des GCs (Mugnier et Chein 1996) emboîtés typés (Chein et Mugnier 1997), avec liens de corréférence (Chein et Mugnier 2004), règles

(Salvat 1998) et contraintes (Baget et Mugnier 2002). L'idée clé de cette implémentation est de modéliser un diagramme de classes UML dans le formalisme des GCs, où d'un côté le support définit les notations UML et d'un autre côté l'agencement des sommets entre eux d'un GC ainsi que les marqueurs individuels des sommets concepts définissent un diagramme de classes donné. Ensuite, nous utilisons les possibilités de raisonnement logique qu'offre le modèle des GCs avec l'utilisation d'opérateurs graphiques. L'application de règles permet de faire ressortir de façon explicite des informations implicites, comme par exemple la transitivité du lien de généralisation entre classes. Les critères de validité positifs et négatifs sont respectivement modélisés par des contraintes positives et négatives. La vérification de ces contraintes détermine si un diagramme de classes est valide ou non selon les critères choisis par l'utilisateur.

La Figure 4 représente la modélisation, en un GC, des classes 'Conducteur' et 'Voiture' de la Figure 1 et leur association. La classe 'Voiture' par exemple est modélisée par un sommet concept de type *class* (type défini dans le support) qui a pour marqueur individuel *Voiture*. Cette classe possède des attributs et des opérations qui eux aussi sont modélisés par des sommets concepts. Ces derniers décrivent la classe 'Voiture' et sont donc emboîtés au sein du sommet concept qui la représente. Tout comme la classe 'Voiture', ses attributs et opérations possèdent leurs propres descriptions qui sont respectivement emboîtés dans les sommets concepts les représentant. Ces emboîtements ne sont pas visibles sur la Figure 4 mais sous-entendus avec les caractères '...'. L'association entre les classes 'Conducteur' et 'Voiture' est centralisée autour de la classe d'association 'Location' par l'intermédiaire de sommets relations de type *association*. Remarquons que les emboîtements d'un sommet concept constituent des niveaux de lecture plus internes de l'information le représentant. L'utilisateur peut ainsi plus ou moins zoomer sur la description d'une partie du diagramme de classes.

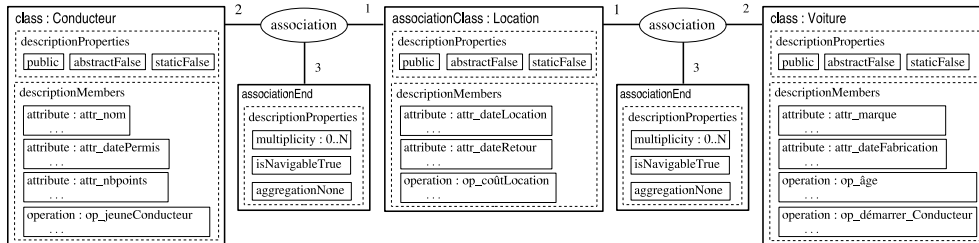


FIG. 4 – Modélisation en un GC des classes 'Conducteur', 'Voiture' et leur association.

4 Résultats et perspectives

Un prototype a été développé pour d'une part réécrire un diagramme de classes UML au format XMI (OMG 2002) dans le formalisme des GCs emboîtés typés au format BCGCT (Haemmerlé O. 1995). D'autre part, règles et contraintes ont été écrites au format BCGCT pour vérifier des diagrammes de classes et satisfaire au concept objet. La vérification tout comme l'interrogation d'un diagramme de classes, sous forme d'un GC, a été testé sur la plateforme CoGITaNT (Genest 2004). Les résultats obtenus sur des exemples simples de diagrammes de classes sont tout à fait satisfaisants. Ils

fournissent les exigences souhaitées et ce de façon quasi instantanée.

Nous souhaitons étendre notre méthode graphique de modélisation, d'interrogation et de vérification aux différents diagrammes UML. Cette extension offrira ainsi un formalisme unique et visuel pour modéliser tous les diagrammes UML ainsi que des requêtes ou des vérifications à effectuer sur et entre ces derniers. Actuellement, une interface est en cours de développement pour permettre à l'utilisateur de formuler ses requêtes et vérifications dans un formalisme UML (cf. Figure 3). Ainsi, le modèle des GCs peut être transparent pour l'utilisateur qui le désire.

Références

- Baget J.F. et Mugnier M.-L. (2002), Extensions of Simple Conceptual Graph : the complexity of Rules and Constraints. JAIR, vol. 16, pp. 425-465, 2002.
- Booch G., Jacobson C. et Rumbaugh J. (1998), The Unified Modeling Language - a reference manual. Addison Wesley.
- Borland (2004), Together, <http://www.borland.com/together/>.
- Chein M. et Mugnier M.-L. (1997), Positive Nested Conceptual Graphs. In Proc. of ICCS'97, vol. 1257 of LNAI, pages 95-109, Springer Verlag.
- Chein M. et Mugnier M.-L. (2004), Concept types and coreference in simple conceptual graphs. In Proc. of ICCS'2004, vol. 3127 of LNAI, pages 303-318, Springer Verlag.
- Genest D. (2004), CoGITaNT 5.1.5, LIRMM-LERIA, <http://cogitant.sourceforge.net>
- Haemmerlé O. (1995), Plate-forme CoGITo, Rapport technique 95012, LIRMM.
- IBM (2004), Rational Software Rose, <http://www-306.ibm.com/software/rational/>
- Lehmann F. (1992), Semantic Networks in Artificial Intelligence, Pergamon Press.
- Mugnier M.-L. et Chein M. (1996), Représenter des connaissances et raisonner avec des graphes, vol.10 dans RIA, n°1, 7-56.
- OMG (2002), Object Management Group Documentation, XML Metadata Interchange (XMI), <http://www.omg.org/technology/documents/formal/xmi.htm>
- Rimbault T. (2004), Une nouvelle méthode graphique pour vérifier et interroger des diagrammes de classes UML, Mémoire de DEA, LERIA.
- Salvat E. (1998), Theorem Proving Using Graph Operations in the Conceptual Graph Formalism. In Proc. of ECAI'98.
- Sowa J.F. (1984), Conceptual Structures - Information Processing in Mind and Machine, Addison-Wesley.

Summary

UML is an industrial reference graphic modeling language to express object systems. However UML is just a language, it does not provide any means to check or interrogate diagrams. There are commercial tools today, but they behave as black boxes where the user cannot accede into. We propose a graphic method to check and interrogate UML class diagrams. The intuitive and drawing aspect of our method make it possible to the user to interrogate contents of UML class diagrams, and to define and adapt its own criteria of checks. The calculative model of our approach is that of conceptual graphs.

Transformation des concepts du diagramme de classe UML en OWL full

Macaire Ahlonsou, Emmanuel Blanchard
Henri Briand, Fabrice Guillet
2bis boulevard Léon Bureau BP96228 de l'Université de Nantes
<http://www.sciences.univ-nantes.fr/lina/fr/>
LINA – Université de Nantes
Ecole polytechnique de l'université de Nantes
La Chantrerie, rue Christian Pauc 44306 Nantes CEDEX 3

{Prenom.Nom}@polytech.univ-nantes.fr

Résumé. Le web peut être considéré comme une grande base de connaissances. La recherche des informations pertinentes sur la toile est rendue de plus en plus difficile, voire impossible avec l'accroissement de la volumétrie des pages disponibles. Le problème réside dans le fait que les outils existants ne peuvent pas s'appuyer actuellement sur une description du contenu des documents. Le web sémantique utilise différents langages pour mieux exploiter et traiter les contenus des ressources web. Dans le but de passer de UML vers OWL, il est intéressant d'étudier la possibilité de transformer chacun des concepts du diagramme de classe UML en OWL.

1 Introduction

UML (Unified Modeling Language) (Gaertner et al. 2002) est un langage de modélisation orienté objet clairement adopté dans le monde industriel. Le web sémantique est une vision du futur web dans lequel l'information serait explicitée de manière à permettre son traitement automatique par des machines. Nombreuses sont les applications qui sont déjà modélisées en UML. Il n'existe pas de langage de modélisation spécifique pour modéliser une base de connaissance. On peut étendre l'utilisation de UML, notamment les diagrammes de classe à cette fin (Walter 1998).

L'objectif de cet article est d'étudier la possibilité de transformer les concepts du diagramme de classe UML (langage semi formel) en OWL (langage formel). Cette transformation est effectuée dans un seul sens (UML vers OWL) et reste dans un monde clos. Nous proposons de transformer trois concepts du diagramme de classe après avoir situé les divers langages du web sémantique les uns par rapport aux autres, en mettant en avant leurs limites.

2 Langages et définitions

2.1 UML

UML est un langage de modélisation plébiscité dans le domaine de la conception. Il permet de représenter les composants statiques et dynamiques des systèmes dépendant des logiciels à travers des modèles représentés par des vues. Ces vues sont manipulées à travers des diagrammes. Cet article ne traitera que le cas des diagrammes de classe.

2.2 Web sémantique

Le web est une notion qui a beaucoup évolué dans le temps. Aujourd'hui, on tend vers le web sémantique qui permet de donner du sens aux ressources utilisées sur le web et d'aider plus efficacement les utilisateurs dans leur recherche (Berners-Lee et al. 2001). Il est basé sur le concept d'Ontologie. «An Ontology is a formal, explicit specification of a shared conceptualizations» (Gruber 1993). Les ontologies sont considérées comme des spécifications partielles et formelles d'une conceptualisation (Guarino et al. 1995).

Quatre langages sont issus du web sémantique pour la formalisation des ontologies :

RDF: La spécification RDF utilise le terme de déclaration (statement) afin de décrire un triplet. Formellement, elle s'exprime de la façon suivante : le sujet possède un prédicat d'objet. En effet, un sujet est une ressource (représentée par un ovale) et est identifié de façon unique par une URI (unique resource identifier). Une URI est un espace de nommage et peut être une URL (unique resource location) ou un identifiant (ID).

RDFS ajoute à RDF la possibilité de définir des hiérarchies de classes et de propriétés dont l'applicabilité et le domaine de valeurs peuvent être contraintes à l'aide des attributs *rdfs:domain* et *rdfs:range*. A chaque domaine applicatif peut être ainsi associé un schéma identifié par un préfixe particulier et correspondant à une URI.

DAML+OIL : RDFS ne décrit pas avec précision la sémantique, ne possède pas de modèle d'inférence ; la cardinalité est limitée à 0 ou 1. DAML+OIL permet de corriger ces lacunes.

OWL (Bechhofer et al. 2004) : DAML+OIL est incapable de vérifier la consistance des ontologies et n'utilise pas une sémantique formelle. OWL corrige les faiblesses de DAML+OIL. Il est décomposé en OWL lite, OWL DL, et OWL full (Hart et al. 2004).

OWL Lite est un sous langage de OWL DL et définit les concepts simples de OWL. La cardinalité est limitée à 0 ou 1.

OWL DL est un sous langage de OWL Full et se base sur la logique descriptive. Il définit les concepts plus complexes comme le complément, l'intersection, l'union, etc.

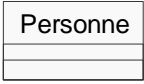
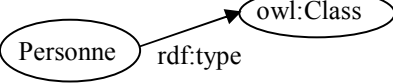
OWL Full autorise le mélange arbitraire avec le schéma RDF. Il est plus complet que OWL Lite et OWL DL et sera utilisé pour la transformation des concepts du diagramme de classe UML.

3 UML vers OWL

Seuls trois concepts du diagramme de classe UML seront traités dans cet article. Les autres concepts et diagrammes sont traités dans un autre document (Macaire et al. 2005).

3.1 Le Concept de classe

Exemple : Donnons le code OWL et le graphe correspondant à la représentation d'une classe de type Personne.

UML	Graphe OWL	Code OWL
		<pre><owl:Class rdf:ID= "Personne"/></pre>

TAB 1 – tableau récapitulatif de la représentation d'une classe en UML et OWL.

Ce tableau exprime la notion de classe en UML, le graphe et le code correspondant à cette classe en OWL. Il est important de noter que les propriétés des relations (transitivité, inverse...) exprimées sur les classes en UML sont aussi exprimables sur les classes OWL.

3.2 Le Concept de propriété

En OWL on a deux types de propriétés : les propriétés d'objet (`owl:ObjectProperty`) ou les associations entre classes et les propriétés d'attribut (`owl:DatatypeProperty`) ou les attributs d'une classe. On peut exprimer la réciproque d'une propriété. La propriété d'objet et sa réciproque sont équivalentes aux rôles en UML ; de même, on peut exprimer des cardinalités sous forme de contraintes de cardinalité sur les propriétés d'objets en OWL.

Exemple : « un auteur connu, crée au moins une œuvre ». Dans cet exemple, les deux classes définies sont : la classe Auteur et la classe Œuvre. La propriété d'objet « crée » a pour domaine la classe «Auteur» et pour Image la classe «Œuvre». La propriété réciproque de «crée» est «est_créée_par». La propriété d'attribut de la classe Auteur est « nom ».

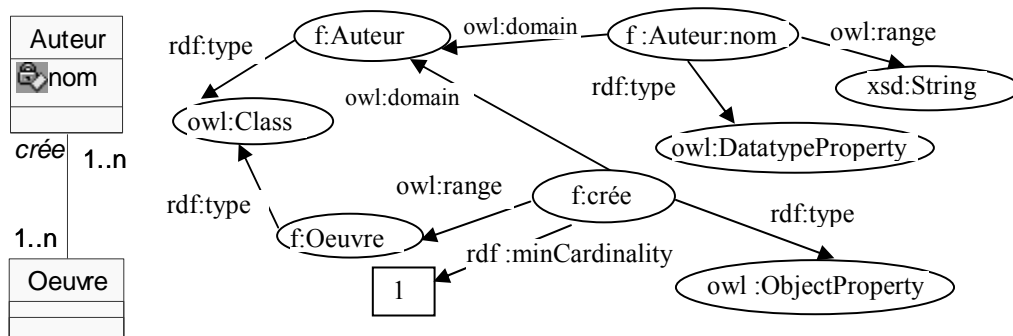


FIG. 1 - Diagramme de Classe UML & Graphe OWL correspondant à l'exemple de propriété OWL.

On pourrait donner le code OWL correspondant au graphique. Notons que ce code peut être obtenu par saisie de cet exemple dans Protégé2000.

3.3 Le Concept de composition

En UML, la composition est une agrégation (association transitive) particulière avec une forte appartenance et dépendance entre les cycles de vie du composé et de ses

Transformation des concepts du diagramme de classe UML en OWL full

composants. La multiplicité du côté du composé est toujours de 1 (Gaertner et al. 2002). La disparition du composé entraîne celle des composants.

En OWL, la composition est une association transitive avec la notion de container matérialisé par « Bag » (ressources non triées) ou « Seq » (ressources triées) et de cardinalité « un » du côté du composé. Les containers « Bag et Seq » sont des collections et permettent de signifier que la disparition du composé entraîne celle du composant. La composition peut être mappée en OWL par des concepts équivalents.

Exemple. Graphe correspondant au diagramme de classe UML de l'exemple : «Un livre de titre connu, paru à une date donnée, contient au moins une page».

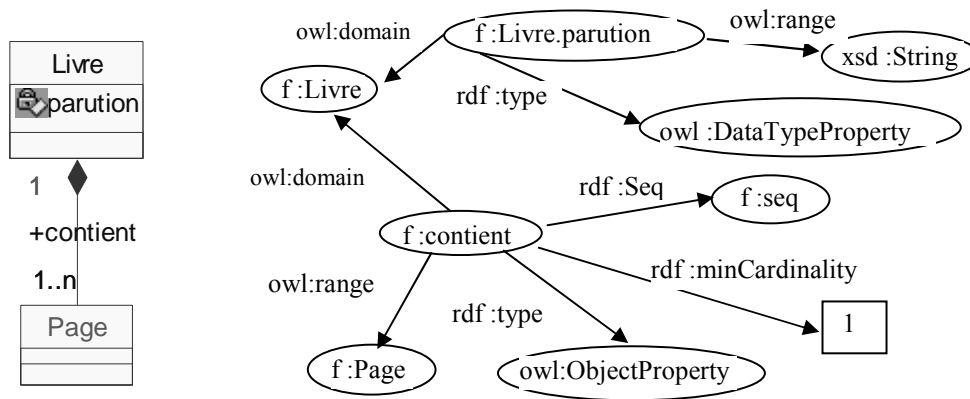


FIG. 2 - Diagramme de Classe UML & Graphe OWL correspondant à l'exemple de la composition.

Une partie du code OWL correspondant à la composition:

```
<owl:ObjectProperty rdf:ID="contient">
  <rdf:type rdf:resource="#owl:TransitiveProperty" />
  <rdfs:domain rdf:resource="#Livre"/>
  <rdfs:range rdf:resource="#Page"/>
  <rdf:Seq ID="#Livre">
    <rdf:li resource="#Page" />
  </rdf:Seq>
</owl:ObjectProperty>
```

4 Conclusion

Dans cet article, nous avons étudié la possibilité de transformer les concepts d'un diagramme de classe UML en OWL. Nous proposons par la suite de transformer tous les autres concepts du diagramme de classe UML en OWL, d'étudier la solution qui consiste à utiliser les méta - modèles. Le passage de UML à OWL permet de développer des applications de type web sémantique et des systèmes à base de connaissances. Après transformation de chacun des concepts du diagramme de classe, on va pouvoir les utiliser

pour saisir dans un outil de gestion de base de connaissances (Protégé par exemple) les modèles de classe. Nous avons implémenté un exemple dans Protégé (Protégé 2005).

Références

- Ahlonsou M. , Blanchard E., Briand H. , Guillet F. (2005), Transformation des concepts du diagramme de classe UML en OWL FULL, EGC 2005, workshop Modelling of Knowledge. January 2005: Paris.
- Protégé2000 (2005), Stanford, California, <http://protege.stanford.edu/>, (2005).
- Fürst F. (2004), Ingénierie des Ontologies : une méthode et outil d'opérationnalisation. Thèse université de Nantes, 2004.
- Bechhofer S., Van Harmelen F., Horrocks J., Deborah L., Mc Guinness, Peter F., Patel-Schneider, Stein L. (2004), OWL web ontology Language. <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>, (2004).
- OMG (2003), Unified Modeling Language Specification v1.5, OMG Document format/03-03-01, <http://www.omg.org/cgi-bin/apps/doc?formal/0303-01.zip>, March 2003.
- Gasevic D., Damjanovic V., Vladan D. (2003), Analysis of the MDA Standards in Ontological Engineering, in: Proceedings of the 6th International Conference of Information Technology, Bhubaneswar, India, 2003, 193-196.
- Gaertner N., Muller P. (2002) Modélisation objet avec UML. Eyrolles, 2002.
- Berners-Lee T., Hendler J. and Lassila O. (2001), The semantic web. Scientific American 284 (2001) 35-43.
- Walter W. (1998), a discussion of the relationship between RDFS and UML. <http://www.w3.org/TR/1998/note-rdf-uml-19980804>, 04-Aug-1998.
- Guarino N. et Giarretta P. (1995), Ontology and knowledge base, towards a terminological clarification, page 25-32. IOS Press 1995.
- Gruber T. (1993), Translation Approach to Portable Ontology Specifications. Knowledge acquisition, 5(2): 199-220, 1993.

Summary

The translation of UML to OWL facilitates the development of semantic web represented by the knowledge base. After the transformation of each concept of UML class diagram, one will be able to use them to put the models of class diagram in management knowledge tool (Protégé for example). In protégé, we can improve the model and then translate it in OWL.

Modéliser des connaissances ontologiques dans le cadre du modèle des Graphes Conceptuels

Frédéric Fürst

LINA - FRE 2729

22 rue de la Houssinière, BP 92208, 44322 Nantes
furst@lina.univ-nantes.fr

Résumé. Cet article présente OCGL (Ontology Conceptual Graph Language), un langage de représentation d'ontologie basé sur le modèle des Graphes Conceptuels. Il décrit en détail la façon dont une ontologie est modélisée en OCGL, et présente l'implémentation de ce langage dans l'atelier d'ingénierie ontologique TooCoM.

1 Introduction

L'ingénierie des ontologies est née de la volonté de diversifier les applications des Systèmes à Base de Connaissances (SBC), et de permettre des représentations de connaissances indépendantes de ces diverses applications (Gomez-Perez et al., 2003). L'intégration d'un tel composant dans un Système à Base de Connaissances suppose alors d'adapter les représentations qu'il intègre à l'objectif opérationnel du système, adaptation qui est l'objet du processus d'opérationnalisation des ontologies (Fürst et al., 2004). D'autre part, les ontologies ont vocation à intégrer toute la sémantique des différents domaines de connaissances, c'est-à-dire des propriétés de base comme la subsumption entre concepts, mais également toute propriété permettant d'exprimer la sémantique du domaine considéré. Les ontologies évoluent ainsi des ontologies légères (lightweight ontologies), n'intégrant qu'un nombre restreint de propriétés, vers des ontologies lourdes (heavyweight ontologies), visant la modélisation de toutes les propriétés nécessaires à la représentation de toute la sémantique d'un domaine (Gomez-Perez et al., 2003).

Dans cet article, nous présentons OCGL (Ontology Conceptual Graph Language), un langage de représentation d'ontologies lourdes, basé sur le modèle des Graphes Conceptuels (GCs) (Sowa, 1984). OCGL est implémenté dans l'outil TooCoM (a Tool to Operationalize an Ontology in the Conceptual Graph Model), dédié à la modélisation et l'opérationnalisation d'ontologies lourdes dans le cadre du modèle des Graphes Conceptuels¹. Nous détaillons ici le modèle de représentation utilisé dans TooCoM, mais ne présentons pas le processus d'opérationnalisation qu'il implémente, renvoyant pour cela le lecteur à (Fürst et al., 2004).

1. Cet outil est disponible sous licence GPL sur le site <http://sourceforge.net/projects/toocom/>

2 OCGL : Ontology Conceptual Graphs Language

Notre objectif consiste à offrir la possibilité de modéliser toutes les connaissances d'un domaine dans le cadre du paradigme Entité/Relation, pour construire des ontologies lourdes dédiées au raisonnement. Le choix du modèle des GCs repose, d'une part, sur l'aspect graphique de ce formalisme, plus intuitif et facilement manipulable par des experts du domaine (qui ne sont a priori pas experts du modèle), et d'autre part sur l'expressivité des GCs, qui permettent de représenter à la fois des connaissances terminologiques et des propriétés de base des ontologies (subsumption, signature des relations, etc), et des propriétés ne correspondant pas à des propriétés classiques, sous forme d'implications. De plus, il existe dans le modèle des GCs des mécanismes de raisonnement qui permettent de mettre en œuvre les représentations pour raisonner.

Ainsi, OCGL reprend en bonne partie le modèle des Graphes Conceptuels, sa syntaxe graphique et sa sémantique formelle, ainsi que ses extensions, notamment la SG-family (Baget and Mugnier, 2002). Il en diffère cependant en ce qu'il est seulement un langage de représentation d'ontologie, et non pas un langage de représentation de connaissances. Ainsi, seules les connaissances ontologiques ont leur place en OCGL, et non les connaissances factuelles portant sur des cas particuliers. De même, seules les instances ontologiques des concepts sont représentées, c'est-à-dire celles qui participent à la définition de la sémantique du domaine.

De plus, les représentations de connaissances dans OCGL ne sont pas opérationnelles, de manière à assurer une réutilisabilité maximum aux ontologies en préservant leur indépendance vis-à-vis des applications. Les représentations du modèle des GCs dotées d'une sémantique opérationnelle (règles et contraintes de graphes conceptuels), c'est-à-dire dont le mode opératoire est fixé, ne sont donc pas reprises dans OCGL. De plus, il est nécessaire d'intégrer aux ontologies exprimées en OCGL des propriétés des primitives conceptuelles (concepts et relations) classiquement utilisées dans les ontologies, et qui ne sont pas explicitement intégrées au modèle des GCs, telles que les propriétés algébriques des relations (Staab and Maedche, 2000). OCGL étend donc une partie du modèle des Graphes Conceptuels, en lui ajoutant un ensemble de propriétés portant sur les types de concepts et les types de relations.

Le langage OCGL comporte un **niveau terminologique**, composé d'un ensemble de primitives conceptuelles (concepts et relations) et d'un ensemble d'instances ontologiques, et un **niveau axiomatique**, où est exprimée la sémantique des primitives conceptuelles. Ce niveau axiomatique se décompose en un **ensemble de schémas d'axiome** correspondant aux propriétés classiques des ontologies légères et un **ensemble d'axiomes** (appelés également axiomes de domaine) complétant si besoin est l'expression de la sémantique des primitives (cf. figure 1).

Un schéma d'axiome possède une forme prédéfinie fixe, instanciée avec des primitives conceptuelles, et porte sur une ou plusieurs primitives conceptuelles (généralement une seule primitive, parfois deux). Par opposition, les axiomes n'ont pas de forme prédéfinie, même s'ils s'écrivent tous sous la forme d'un couple de graphes liés par des liens entre sommets concepts. Ils présentent ainsi la même forme que les règles GC, mais ne font que spécifier une connaissance, et ne sont pas forcément destinés à produire des connaissances. De plus, ils peuvent ne pas exprimer une propriété portant sur une ou deux primitives en particulier, mais ils représentent plutôt des propriétés mettant

en jeu plusieurs primitives. Ainsi, l'axiome « *Les amis de mes amis sont mes amis* » peut être vu comme une propriété de transitivité de la relation *ami*(*Humain*,*Humain*). Mais l'axiome « *Les ennemis de mes ennemis sont mes amis* » peut difficilement être exprimé comme une propriété algébrique ou une cardinalité d'une primitive conceptuelle particulière². Ce dernier axiome sera donc représenté dans une ontologie par un axiome³.

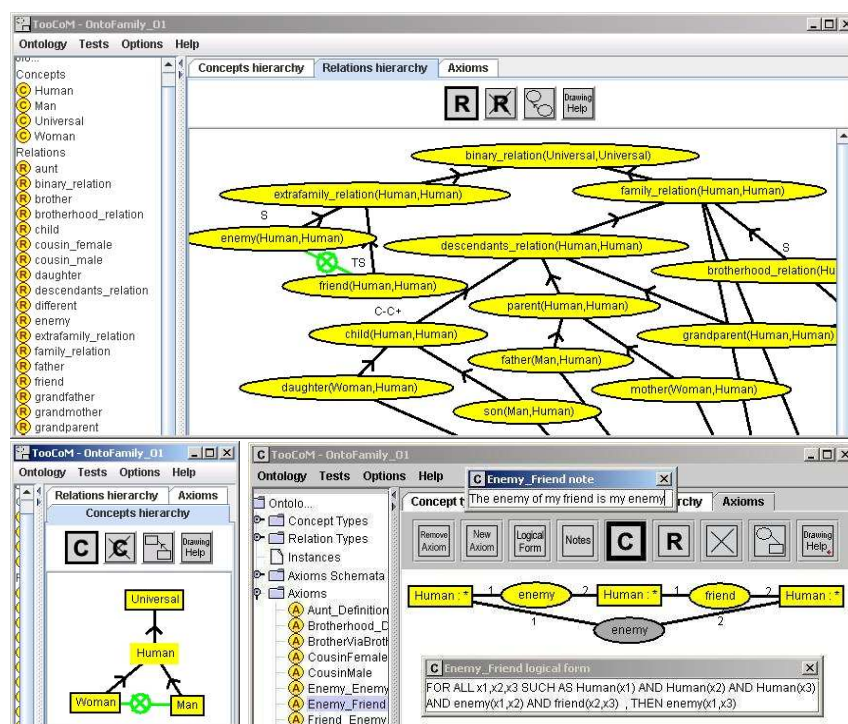


FIG. 1 – Représentation d'une ontologie des relations familiales en OCGL dans l'atelier TooCoM. La partie supérieure présente la hiérarchie de relations, la partie inférieure gauche, la hiérarchie des concepts et la partie inférieure droite, un axiome. Dans les hiérarchies, les flèches représentent des liens sorte-de, et les propriétés sont indiquées par des symboles (un cercle barré pour la disjonction entre concepts ou l'exclusivité entre relations, un S pour la symétrie d'une relation, un T pour la transitivité, etc.). Dans la partie axiome, les nœuds clairs représentent la partie antécédent, les nœuds sombres la partie conséquent. Chaque partie contient des nœuds concepts (rectangles) et des nœuds relations (ellipses).

Les schémas d'axiome spécifiés sont représentés par des symboles décorant les

2. A la rigueur, cet axiome peut être vu comme une sorte de combinaison entre l'inverse d'une relation et une transitivité.

3. Les axiomes et schémas d'axiome ont pour but commun l'expression de la sémantique du domaine. C'est pourquoi, dans la suite, et quand cela ne prête pas à confusion, le terme axiome sera utilisé pour désigner à la fois les axiomes (de domaine) et les schémas d'axiome.

hiérarchies. Ceci permet à l'utilisateur d'avoir une vue aussi globale et synthétique que possible de l'ontologie. Les schémas d'axiome portant sur les concepts sont : le lien **sorte-de** entre concepts, l'**abstraction** d'un concept (un concept c est abstrait si toute instance de c est aussi instance d'un de ses concepts fils), la **disjonction** entre deux concepts c_1 et c_2 (deux concepts c_1 et c_2 sont disjoints si aucune instance commune aux deux concepts n'existe. Les propriétés combinées d'abstraction d'un concept et de disjonction deux à deux entre ses concepts fils indiquent que les concepts fils constituent une partition du concept père.).

Les schémas d'axiome portant sur les relations sont : le lien **sorte-de** entre relations, la **signature** d'une relation, qui précise les concepts les plus spécifiques qu'elle peut lier, l'**incompatibilité** de deux relations ayant même signature, qui indique qu'il est impossible que ces relations lient le même ensemble d'instances, l'**exclusivité** de deux relations ayant même signature, qui indique que si une des relations est niée entre individus, alors l'autre est établie, les **propriétés algébriques** d'une relation binaire liant les mêmes concepts (symétrie, transitivité, réflexivité, anti-réflexivité, anti-symétrie), les **cardinalités maximum et minimum** portant sur un concept donné de la signature d'une relation.

Le seul schéma d'axiome portant sur les instances est le **type** d'une instance : le type d'une instance est le concept le plus spécifique auquel elle appartient.

Les schémas d'axiome ne permettant pas toujours d'exprimer toute la sémantique d'un domaine, il est nécessaire d'ajouter au langage des **axiomes** (ou axiomes de domaine) offrant la possibilité d'exprimer graphiquement des propriétés. Un axiome est constitué d'un couple de graphes conceptuels, liés par des nœuds concepts, et construits sur le support (au sens du modèle des GCs) constitué par les ensembles de concepts, de relations et d'instances ontologiques, les relations sorte-de, les signatures des relations, et les types des instances. Les sommets concepts sont, soit des concepts génériques, au sens des Graphes Conceptuels⁴, soit des instances ontologiques.

3 Conclusion

Dans cet article, nous présentons OCGL, un langage de représentation d'ontologie utilisant le modèle des Graphes Conceptuels. L'un des principaux avantages d'OCGL par rapport aux langages de représentation d'ontologie existants est son aspect graphique qui offre un formalisme intuitif et facilement manipulable. D'autre part, OCGL s'appuie sur la SG-Family pour représenter tout type de connaissances axiomatiques, sans se limiter aux propriétés classiquement considérées dans les ontologies légères, comme c'est à l'heure actuelle le cas dans les langages du Web sémantique, RDF et OWL. Enfin, OCGL permet de spécifier ces connaissances axiomatiques sans leur imposer de sémantique opérationnelle, ce qui augmente la réutilisabilité des ontologies. Cette approche diffère donc de celles qui visent à ajouter aux ontologies légères des règles (par exemple dans le langage SWRL) ou des contraintes (par exemple dans le langage PAL de l'outil Protégé) dont la sémantique opérationnelle conditionne l'usage.

4. Ils sont alors étiquetés par un type et par le marqueur *.

Références

- Baget, J. and Mugnier, M. (2002). Extensions of Simple Conceptual Graphs: the Complexity of Rules and Constraints. *Journal of Artificial Intelligence Research*, 16:425–465.
- Fürst, F., Leclère, M., and Trichet, F. (2004). Operationalizing domain ontologies: a method and a tool. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'2004)*, volume 110, pages 318–322. IOS Press.
- Gomez-Perez, A., Fernandez-Lopez, M., and Corcho, O. (2003). *Ontological Engineering*. Springer-Verlag, Advanced Information and Knowledge Processing.
- Sowa, J. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.
- Staab, S. and Maedche, A. (2000). Axioms are objects too: Ontology engineering beyond the modeling of concepts and relations. Research report 399, Institute AIFB, Karlsruhe.

Summary

This article presents OCGL (Ontology Conceptual Graph Language), a language dedicated to the representation of ontologies and based on the Conceptual Graphs model. The article describes in detail how an ontology is designed in OCGL, and shows the implementation of this language in the ontological engineering tool TooCoM.

Cartes cognitives de graphes conceptuels

David Genest, Stéphane Loiseau

LERIA – Université d'Angers, 2, Boulevard Lavoisier – 49045 Angers cedex 1
{genest,loiseau}@info.univ-angers.fr

Résumé. Le modèle des cartes cognitives offre une représentation graphique d'un réseau d'influences entre différentes notions. Nous proposons un nouveau modèle de cartes cognitives qui intègre la partie représentation des connaissances et l'opération de projection du modèle des graphes conceptuels.

1 Introduction

Une carte cognitive (Tolman 1948) contient deux types d'informations : des nœuds appelés états représentant des concepts et des arcs entre ces nœuds représentant des liens d'influence positifs ou négatifs. Un mécanisme d'inférence propage les influences.

Une première faiblesse des cartes cognitives est sa trop grande souplesse car un état peut être représenté par n'importe quelle étiquette linguistique. Une seconde faiblesse du modèle est l'absence de structuration des états, qui fait que des liens entre états, autres que ceux d'influence, ne peuvent pas être exprimés.

Le modèle des graphes conceptuels (Sowa 1984) est un modèle de représentation graphique de connaissances. Un graphe conceptuel est défini sur une structure appelée support permettant de spécifier en hiérarchie le vocabulaire. Une opération d'inférence, appelée projection, permet de rechercher des graphes qui sont sémantiquement liés entre eux.

L'idée du modèle des *cartes cognitives de graphes conceptuels* que nous proposons consiste à décrire chaque état par un graphe. D'abord, l'utilisation d'un graphe conceptuel, associé à chaque état, permet de définir chaque état en référence à une ontologie qui est le support. Ensuite, on peut calculer ou regrouper des classes d'états qui sont liés entre eux dans une collection. Enfin, cette classification peut se combiner avec le calcul d'influence.

Dans la partie 2, nous décrivons le modèle des cartes cognitives de graphes conceptuels. La partie 3 décrit la notion de collection. Dans la partie 4, nous définissons les opérations permettant le raisonnement dans le modèle et décrivons les apports de ces opérations.

2 Modèle des cartes cognitives de graphes conceptuels

Le modèle des graphes conceptuels utilisé est celui défini dans (Chein et Mugnier 1992). Tout graphe conceptuel est défini sur un support qui organise, à l'aide de relations « sorte de », un vocabulaire composé de types de concepts et de types de relations (figure 1). Un graphe conceptuel G est formé d'un ensemble de sommets concepts (C_G), un ensemble de sommets relations (R_G), un ensemble d'arêtes (E_G) et une application qui associe à tout sommet et à toute arête une étiquette ($étiq_G$). Le graphe conceptuel de la figure 2 représente un accident mortel (accident dans lequel une personne est morte)

Une carte cognitive de graphes conceptuels permet de représenter des relations d'influence entre différentes notions, appelées états, chacun de ces états étant défini par un graphe conceptuel.

Cartes cognitives de graphes conceptuels

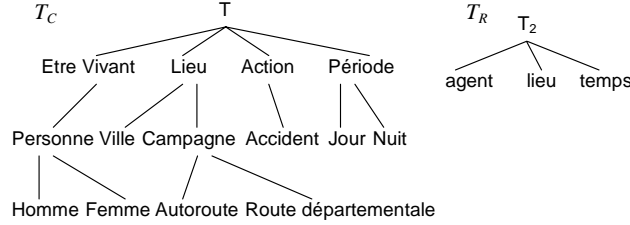


FIG. 1 – Un support

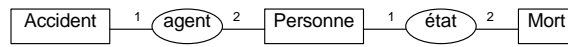


FIG. 2 – Un graphe conceptuel

Définition : état.

Un état (*d'une carte cognitive de graphes conceptuels*) défini sur un support S est un couple (i, G) où i est un intitulé décrivant l'état et $G = (C_G, R_G, E_G, \text{étiqu}_G)$ un graphe conceptuel défini sur S . On dit que G définit l'état.

Définition : carte cognitive de graphes conceptuels.

Une *carte cognitive de graphes conceptuels* définie sur un support S est un graphe orienté $X = (E_X, L_X, \text{étiqu}_X)$ où :

E_X est l'ensemble des nœuds du graphe.

L_X est l'ensemble des arcs du graphe, appelés *liens* de la carte.

étiqu_X est une fonction d'étiquetage qui à tout élément l de L_X associe une étiquette $\text{étiqu}_X(l) \in \{+, -\}$ et à tout élément e de E_X associe un état défini sur S : $\text{étiqu}_X(e) = (i_e, G_e)$.

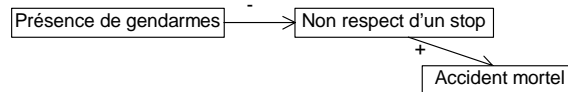


FIG. 3 – Une carte cognitive de graphes conceptuels (graphes conceptuels masqués)

La figure 3 présente un exemple de carte cognitive de graphes conceptuels dans laquelle les graphes conceptuels ont été masqués. La partie droite de la carte peut être interprétée de la façon suivante : « le non respect d'un stop augmente le risque d'accident mortel ». L'influence est représentée par un lien étiqueté $+$. La carte représente aussi que « la présence de gendarmes diminue les infractions de non respect du stop ».

3 Sélection d'états

Dans cette partie nous décrivons comment les classes d'états, appelées *collections*, peuvent être construites.

3.1 Collection d'états

Définition : collection d'états.

Une *collection d'états* F d'une carte cognitive de graphes conceptuels $X = (E_X, L_X, \text{étiq}_X)$ est un sous-ensemble de l'ensemble des états de X : $F \subseteq E_X$. Un *intitulé* peut lui être associé, ce qui permet de nommer une collection.

Par exemple, une collection intitulée « accidents » peut être créée manuellement par le concepteur de la carte et contenir tous les états où il est question d'accidents. Toutefois, la collection intitulée « accidents » pourrait regrouper automatiquement tous les états qui contiennent un sommet concept de type « Accident » dans leur graphe de définition.

3.2 Détermination automatique d'une collection

La détermination automatique d'une collection va s'effectuer à partir d'un graphe par l'opération de projection (Mugnier et Chein 1996). De façon intuitive, un graphe G se projette dans un graphe H si l'information représentée par G est incluse dans H .

Définition : collection d'un graphe requête.

La *collection d'un graphe requête* G défini sur un support S dans une carte cognitive de graphes conceptuels $X = (E_X, L_X, \text{étiq}_X)$ définie sur le même support, est la collection formée des nœuds $e \in E_X$ étiquetés par (i_e, G_e) tels qu'il existe une projection de G dans G_e .

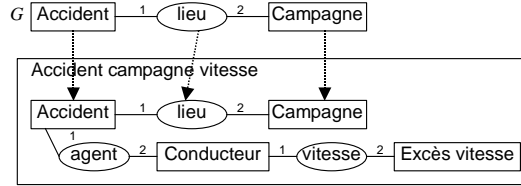


FIG. 4 – Calcul de la collection d'un graphe requête

La figure 4 montre un exemple d'utilisation d'un graphe requête. En créant le graphe requête G représentant un « accident à la campagne », il est possible d'interroger la carte pour déterminer les états qui concernent des accidents à la campagne. Par extension, il est possible de calculer la collection d'un état requête, à partir d'un état de la carte. Cette collection est calculée en considérant le graphe de définition de l'état comme graphe requête. Ainsi, choisir un état requête e étiqueté par un graphe G est équivalent à choisir tous les états qui sont étiquetés par des graphes qui représentent des faits plus spécialisés que G . On les choisit, mais sans être obligé de les spécifier explicitement, grâce à la projection.

4 Inférence

Outre la sémantique associée à chaque état par un graphe conceptuel, un des points forts de ce nouveau modèle réside dans le raisonnement qu'il autorise.

4.1 Propagation

L'effet que produit un état sur un autre est fonction des chemins qui existent dans la carte entre ces deux états, et des étiquettes portées par les liens. Cet effet peut être positif (noté +), négatif (-), nul (0) ou ambigu (?).

Définition : influence.

L'influence I de X est une application de $E_X \times E_X$ dans $\{+, -, 0, ?\}$ telle que

$$I(e_i, e_j) = \bigoplus_{H \in H_{i,j}} \bigotimes_{k \in [1, |H|-1]} I_1(h_k, h_{k+1})$$

$H_{i,j}$ étant l'ensemble des chemins ayant pour origine e_i et extrémité e_j . Chacun de ces chemins étant de la forme $H = (h_1, \dots, h_n)$ avec $n=|H|$.

I_1 étant une application de $E_X \times E_X$ dans $\{+, -, 0\}$ telle que $I_1(e_i, e_j) = \text{étiqu}_X(l)$ si il existe un $l = (e_i, e_j)$ dans L_X et 0 sinon.

\oplus et \otimes étant des applications de $\{+, -, 0, ?\} \times \{+, -, 0, ?\}$ dans $\{+, -, 0, ?\}$ définies ainsi :

\oplus	+	-	0	?
+	+	?	+	?
-	?	-	-	?
0	+	-	0	?
?	?	?	?	?

\otimes	+	-	0	?
+	+	-	0	?
-	-	+	0	?
0	0	0	0	0
?	?	?	0	?

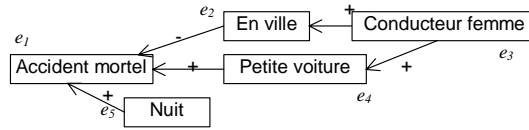


FIG. 5 – Une carte cognitive

La figure 5 peut être interprétée par « Les femmes roulent beaucoup en ville, et conduisent souvent de petites voitures. Le risque d'accident mortel est plus fort la nuit, et avec de petites voitures, et moins fort en ville ». $I(e_5, e_1) = I_1(e_5, e_1) = +$. On peut en déduire qu'il y a plus de risques d'accidents mortels la nuit. $I(e_3, e_1) = (I_1(e_3, e_2) \otimes I_1(e_2, e_1)) \oplus (I_1(e_3, e_4) \otimes I_1(e_4, e_1)) = (+ \otimes -) \oplus (+ \otimes +) = ?$. L'influence est donc ambiguë, ce qui signifie qu'il y a à la fois une influence positive et une influence négative.

4.2 Influence et collections

Pour l'aide à la prise de décision, un utilisateur préfère souvent sélectionner un ou plusieurs états *choix* e_1, e_2, \dots, e_m , correspondant aux alternatives qu'il envisage et un état *cible* e_j , correspondant à l'objectif qu'il s'est fixé. Le système doit alors déterminer l'influence de ces différents choix sur la cible : l'influence de chaque état choix sur l'état cible peut alors être utilisée pour renseigner l'utilisateur. Si $I(e_x, e_j) = +$, l'état choix e_x a un effet positif sur l'état cible, et le choix doit être considéré. Si la valeur obtenue est -, l'effet est négatif, et le choix est rejeté. Enfin, si la valeur obtenue est 0 ou ?, aucun conseil ne peut être donné. Par exemple, afin de repérer les causes d'accidents mortels pour cibler une

campagne de prévention, en utilisant la carte cognitive de la Figure 5, on peut choisir « accident mortel » comme cible et « nuit » et « conducteur femme » comme choix. Le calcul des influences donné en exemple plus haut permet de déduire que les conducteurs de nuit peuvent être ciblés dans la campagne ($I(e_5, e_1) = +$), mais le système ne donne aucun conseil sur les conductrices ($I(e_3, e_1) = ?$). Plutôt que de construire manuellement une collection, l'utilisateur peut choisir de définir un graphe requête, ou partir d'un état pour demander le calcul de la collection du graphe associé. La collection correspondante est alors automatiquement calculée, et les influences sont présentées à l'utilisateur.

5 Conclusion

Le modèle présenté ici a deux atouts essentiels. Le premier concerne la représentation des connaissances qui combine les caractéristiques des graphes conceptuels et des cartes cognitives. D'abord, le support sur lequel est définie une carte permet une représentation des connaissances plus formelle. Ensuite, des regroupements d'états sémantiquement liés peuvent être calculés ou construits grâce au concept de collection. Enfin, des relations d'influence entre graphes conceptuels peuvent être représentées et exploitées par l'utilisateur. Le deuxième avantage réside dans la façon de raisonner. L'opération de projection du modèle des graphes conceptuels est combinée avec les opérations de propagation d'influences. On obtient de cette façon un nouveau type d'inférence qui offre la possibilité de calculer une collection d'états. Il est alors possible de prendre en compte des ensembles d'états dans les opérations, plutôt que des états isolés. Au final, le modèle proposé dispose d'une représentation graphique simple à interpréter pour l'utilisateur et d'opérations qui aident l'utilisateur à exploiter les connaissances représentées. Il s'agit donc d'un modèle qui est adapté à l'aide à la prise de décision, car il offre des facilités à l'utilisateur afin qu'il puisse faire son choix. Des extensions à des cartes dont l'influence est numérique serait aisée en prenant une des sémantiques fournies dans (Louça 2000).

Références

- Chein M. et Mugnier M.-L. (1992), Conceptual Graphs: Fundamental Notions. *Revue d'Intelligence Artificielle*, 6(4), pp 365-406, 1992.
- Louça J.A. (2000), Cartographie cognitive, réflexion stratégique et interaction distribuée. Doctorat de Paris IX-Dauphine, 2000.
- Mugnier M.-L. et Chein M. (1996), Représenter des connaissances et raisonner avec des graphes. *Revue d'intelligence artificielle*, 10(1), pp 7-56. 1996.
- Sowa J.F. (1984), *Conceptual Structures : Information Processing in Mind and Machine*. Addison Wesley, 1984.
- Tolman E. (1948), Cognitive maps in rats and men. *Psychological Review*, Vol. 55, 1948.

Summary

The cognitive map of conceptual graph offers a graphical representation of influences between notions. We propose a new model of cognitive maps where each notion is precisely defined by a conceptual graph. One of the major advantages of our model is the combination of operations of cognitive maps and operations of conceptual graphs.

Modélisation des connaissances émotionnelles par les cartes cognitives floues

Nathalie Ronarc'h², Gaële Rozec¹,
Fabrice Guillet², Alexis Nédélec³, Serge Baquedano¹, Vincent Philippé¹
¹ Performanse SA Atlanpôle La Fleuriaye 44470 CARQUEFOU
<http://www.performanse.fr>

² LINA - Polytech' Nantes rue Christian Pauc BP50609 44306 Nantes CEDEX 3
{Prenom.Nom}@polytech.univ-nantes.fr

³ CERV/ ENIB de Brest Technopôle Brest Iroise ; CP 15 ; 29608 Brest Cedex
{nom}@enib.fr

Résumé: Les recherches en psychologie ont permis d'établir une relation entre émotions et prise de décision. La prise en compte de caractéristiques humaines telles que les émotions et la personnalité dans les processus d'interaction entre agents est au centre de ce travail. Il s'inscrit dans le cadre du projet GRACE (Groupes Relationnels d'Agents Collaborateurs Emotionnels)/ RIAM (Réseau des Industries, de l'Audiovisuel et du Multimédia) .

1 Introduction

Aujourd'hui peu d'outils permettent de décrire facilement les comportements d'individus. Les sciences humaines apportent leur expertise en proposant des modèles émotionnels décrivant le processus de décision et les comportements des humains dans un contexte donné. Les cartes cognitives émotionnelles basées sur le modèle Performanse SA nous servent d'outil de recueil d'informations sur l'évolution des émotions, suite à un événement donné. Douze émotions, influencées par les traits de personnalité de l'individu, sont ainsi représentées. La problématique est de traduire ces modèles en langage informatique. La combinaison de la socio-psychologie et du domaine multi-agent (Ferber 95) nous apporte les éléments nécessaires à la modélisation pertinente des comportements et des interactions d'agents humains pouvant évoluer dans un milieu virtuel.

Dans ce contexte les travaux du CERV/ LI2, associé au projet GRACE / RIAM , sur les SMA et la réalité virtuelle ont donné lieu à l'élaboration d'une simulation sur la plateforme AréVi (Harrouet) dans laquelle les entités considérées, c'est à dire des agents autonomes, sont capables de percevoir tout ou partie de leur environnement, de réagir aux événements en fonction de leur état interne et de leurs connaissances. Nos agents possèdent des capacités de raisonnement, ils sont munis d'états mentaux, ce sont des agents cognitifs. De plus les agents ont une personnalité qui nous permet de les distinguer.

2 Représentation du processus de décision de l'agent

Le processus de prise de décision de l'agent repose sur la boucle Perception – Décision – Action du fonctionnement des agents cognitifs. Ainsi les cartes cognitives émotionnelles (CEF) sont intégrées à un dispositif plus complexe de prise de décision de l'agent, tenant compte de ses connaissances, de l'évaluation de l'événement perçu et de son estimation de l'environnement. Le processus de prise de décision des agents, représenté Figure 1 et

Modélisation des connaissances émotionnelles par les cartes cognitives floues

proposé par l'expertise psychologique, comprend différents modules permettant d'affiner, d'ajuster et de rendre cohérents les comportements des agents en situation d'interactions sociales:

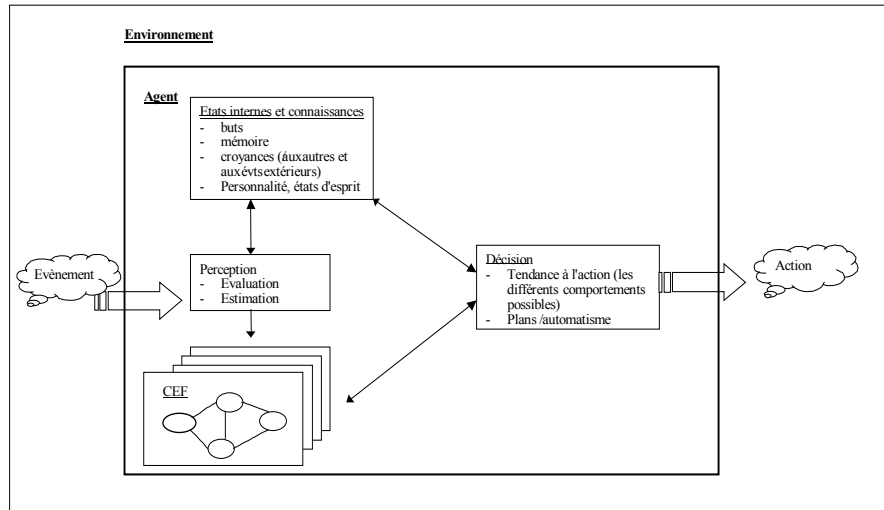


FIG. 1- Architecture générale du processus de prise de décision de l'agent

- un **module d'évaluation** de l'événement, composé de capteurs de l'environnement
- les **Etats internes** propres à l'agent jouant un rôle essentiel dans la prise de décision
- des **CEF** (FEM) sont des cartes cognitives émotionnelles capables de régir les émotions
- un **module de décision**, chargé d'évaluer le comportement à exprimer

L'agent éprouve une émotion selon l'événement provenant de l'extérieur, il y répond par un certain comportement, suite à un processus de décision, mêlant la dynamique des cartes cognitives émotionnelles mises en œuvre ainsi que les connaissances de l'agent et sa perception. Le comportement d'un agent est donc perceptible en terme d'actions, de plans, de décisions, d'interactions... qui découlent de sa structure et ses mécanismes internes.

3 Principe des cartes cognitives floues

Les cartes cognitives sont issues des travaux des psychologues (Tolman 48) qui introduisirent ce concept pour décrire le comportement animal. Tolman a étudié la capacité d'adaptation des rats face à des changements intervenus dans leur environnement, représentée par des cartes comportementales ou cartes cognitives. Il s'est aperçu que la représentation graphique d'une carte cognitive pouvait aider à la compréhension des mécanismes psychologiques qui justifie le comportement (*cognitive maps* (Tolman 48)). Elles furent associées à la logique floue pour devenir les cartes cognitives floues (CCF ou FCM : *Fuzzy Cognitive Maps* (Kosko 86)).

A l'image des réseaux sémantiques (Sowa 91), les CCF sont des graphes orientés dont les nœuds sont des concepts (C_i) et les arcs des liens d'influence munis de poids (E_{ij}) entre ces

concepts (figure 2) mesurant le *degré d'influence*. Un degré d'activation (A_i) est associé à chaque concept, alors que le poids E_{ij} d'un arc traduit une relation d'inhibition ($E_{ij} < 0$) ou d'excitation ($E_{ij} > 0$) du concept C_i vers le concept C_j . Le niveau d'activation A_i de chaque concept C_i et C_j est calculée par la règle suivante (Aguilar 03):

$$A_i^{(t)} = \sum (A_j^{(t)} E_{ji}) + A_i^{t-1}$$

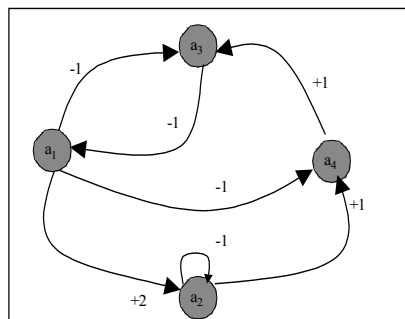
Où

$A_i^{(t)}$ est le niveau d'activation C_i à l'instant t

A_i^{t-1} est le niveau d'activation C_i à $t - 1$ (niveau précédent)

E_{ji} est la matrice indiquant la valeur du poids entre les concepts C_i et C_j

La dynamique de la carte est calculée mathématiquement par produit matriciel.



La carte ci-contre est formée de 4 concepts et possède 7 arcs. Chaque concept C_i a un degré d'activation A_i . E est la matrice des liens

$$E = \begin{pmatrix} 0 & +2 & -1 & -1 \\ 0 & -1 & 0 & +1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & +1 & 0 \end{pmatrix}$$

Un zéro dans la matrice des liens $E_{ij} = 0$ désigne l'absence d'arc du concept C_i vers le concept C_j et un élément non nul de la diagonale $E_{ii} = 0$ correspond à un arc du concept C_i sur lui-même.

Ici par ex $a_3^{(t)} = a_1^{(t)} * (-1) + a_4^{(t)} * (+1) + a_3^{t-1}$

FIG. 2 - Schéma de principe des cartes cognitives (Tisseau 01)

4 Les cartes cognitives émotionnelles

Les cartes cognitives émotionnelles sont basées sur l'expertise psychologique de la société Performanse SA (Gras et al 03) et fonctionnent sur le principe des CCF. Les concepts peuvent être les émotions et le poids des arcs, les traits de personnalité. Nous avons une carte cognitive émotionnelle par type d'émotion.)

Les 12 émotions retenues sont représentées dans 12 CCF. Le modèle Performanse SA s'inspire de l'approche du modèle OCC (Ortony et al. 1988). Des émotions positives et négatives ont été distinguées, selon qu'elles sont :

- propres à l'individu : Joie - Tristesse et Fierté - Honte
- en rapport aux échanges relationnels: Approbation - Opposition et Sympathie - Antipathie
- en rapport aux événements extérieurs: Espoir - Inquiétude et Satisfaction - Déception

Les paramètres variables dans le comportement des agents sont leurs états émotionnels. Les traits de personnalité, eux, sont fixes. Le modèle de comportement, sur lequel nous nous basons, est donc fonction des traits de personnalité de l'individu. Celui-ci a été mis en évidence à travers un modèle de personnalité, conçu par la société Performanse SA dans le cadre du logiciel **Dialecho** qui détermine le profil comportemental d'un individu à partir des réponses à un questionnaire regroupant 70 questions (Gras et al 03).

Les 10 dimensions comportementales bipolaires, ou *compétences comportementales*, dominants chez l'homme, et leurs antagonistes, sont regroupés ci-dessous:

Modélisation des connaissances émotionnelles par les cartes cognitives floues

EXT extraversion/ INT introversion; DIN dynamisme intellectuel/ CIN conformisme intellectuel; ANX anxiété/ DET détente; COM combativité/ CCL conciliation; AFF affirmation/RMC remise en cause; REA réalisation/ FAC facilitation; REC réceptivité/ DTM détermination; APP appartenance/ IND indépendance; RIG rigueur/ IMP improvisation; POU pouvoir/ PRO protection.

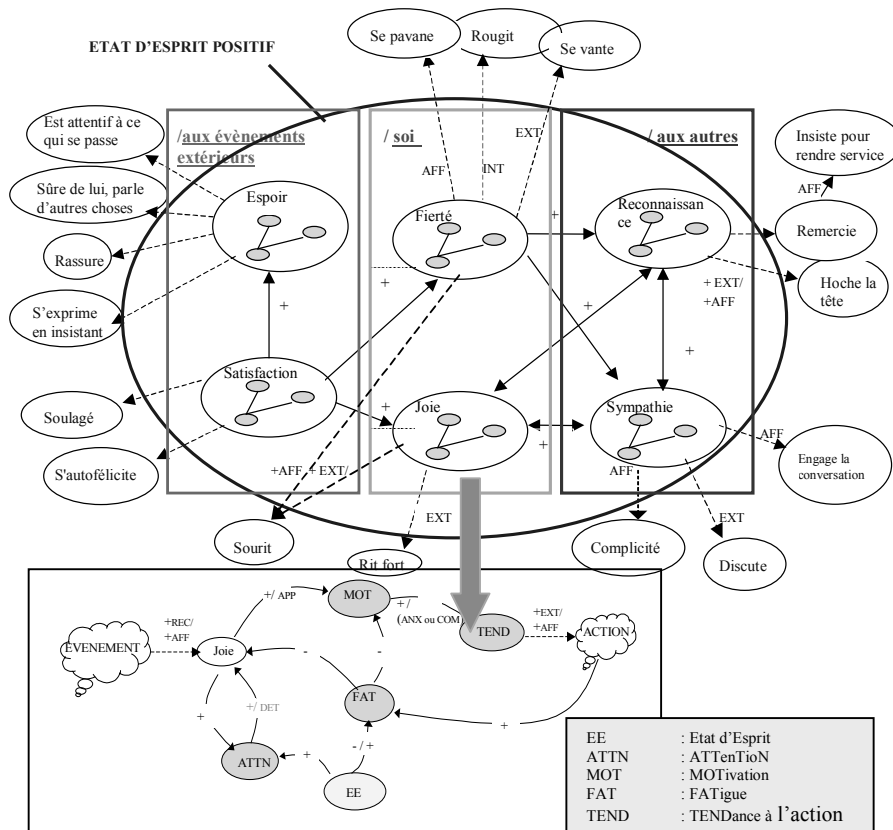


FIG. 3- Les cartes sont reliées entre elles, de même avec l'état d'esprit négatif.

La figure 3 présente un exemple de carte pour l'émotion JOIE. Suivant l'événement l'agent exprime une certaine émotion, comme par exemple la JOIE. Les concepts (l'émotion, l'attention à, la fatigue, la tendance à l'action et l'état d'esprit) sont reliées par des arcs pondérés par les profils comportementaux représentés par les traits de personnalité, ceux-ci sont pris en compte dans le calcul du niveau d'activation des concepts. Ces liens d'influence traduisent les relations excitatrices ou inhibitrices entre les concepts associés. La personnalité définit donc la façon dont varient les émotions.

Les 6 cartes des émotions positives sont reliées entre elles et représentent un état d'esprit positif (comme le montre la figure 3) et sont également reliées aux 6 cartes des émotions

négatives, représentant l'état d'esprit négatif. Ainsi l'état d'esprit montre une vue d'ensemble des états émotionnels chez un même agent.

5 Conclusion

La problématique est d'intégrer les notions complexes d'émotions et de personnalité afin de modéliser des comportements d'agents les plus fidèles. Au vu de ces différents éléments, l'utilisation de cartes cognitives floues, complétée des modules d'évaluation, de connaissances et de décision, nous apparaît donc pertinente et originale pour modéliser les trois niveaux représentatifs d'un comportement.

Formaliser des connaissances liées à la psychologie humaine permet d'envisager des perspectives intéressantes telles que la simulation des comportements et les dynamiques de groupe. Les domaines d'applications sont alors très divers : jeux, contexte professionnel.

Références

- Aguilar J. (2003), A Dynamic Fuzzy- Cognitive Map Approach based on Random Neural Networks, *International Journal of Computational Cognition*, Vol 1, number 4, Pages 91 – 107, December 2003.
- Ferber J. (1995), Les Systèmes Multi-Agents, vers une intelligence, *Inter-éditions* 1995
- Gras R., Peter P., Baquedanno S. et Philippé J. (2003), Structuration de comportements de réponse à un questionnaire par des méthodes multi-dimensionnelles, *Numéro spécial Extraction et Gestion des Connaissances, Extraction des connaissances et Apprentissage*, Vol 17, number 1-3, Pages 105-118, Hermès, 2003.
- Kosko B. (1986), Fuzzy Cognitive Maps, *International Journal Man-Machine Studies* Vol 24, Pages 65-75, 1986.
- Harrouet, <http://www.enib.fr/~harrouet/>
- Louçà J.A. (2000), Cartographie Cognitive, Réflexion Stratégique et Interaction Distribuée: une Approche Multi-Agent, *Thèse de doctorat*, Université Paris Dauphine, Juin 2000
- Ortony A., Clore G., Collins A. (1988). The cognitive structure of emotion. *Cambridge University Press*, 1988.
- Sowa J.F.(1991), Principles of semantic network : exploration in the representation of knowledge, *Morgan Kaufmann Publisher Inc.*, San Mateo, 1991.
- Tisseau J. (2001), Réalité Virtuelle, autonomie in virtuo, Habilitation à diriger des recherches, *Université de Rennes I*, Décembre 2001.
- Tolman E.C. (1948), Cognitive maps in rats and men, *Psychological Review* Vol 42, number 55, Pages 189-208, 1948.
- Wooldridge M. (2001), An introduction to Multiagent Systems. Wiley, 2001.

Summary

Research in psychology made it possible to establish a relation between emotions and decision-making. A person deprived of emotions would be unable to make a decision or to carry out a reasoning. The taking into account of human characteristics such as the emotions and the personality in the processes of interaction between agents is in the center of this work.

Logique Floue appliquée à l'inférence du « Risque Inhérent » en audit financier

Souhir Fendri-Kharat*
Hassouna Fedhila**
Pierre-Yves Glorennec***

* Ecole Supérieure de Commerce, Rte de l'Aéroport-Km 4, BP 1081, Sfax 3018-Tunisie
[souhir.kharat@escs.rnu.tn]

** Institut Supérieur de Comptabilité et d'Administration des Entreprises, Campus, Manouba
2001-Tunisie
[hass530@yahoo.fr]

*** Institut National des Sciences Appliquées de Rennes, 20, Av. des buttes de Coësmes-CS
14315-35043 RENNES Cedex-France
[pierre-yves.glorennec@irisa.fr]

Résumé : Le Risque d'Audit est un indice d'existence d'erreurs dans les états financiers d'une entreprise. Trois modèles mathématiques sont associés à ce concept du RA : un modèle « Bayésien », un modèle « évidentialiste », et un modèle « flou ». Ces trois modèles accusent des incohérences mathématiques et des difficultés d'application pratique, surtout au niveau de la composante « Risque Inhérent » du risque d'audit. Ils considèrent le processus cognitif d'estimation du RI en tant que « boîte noire ». Nous proposons un simple algorithme d'inférence flou interprétable pour capter le processus cognitif d'estimation du RI, algorithme basé sur l'induction d'arbre de décision flou. Notre objectif est d'identifier les éléments de cette structure et de démontrer que l'utilisation d'une telle structure d'inférence floue est proche de la décision réelle d'estimation du RI. Il s'agit d'une recherche exploratoire et expérimentale.

1 Introduction

Aussi bien les normes internationales [ISA]¹ que les normes américaines [SAS]¹ d'audit s'accordent sur le fait que l'audit financier est un audit de conformité entre les réglementations en vigueur et les états financiers d'une entreprise. Les SAS n° 39, 47 & 55 et les ISA n° 400 à 408, stipulent que la non-conformité de la comptabilité d'une firme à des réglementations en vigueur, est l'essence même de l'erreur comptable (AICPA 2003 et IFAC 2003). Le concept de « Risque d'Audit » [RA] est un indice de l'occurrence d'erreurs dans les rapports financiers : ces normes conceptualisent le « Risque d'Audit » [RA] en tant qu'une intersection entre trois ensembles, à savoir : « Risque Inhérent » [RI], « Risque de Non Contrôle » [RNC] et « Risque de Non Détection » [RND]. La première composante RI, indique l'ensemble des erreurs pouvant s'infiltrer dans les états financiers et provenant de

¹ SAS :Statements on Auditing Standards (normes d'audit de l'AICPA)

ISA :International Standards of Auditing (normes d'audit de l'IFAC)

AICPA : ...American Institute of Certified Public Accountants (Ordre des experts comptables-USA).

IFAC :International Federation of Accountants (Fédération internationale des experts comptables).

l'environnement de la firme auditée. La seconde, RNC, indique l'ensemble des erreurs pouvant s'infiltrer à travers le système de contrôle interne de la firme auditée. Enfin, la troisième, RND, indique l'ensemble des erreurs pouvant s'infiltrer par manque d'efficacité de l'auditeur. Par conséquent, le RA désigne l'ensemble des erreurs pouvant s'infiltrer dans les états financiers d'une firme auditée -durant une période comptable donnée- et ce malgré l'existence de contrôles dans l'environnement et dans le système de contrôle interne de la firme auditée, et malgré la compétence de l'auditeur. Seule la composante RI fait l'objet de notre étude. Trois théories mathématiques sont associées au concept « ensembliste » du RA et s'appliquent par conséquent au RI :

D'abord, la théorie bayésienne de la probabilité, adoptée par les normes dès 1983, exprime les trois composantes du RA par des probabilités subjectives et leur intersection par une simple multiplication arithmétique. Or, une croyance dépend pour beaucoup de son contexte, elle ne peut pas logiquement être générée par une règle basée sur une fréquence ou répétition d'occurrence (Shafer 1976, Dusenbury et al. 1996). En plus, la théorie de la Probabilité bayésienne est inadéquate pour une structure dont les éléments sont dépendants entre eux, et essentiellement qualitatifs (Lea et al. 1992). Ce qui est le cas des facteurs déterminants du RI (Friedlob et Schleifer 1999). En pratique d'audit, les firmes d'audit sont en train progressivement d'abandonner ce modèle au profit de modèles linguistiques conventionnés (Casta et Lesage 2003).

Ensuite, la théorie de l'Evidence (fonction de croyance de Shafer 1976) a été appliquée au RA en remplaçant les probabilités subjectives par des vecteurs masses d'évidence, et l'intersection par la règle de Dempster-Shafer (Srivastava et Sahfer 1992). Mais ce modèle souffre des mêmes problèmes d'indépendance et d'inadéquation que le modèle bayésien (Srivastava et Lu 2002).

Enfin, la théorie de la Logique Floue (Zadeh 1965) a été appliquée au RA en remplaçant les composantes par des nombres flous trapézoïdaux, et l'intersection par le « et » optimiste (max-min) et le « ou » de Bühler, flous (Lesage 2000). Ce modèle semble rencontrer moins de difficultés théoriques, et être adapté aux modèles linguistiques du RA adoptés par les praticiens. L'adoption d'une approche inductive comme l'approche cognitive explique en partie cet avantage. En plus, le choix de la Logique Floue comme base théorique permet d'assimiler tout subjectivisme inhérent à l'estimation d'une croyance (Lesage 2000 et Casta et Lesage 2003).

D'autres auteurs se sont intéressés au RA. Ils ont utilisé notamment les réseaux de neurones comme outil d'apprentissage pour l'estimation de certains aspects de l'audit financier, sans pour autant affirmer qu'un modèle neuronal complet du RA ait pu être développé (Calderon & Cheh 2002). Pour tous ces modèles ci-exposés, plusieurs problèmes sont mis en évidence au niveau spécifique de l'estimation du RI :

2 Problématique

Au niveau théorique mathématique, tous ces modèles formalisent un RI simple et ne le structurent pas. Le RI n'est qu'une simple probabilité subjective, ou vecteur de croyance, ou nombre flou trapézoïdal. Mais aucun de ces travaux académiques ne traite comment ces estimations sont obtenues, ni par quel type d'inférence ou type de structure de raisonnement. Aucun de ces modèles n'essaye d'identifier les facteurs déterminants du RI, ni comment leurs effets sont agrégés en un niveau ou quantité unique de RI. Ces modèles mathématiques

considèrent le processus cognitif de l'estimation du RI en tant que « boîte noire ». Il y a absence de recherches dans ce sens.

Au niveau conceptuel des normes d'audit, elles concèdent que le processus d'estimation du RI se base sur des facteurs issus de l'environnement de la firme auditée, mais ne les identifient pas ou peu. Elles restent en plus muettes quant à leur mode d'agrégation. D'un côté, chaque facteur devrait avoir, à notre sens, une importance propre (ou pondération) différente de celle d'autres facteurs. D'un autre côté, un effet conjugué de plusieurs facteurs sur un niveau précis et unique de RI peut exister (Dusenbury et al. 1996, Haskins & Dirsmith 1993 et Lea et al. 1992). Ce qui est complètement ignoré par les normes d'audit ainsi que les modèles mathématiques. Nous allons alors développer une démarche quantitative du même type que celui des réseaux de neurones mais en utilisant le formalisme des arbres de décision flous.

3 Objectif et intérêt

Notre objectif est de proposer en premier lieu une structure de raisonnement pour inférer le risque inhérent d'audit financier. Il s'agit de capter le processus cognitif d'estimation du RI, grâce à la construction d'un arbre de décision flou, en identifiant et en y plaçant les facteurs, déterminants du RI, les plus discriminants pour l'auditeur. Puis, nous comptons démontrer que l'utilisation d'une telle structure d'inférence floue est proche d'une décision réelle de l'estimation du RI par l'auditeur. Il s'agit d'une recherche exploratoire et expérimentale. Elle s'intègre au domaine de la capitalisation des connaissances humaines. Elle essaye de résoudre un problème de décision.

L'intérêt de notre recherche est multiple. Elle est utile pour les auditeurs novices, qui connaissent encore peu comment estimer un RI au début d'une mission d'audit. Elle est utile pour les instances de normalisation de l'audit, qui pourraient intégrer nos résultats à une normalisation approfondie de l'estimation du RI. Enfin, elle est utile pour l'amélioration et l'approfondissement des systèmes d'extraction et de gestion des connaissances de l'auditeur, utilisés actuellement par certains grands cabinets professionnels internationaux.

4 Méthodologie

Notre objectif premier est de chercher une structure théorique valable du raisonnement de l'auditeur dans le processus d'estimation du RI d'audit. Le choix de la logique floue comme outil de modélisation et de représentation des connaissances en langage naturel, se justifie par les faits qu'elle permet d'éviter les seuils booléens, qu'elle donne plus de liberté pour les nuances et qu'elle permet de lier une croyance ou une estimation à son contexte (Glorennec 1999). Le choix de l'arbre de décision flou se justifie par les faits qu'il permet de classer le facteur optimal à chaque nœud, qu'il permet l'extraction et l'usage de règles d'inférence incomplètes, et qu'il permet de parcourir plusieurs branches en parallèle, ce que les arbres booléens ne permettent pas (Glorennec 2002).

Les arbres de décision sont un outil puissant de classification. Notre problème de classification, serait d'associer chaque entreprise auditée, à son estimation de RI, donc, de classer des combinaisons de facteurs et de modalités à une classe de valeurs (ou niveaux) de RI. Les facteurs seraient des variables explicatives (évaluées par des modalités) et la classe des niveaux du RI serait la variable expliquée. Chaque chemin, de la racine à une feuille, est une règle du type « si A et/ou B et/ou C... alors Y ».

Logique floue et inférence du risque inhérent d'audit

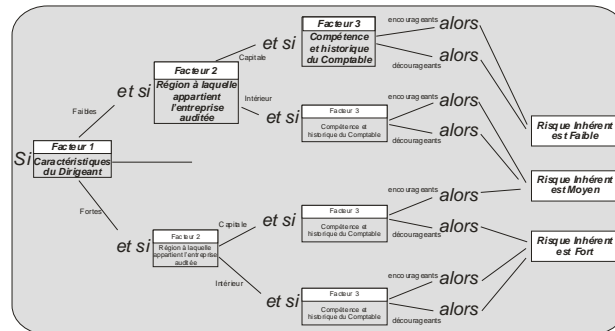


FIG. 1 - une approche d'un arbre de décision encore intuitif pour l'estimation du RI d'audit

Cet outil est un modèle d'entrée/sortie non linéaire (et non paramétrique). Sa capacité à hiérarchiser les variables selon des mesures de discrimination (entropie et gain d'information flous), le favorise par rapport aux réseaux de neurones et aux systèmes d'inférence floue. Il est en fait interprétable, lorsque des contraintes y sont insérées. Il gère aussi bien les variables numériques que symboliques (Marsala 1998), comparativement aux réseaux de neurones, qui ne sont qu'un ensemble de méthodes d'analyse et de traitement de données, non linéaires certes, utiles à la classification certes, mais des boîtes noires ininterprétables et essentiellement numériques. Il permet un résultat comparable avec un nombre plus faible de règles (Glennec 2002), comparativement aux systèmes d'inférence floue.

Les connaissances de l'auditeur sont caractérisées d'imparfaites, imprécises, vagues, majoritairement linguistiques, et approximatives (Lesage 2000 et Casta et Lesage 2003). Le moyen habituel d'obtenir des connaissances complexes est de les demander à des experts du domaine. Or, il n'est pas évident de trouver l'expert proprement dit, ni que ses connaissances « complexes » (règles) soient explicites. La construction de l'arbre doit donc se faire par induction automatique, à partir d'un ensemble de données réelles, retrouvées dans les banques de données des firmes d'audit tunisiennes. Elle permet l'extraction de connaissances complexes (règles) nouvelles et robustes, et la schématisation graphique d'un raisonnement et donc sa structuration. Les experts devraient valider ensuite les règles extraites.

5 Conclusion

Notre recherche est exploratoire dans le sens que la structuration du processus cognitif d'estimation du RI est encore un domaine de recherche vierge. Elle est aussi une recherche expérimentale, dans le sens que le mode d'inférence choisi, tel l'arbre de décision flou induit, peut ne pas correspondre au mode d'inférence cognitif réel des auditeurs en Tunisie. Notre recherche vise à passer du subjectif qu'est l'opinion des auditeurs, à l'objectif qu'est l'arbre de décision extrait automatiquement et validé à posteriori par ces mêmes auditeurs.

Références

- A.I.C.P.A. (2003) « Professional standards, U.S. auditing standards » A.I.C.P.A. Vol.1 - March 2003.
- Calderon T.G., Cheh J.J. (2002) « A Roadmap for future neural networks research in auditing and risk assessment » International Journal of Accounting Information System - Vol.3 - 2002 pp.203-236

- Casta J-F., Lesage C. (2003) « La recherche en audit : Méthodologie positive, méthodologie cognitive » Workshop CEREG - dauphine DEA audit - 26/06/2003 - 10 pages.
- Dusenbury R.B., Reimers J.L., Wheeler S.W. (1996) « An empirical study of belief-based and probability-based specification of audit risk » *Auditing : A Journal of Practice and Theory* - Vol.15 n°2 - Fall 1996 pp.12-28
- Friedlob G.T., Schleifer L.L.F. (1999) « Fuzzy logic: Application for audit risk and uncertainty » *Managerial Auditing Journal* - Vol.14 n°3 - 1999 pp.127-135.
- Glorennec P.-Y. (2002) « Induction et optimisation d'arbres de régression flous » *Systèmes numériques - projet simbad* - INSA de Rennes - IRISA Publication interne n°1363 - Janvier 2002, 38 pages.
- Glorennec P.-Y. (1999) « Algorithme d'apprentissage pour systèmes d'inférence flou » éditions Hermès - 1999.
- Haskins M.E., Dirsmith M.W. (1995) « Control and inherent risk assessments in client engagements : An examination of their interdependencies » *Journal of Accounting and Public Policies* - Vol.14 n°1 - Spring 1995, pp.63-83
- IFAC (2003) « IFAC Handbook : Normes Internationales d'Audit » traduit par CNCC et OEC, Juin 2003, 524 pages.
- Lea R.B., Adams S.J., Boykin R.F. (1992) « Modeling of the audit risk assessment process at the assertion level within an account balance » *Auditing : A Journal of Practice and Theory* - Vol.11 - Supplement 1992, pp.152-179
- Lesage C. (2000) « Audit risk assessment : An imperfect knowledge based model » in Bouchon-Meunier B., Yager R.R. & Zadeh L.A. (eds.) « *Uncertainty in intelligent and information systems, advances in fuzzy systems - applications and theories* » Vol.20, September, World Scientific, USA.
- Marsala C. (1998) « Apprentissage inductif en présence de données imprécises : Construction et utilisation d'arbres de décision flous » Thèse - Université Paris 6 - 232 p.
- Shafer G. (1976) « A Mathematical Theory of Evidence » Princeton University Press, New Jersey -1976, 297 p.
- Srivastava R.P., Lu H. (2002) « Structural analysis of audit evidence using belief functions » *Fuzzy Sets and Systems* - Vol.131 - 2002, pp.107-120
- Srivastava R.P., Shafer G.R. (1992) « Belief-function formulas for audit risk » *Accounting Review* - Vol.67 n°2 - 1992, pp.249-283
- Zadeh L.A. (1965) « Fuzzy sets » *Information and Control* - Vol.8 - June 1965, pp.338-353.

Summary

Audit risk concept is a surrogate of errors existence in financial reports. Auditing standards consider it as an aggregation of three components : IR as inherent risk, CR as control risk, and DR as detection risk. Three mathematical models are associated to audit risk concept : A Bayesian model, a belief-function model and, a fuzzy model. These three models show mathematical and practical incoherencies for IR assessment. All three consider IR assessment process as a black box. We propose a simple fuzzy inference algorithm to capture inherent risk cognitive assessment process. It is based on fuzzy decision tree. We try to see if risk level obtained from our model is near to real assessed risk level. Our research is an exploratory and experimental one. It try to resolve a decision problem.

Méthode sémantique pour la classification et l'interrogation de sources de données biologiques

Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, Malika Smail-Tabbone

UMR 7503 LORIA, BP 239, F-54506 Vandoeuvre-Lès-Nancy, FRANCE
{messai,devignes,napoli,smail}@loria.fr
<http://www.loria.fr/equipes/orpailleur>

Résumé. Nous présentons une méthode de classification et de recherche de sources biologiques. Elle consiste à construire un treillis de Galois à partir d'un ensemble de méta-données associées aux sources et converties en propriétés booléennes. Un concept construit à partir d'une requête utilisateur est ensuite inséré dans le treillis grâce à un algorithme de construction incrémentale. Le calcul du résultat se ramène à extraire l'ensemble des sources figurant dans les extensions des subsumants du concept requête dans le treillis de Galois résultant. L'ordre de pertinence des sources est déduit à partir de l'ordre de subsumption des concepts correspondants dans le treillis. Une amélioration de la méthode consiste à enrichir la requête à partir d'ontologies de domaine avant de l'insérer dans le treillis. Deux modes d'enrichissement sont possibles: l'enrichissement par généralisation et l'enrichissement par spécialisation.

1 Introduction

Suite aux progrès accomplis dans la production et l'analyse de données biologiques, un grand nombre de données est rendu accessible via le Web. Ces données sont répertoriées dans des sources biologiques offrant des interfaces d'interrogation afin de faciliter l'accès à leurs contenus. La diversité de ces sources et la complémentarité des données qu'elles contiennent permettent aux utilisateurs d'avoir des informations plus complètes. Cependant, l'absence d'un schéma unique, l'incompatibilité des formats de données et l'absence (ou la faible fréquence) de mise à jour du contenu des sources peuvent entraîner des incohérences au niveau des réponses aux requêtes posées. Face à un tel problème, il peut se révéler utile de disposer d'une classification des sources selon des informations supplémentaires permettant de juger la pertinence des sources vis à vis des requêtes. Cette classification peut être faite sur la base d'un ensemble de critères documentant le contenu et la qualité des sources et appelés méta-données. À partir de la hiérarchie de sources obtenue, nous devons être capables d'extraire les sources susceptibles de répondre au mieux à une question donnée. La méthode d'interrogation des sources doit, en outre, prendre en compte la sémantique des requêtes qu'elle traite pour améliorer les résultats de la recherche. Ainsi le problème se ramène, d'une part à l'exploitation des connaissances (méta-données) décrivant les sources disponibles sur le Web dans le but d'identifier des sources pertinentes pour une question posée et d'autre part à l'analyse sémantique de la requête en se référant à des ontologies de domaine dans le but de raffiner cette requête et d'améliorer la réponse.

2 Généralités sur les treillis de Galois

Nous ne rappelons ici qu'une définition générale des treillis de Galois. Des définitions plus détaillées peuvent être trouvées dans (Barbut 1970) et (Birkhoff 1973).

Un treillis est un ensemble ordonné tel que chaque couple d'éléments possède une borne *sup* (*supremum*) et une borne *inf* (*infimum*). Un treillis de Galois est le produit de deux treillis isomorphes appelés *treillis des extensions* et *treillis des intensions*. Les éléments du treillis sont des concepts. Un concept est formé d'un couple (*extension*, *intension*) où *extension* est un ensemble d'individus qui ont en commun l'ensemble des attributs *intension*. La relation d'ordre définie sur un treillis de Galois est appelée relation de subsumption.

3 Utilisation des treillis de Galois pour la classification et la recherche des sources de données biologiques

3.1 Construction incrémentale du treillis de Galois

En considérant la relation entre les sources et leurs propriétés comme une relation binaire entre un ensemble de départ (l'ensemble des sources) et un ensemble d'arrivée (l'ensemble des propriétés possédées par les sources), un treillis de Galois peut être construit. Nous nous retrouvons dans un cas similaire à la recherche documentaire en utilisant les treillis de Galois présentée dans (Godin et al. 1995) et (Carpineto 2000). Nous avons choisi "Incremental Structuring of Knowledge Bases" (Godin et al. 1995) comme algorithme de construction de treillis de Galois. Le choix de cet algorithme repose sur le fait qu'il offre la possibilité d'ajout de nouveaux concepts à un treillis déjà construit ce qui nous permettra d'une part l'insertion des requêtes dans le treillis pour récupérer les sources pertinentes et d'autre part l'ajout de nouvelles sources pour mettre à jour le treillis.

3.2 Recherche des sources pertinentes

Après avoir construit le treillis de Galois, un concept requête (dont l'intension est formé de l'ensemble des propriétés de la requête) est inséré dans ce treillis. L'étape suivante consiste à chercher, dans le treillis résultant, le concept le plus spécifique contenant l'ensemble des propriétés exprimées dans la requête. On notera C_R ce concept. L'extension de C_R est l'ensemble des sources ayant toutes les propriétés exprimées dans la requête. À ce niveau, on ajoute au résultat à retourner (encore vide) les sources de cette classe qu'on notera R_0 (ensemble de réponses initiales de rang 0). On considère ensuite l'ensemble des concepts parents directs de C_R dans le treillis, appelés aussi subsumants les plus spécifiques de C_R . Notons SS_1 cet ensemble. L'ensemble des réponses de rang 1, noté R_1 , est formé des sources figurant dans les extensions des concepts de SS_1 sans considérer celles appartenant à R_0 . De la même façon on détermine le reste des parties du résultat R_2 , R_3 , etc. jusqu'à atteindre le concept subsumant de C_R le plus général dans le treillis. Si l'intension de ce concept est vide alors aucune propriété n'est partagée par cet ensemble de sources avec la requête. Il n'y a donc plus de sources à ajouter au résultat d'où l'arrêt de la recherche.

4 Enrichissement sémantique de requêtes

4.1 Ontologies de domaine

La méthode de classification et de recherche de sources de données biologiques présentée ci-dessus est purement syntaxique et n'exploite pas les relations sémantiques susceptibles d'exister entre différentes propriétés. De fait dans la plupart des cas des relations sémantiques existent entre ces propriétés et sont exprimées dans des ontologies de domaine. Une ontologie est « *une spécification explicite et formelle d'une conceptualisation faisant l'objet d'un consensus* » (Gruber 1993). Elle réunit à la fois des éléments, concepts ou mots, et des règles permettant de manipuler ces éléments ou d'effectuer un certain nombre d'inférences (Berners-Lee 2001). La relation « *is-a* » définissant le lien de généralisation entre concepts est « *choisie comme relation de structuration de l'arborescence ontologique* » (Charlet et al. 2003). Nous représentons une ontologie par un arbre qu'on notera $T = (V, E)$ où V est l'ensemble des sommets de T qui représentent chacun une propriété appartenant ou non à l'ensemble des propriétés possédées par les sources et E est l'ensemble des arêtes entre les sommets de V . T est un arbre *enraciné* de racine r .

4.2 Enrichissement de la requête

Des méthodes de recherche d'information combinant les ontologies de domaine et les treillis de Galois ont été proposées dans (Priss 2000) et (Safar et al. 2004). Dans le premier cas, un thesaurus est utilisé pour enrichir l'indexation dans le treillis et améliorer le processus de recherche d'information. Dans le deuxième cas les ontologies de domaine sont utilisées pour construire un treillis raffiné selon les préférences de l'utilisateur en évitant ainsi la construction complète du treillis. Dans notre proposition, l'enrichissement consiste à ajouter à la requête utilisateur de nouvelles propriétés à partir des ontologies de domaine disponibles dans le but d'avoir un résultat plus riche. Il s'agit donc, pour une requête donnée, de considérer les propriétés figurant dans l'une des ontologies de domaine. Pour chacune de ces propriétés, on effectue un parcours de l'arbre T correspondant pour en extraire des sommets qui seront ajoutés en tant que propriétés à la requête initiale. On distingue deux types de parcours de T : un parcours par généralisation et un parcours par spécialisation. Ces deux modes correspondent respectivement à l'enrichissement par généralisation et à l'enrichissement par spécialisation.

4.2.1 Enrichissement par généralisation

L'enrichissement par généralisation consiste d'abord à localiser le sommet a de T correspondant à l'une des propriétés figurant dans la requête. Il faut ensuite parcourir le chemin de a jusqu'à la racine et ajouter à la requête les sommets rencontrés et qui appartiennent à l'ensemble des propriétés possédées par les sources. Cet enrichissement est effectué en particulier lorsque a est une feuille de T et que cette propriété n'apporte pas de source au résultat. Il permet d'obtenir une réponse enrichie par des sources plus générales que celles demandées par l'utilisateur vis-à-vis de la propriété appartenant à l'ontologie et figurant dans la requête initiale.

4.2.2 Enrichissement par spécialisation

L'enrichissement par spécialisation diffère de celui par généralisation par les sommets de T à ajouter à la requête initiale. En effet, au lieu d'ajouter les *ancêtres* du sommet a dans T qui figurent dans l'ensemble des propriétés possédées par les sources (a est le sommet qui correspond à l'une des propriétés de la requête), nous ajoutons les sommets du sous arbre $T' = (V', E')$ de T enraciné en a . Les descendants de a dans T représentent des spécialisations de la propriété représentée par le sommet a . L'ajout de ces propriétés à la requête se fait dans le but d'enrichir le résultat par des sources répondant à une partie de la requête considérée.

4.3 Apports de l'enrichissement sémantique de la requête

Après l'enrichissement de la requête, on effectue la recherche de sources pertinentes décrite dans la section 3.2 en considérant la nouvelle requête (enrichie). Le résultat de cette nouvelle recherche est plus riche en termes de nombre de sources pertinentes et l'ordre des sources dans le résultat est plus précis. En conclusion, l'enrichissement de la requête permet, en plus de l'enrichissement du résultat en nombre de sources, de réordonner plus précisément les sources selon leur pertinence et séparer éventuellement des sources qui avaient le même degré de pertinence (rang) dans le cas de la requête simple.

La pertinence des sources ajoutées découle du fait que celles-ci figurent dans le résultat grâce à une ou plusieurs propriétés ajoutées à la requête lors de l'enrichissement sémantique. Or, comme détaillé plus haut, tout ajout de propriété à la requête est dicté par une relation sémantique traduisant une certaine similarité entre les contenus des sources ayant la propriété ajoutée. De ce fait, le risque d'apparition de sources non pertinentes dans la réponse à retourner à l'utilisateur est écarté. Toutefois le degré de pertinence des sources ajoutées peut être faible.

5 Conclusion

Dans cet article nous avons étudié le problème de la classification et de l'identification des ressources biologiques pertinentes pour la réponse à une question donnée. La méthode que nous proposons s'appuie sur le fondement mathématique des treillis de Galois. Elle permet de construire une classification des sources sur la base du partage des propriétés et d'obtenir des réponses exactes lors de l'insertion d'une requête dans le treillis. Ce fondement mathématique permettra d'explorer de manière formelle le processus de recherche d'information tout en prouvant à chaque étape les résultats obtenus. La prise en compte de la sémantique valorise notre proposition vu l'intérêt grandissant accordé à cet aspect dans la majorité des domaines de recherche et notamment celui du Web sémantique.

Références

- Barbut M. et Monjardet B. (1970), *Ordre et classification : Algèbre et combinatoire* (2 tomes), Hachette, Paris, 1970.
- Berners-Lee T., Hendler J. et Lassila O. (2001), *The Semantic Web*, Scientific American, (284)5, pp 35-43, 2001.

- Birkhoff G. (1973), *Lattice Theory*. American Mathematical Society Colloquium Publications, Rhode Island, 1973.
- Carpineto C. et Romano G. (2000), Order-theoretical ranking, *Journal of the American Society for Information Science*, (51)7, pp 587-601, 2000.
- Charlet J., Bachimont B. et Troncy R. (2003), *Ontologies pour le Web sémantique*, Action spécifique 32 CNRS/STIC Web sémantique Rapport final, Vol. 2, pp 43-63, 2003.
- Godin R., Mineau G. et Missaoui R. (1995), Incremental Structuring of Knowledge Bases, *Proceedings of the 1st International Symposium on Knowledge Retrieval, Use, and Storage for Efficiency (KRUSE'95)*, Santa Cruz (CA), USA, pp 179-193.
- Godin R., Mineau G., Missaoui R. (1995), Méthodes de classification conceptuelle basées sur les treillis de Galois et applications, *Revue d'intelligence artificielle*, (9)2, pp 105-137, 1995.
- Gruber T.R. (1993), A translation approach to portable ontology specification, *Knowledge Acquisition*, (5)2, pp 199-220, 1993.
- Priss U. (2000), Lattice-based Information Retrieval, *Knowledge Organization*, (27)3, pp 132-142, 2000.
- Safar B., Kefi H. et Reynaud C. (2004), *OntoRefiner : a user query refinement interface usable for Semantic Web Portals*, *Proceedings of the ECAI 2004, Workshop on Application of Semantic Web Technologies to Web Communities*, Valencia, Spain.

Summary

In this paper we present a method of biologic data sources classification and retrieval. It consists in building the Galois lattice of the binary relation between a set of data sources and their metadata converted to boolean properties. A Query concept can then be inserted into the Galois lattice using an incremental algorithm. The result is built with the set of data sources in the extensions of the query subsumers in the resulting Galois lattice. The ranking in the result is given by the subsumption order between the corresponding concepts in the lattice. The use of domain ontologies to refine the query before inserting it into the Galois lattice improves the retrieval performances. Two types of query refinement are possible : generalisation refinement and specialisation refinement.