

Construction et manipulation d'ontologies multi-lingue, sémantique et dynamique dans Amine Platform

Karim Bouzouba

*IERA, Université Mohamed V Souissi,
Rabat, Maroc, B.P 6216
karim.bouzoubaa@iera.ac.ma*

Adil Kabbaj

*INSEA,
Rabat, Maroc, B.P. 6217
akabbaj@insea.ac.ma*

Résumé

Disposer d'un environnement de développement intégré est un souci constant en Intelligence Artificielle. Par ailleurs, suite à l'utilisation croissante d'ontologies dans les applications intelligentes, un tel environnement est appelé à intégrer non seulement les outils de développement classiques mais aussi des outils répondant aux utilisations variées des ontologies: utilisation d'une ontologie existante, création et mise à jour de sa propre ontologie, utilisation d'ontologies monolingue et multi-lingue et formation dynamique d'une ontologie.

Le projet Amine Platform vise à offrir un tel environnement. Il comprend des outils de définition et de manipulation d'ontologies, un langage de programmation logique et conceptuel et un module de développement utilisant le paradigme multi-agents.

Cet article présente brièvement cette plate-forme, en mettant l'accent sur sa composante ontologie.

Mots clés

Intelligence Artificielle, Environnement de développement intégré, Modélisation et représentation des connaissances, Ontologie Multi-lingue, Formation dynamique d'une ontologie.

1. INTRODUCTION

L'ontologie est composée de concepts qui décrivent des objets concrets ou abstraits censés modéliser une partie de l'univers [3]. Plusieurs disciplines, informatiques et autres, se penchent sur l'étude et l'utilisation d'ontologies. Dans le cas de bases de données distribuées et hétérogènes, l'utilisation d'une ontologie est primordiale afin que ces bases puissent s'échanger les données. Dans un système de traitement du langage naturel, l'usage d'une ontologie est nécessaire pour les divers niveaux de traitement (lexicale, morphologique, etc.). Citons à titre d'exemple le cas de WordNet, une ontologie lexicale comprenant noms, verbes, adjectifs, et adverbes de la langue Anglaise organisée sous un ensemble de synonymes [5]. L'ontologie est également très importante pour les cas du web sémantique. En effet, un web sémantique est une extension du web courant où chaque information détient un sens unique et bien définie et ce afin de permettre aux internautes et aux machines de coopérer [2].

Pour les applications en Intelligence Artificielle (IA), on constate un besoin grandissant et varié concernant l'utilisation des ontologies :

- utilisation d'une ontologie existante comme le cas de Cyc [10]
- création et mise à jour de sa propre ontologie
- utilisation d'ontologies monolingue et multi-lingue
- formation incrémentale et automatique d'une ontologie
- intégration d'une ontologie dans un environnement de développement dédié à l'IA

Le projet Amine Platform vise à répondre à ces besoins [1]. C'est un environnement de développement d'applications intelligentes offrant un outil de définition et de manipulation d'ontologies, un langage de programmation logique et conceptuel et un outil de développement utilisant le paradigme multi-agents.

Dans le cadre de cet article, nous nous concentrons sur l'outil de manipulation d'ontologies. Cet outil permet la création, l'édition et la construction d'une ontologie *sémantique*, *multi-lingue* et *dynamique*. Une ontologie dans Amine Platform est sémantique dans la mesure où les mots sont associés à des concepts définis par des structures conceptuelles. Une ontologie dans Amine Platform est multi-lingue puisqu'un même concept peut être référé dans plusieurs langues à la fois. L'ontologie dans Amine Platform peut être dynamique car cette dernière offre un mécanisme automatique d'intégration de nouvelles informations (concepts et structures conceptuelles) impliquant une réorganisation et une mise à jour automatique de l'ontologie. Cette réorganisation se base sur le modèle de la mémoire basé sur la généralisation [7].

Pour la représentation des connaissances, nous utilisons le formalisme des Graphes Conceptuels (GC), développé initialement par John Sowa [12, 13], et utilisé actuellement dans différentes applications IA. La théorie des GCs est un système de logique basé sur les graphes existentiels de

Peirce [11] et les réseaux sémantiques. Un GC est un graphe (un réseau) de concepts reliés par des relations conceptuelles. Par exemple, le sens de la phrase "le chat mange la souris" peut être représenté par le GC suivant (forme textuelle), composé de trois concepts ([Manger], [Chat], [Souris]) et deux relations (agnt, obj) :

```
[Manger]-  
  -agnt->[Chat];  
  -obj->[Souris]
```

le GC peut être lu de la manière suivante: "l'agent de Manger est un chat et son objet est une Souris".

Le reste de l'article se présente comme suit. La seconde section présente brièvement l'architecture générale de Amine Platform, composée de trois couches principales. Ensuite, nous détaillons le niveau sémantique d'une ontologie avec les GCs comme formalisme de représentation des connaissances. La section 4 expose la manière avec laquelle une ontologie supporte divers langages à la fois. Dans la section 5, nous introduisons le caractère dynamique d'une ontologie dans Amine Platform.

2. L'ARCHITECTURE DE LA PLATFORME

Amine platform est une suite logicielle "Open Source" basée sur la théorie des GCs. La platform est développée en Java et a été conçue de la manière la plus modulaire possible. Amine platform est composé de trois couches logistiques superposées, la première constitue le noyau, la seconde offre les composants principaux (langages de programmation et mécanismes divers) et la troisième implémente les concepts d'agents et systèmes multi-agents.

2.1 La première couche

La première couche correspond au noyau de la platform (Figure 1). Elle comprend des APIs pour les GCs, les notations des GCs (forme linéaire, forme CGIF et forme graphique), les opérations sur les GCs et l'ontologie. Cette couche offre aussi trois interfaces :

1. Une interface multi-documents, multi-lingue et multi-formes pour l'édition des GC ;
2. Une interface pour la création, l'exploration et la mise et jour d'une ontologie ;
3. Une interface pour la mise en œuvre des opérations sur les GC (jointure maximale, généralisation et subsomption).

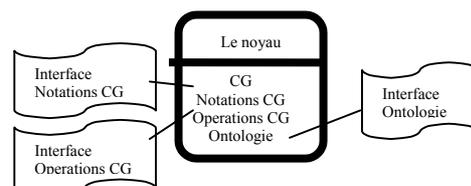


Figure 1 : La première couche de Amine Platform

2.2 La seconde couche

La seconde couche de Amine Platform concerne trois composantes : le langage Prolog+CG, un mécanisme de formation dynamique de l'ontologie et le langage Synergy. Prolog+CG [7, 9] est une extension conceptuelle basé objet du langage Prolog (un langage pour la programmation en logique). En effet, en plus des structures usuelles de Prolog, Prolog+CG incorpore les GCs et les opérations sur les GCs. Le moteur de formation dynamique de l'ontologie permet l'intégration automatique de l'information dans l'ontologie [6, 7]. Synergy est un langage pour la programmation visuelle et la programmation par activation des GCs [7, 8]. Bien sûr, la seconde couche se base sur la première ; dans l'utilisation et la manipulation des GCs, de l'ontologie et des opérations sur les GC.

2.3 La troisième couche

La troisième couche concerne le niveau Systèmes Multi-agents [4]. Cette couche sera intégrée dans les versions ultérieures de Amine Platform.

3. UNE ONTOLOGIE BASEE SUR LES GC

Une ontologie dans Amine Platform peut être vue comme une base de connaissances utilisant les CGs. Une ontologie est composée d'une hiérarchie de types enrichie de structures conceptuelles (définitions, canons, situations et individus). Ces dernières sont spécifiées sous forme de GCs. La relation "plus général que" indique l'ordre dans la hiérarchie de types. Ainsi, la signification d'un concept est spécifiée par :

- Ces super-types et sous-types immédiats
- La définition du concept
- La spécification d'un canon qui décrit les conditions minimales pour assurer une bonne utilisation du concept
- La spécification de situations où le concept est utilisé

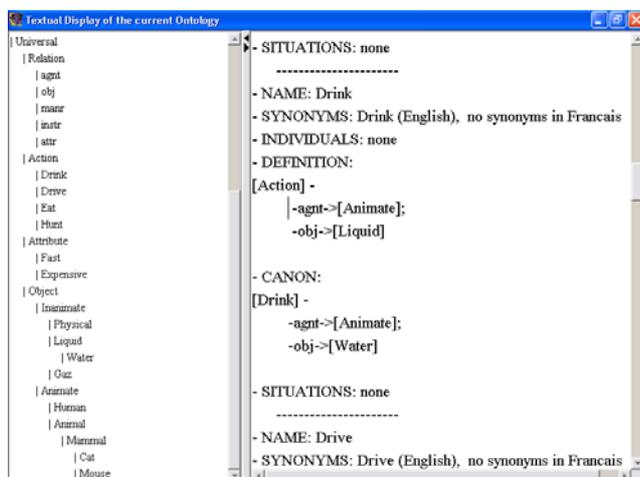


Figure 2 : Exemple d'ontologie

Par exemple, considérons l'ontologie décrite par la figure 2. Le concept "Drink" :

- Est Une action
- Sa définition est "Action faite par un agent animé et dont l'objet est un liquide", représentée par le GC

```
[Action]-
  -agnt->[Animate];
  -obj->[Liquid]
```

- Une situation peut être "L'humain boit de l'eau", représentée par le GC

```
[Drink]-
  -agnt->[Person];
  -obj->[Water]
```

4. UNE ONTOLOGIE MULTI-LINGUE

Une ontologie dans Amine Platform peut être multi-lingue, permettant ainsi à plusieurs personnes, utilisant des langages différents, de partager une même ontologie. En terme technique, ceci est rendu possible grâce à la séparation entre les éléments conceptuels (types conceptuels, référent, type relationnel) et leurs identificateurs dans un langage spécifique. Le multilinguisme est une des caractéristiques importantes de Amine Platform. Ainsi, cette dernière fournit une traduction automatique des concepts et du sens des structures conceptuelles.

De plus, l'utilisateur peut spécifier différents synonymes pour un même concept, soit dans le même langage ou dans d'autres langages.

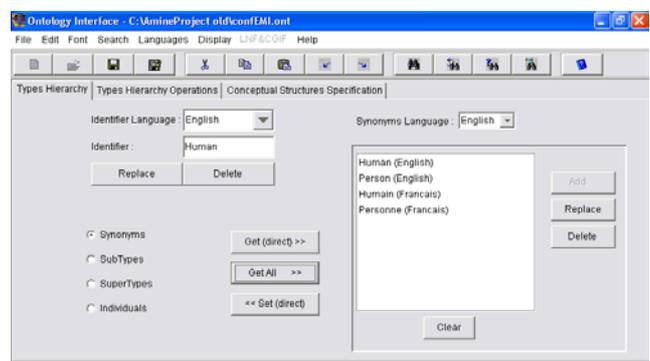


Figure 3: Synonymes du concept 'Human'

La figure 3 présente par exemple un écran où l'utilisateur peut spécifier les synonymes d'un certain concept. Dans ce cas de figure, il s'agit du concept 'Human' (exprimé en Anglais), de son autre synonyme en Anglais 'Person' et de ses deux synonymes en Français 'Humain' et 'Personne'.

Après avoir spécifié les synonymes des concepts utilisés dans l'ontologie, l'utilisateur est en mesure d'obtenir la traduction automatique des GC exprimés.

Par exemple, dans la figure 4, l'utilisateur spécifie une situation pour le concept 'Man' en Anglais (onglet 'English' est sélectionné) :

```
[Drink]-
  -agnt->[Human];
  -obj->[Water]
```

Par un simple click sur l'onglet 'Français', l'utilisateur obtient la traduction de la même situation en Français (Figure 4):

```
[Boire]-
- agnt->[Humain];
- obj->[Eau]
```

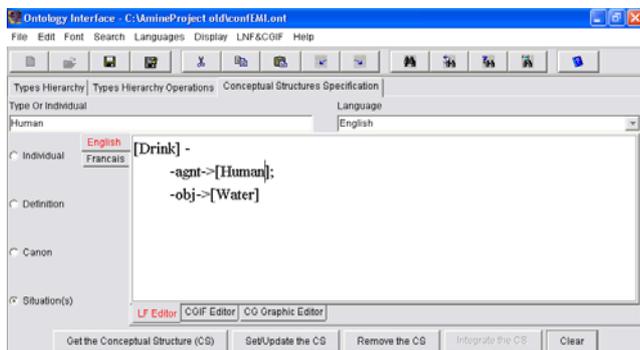


Figure 4 : Une situation du concept 'Human' en Anglais

5. UNE ONTOLOGIE DYNAMIQUE

L'utilisateur peut utiliser l'interface de l'ontologie pour créer et mettre à jour manuellement son ontologie. Il peut utiliser aussi le mécanisme de construction dynamique de l'ontologie pour intégrer automatiquement une nouvelle information dans l'ontologie. Comme la plupart des algorithmes de classification [8], la formation d'une ontologie par intégration développe un graphe de généralisation et est concernée par le stockage, la recherche et l'utilisation de la connaissance contenue dans l'ontologie. Elle se caractérise toutefois par les points suivants :

- *Différents types d'informations à intégrer* : une nouvelle structure à intégrer peut correspondre à la définition d'un concept, à la description d'un individu, à la description d'un canon ou situation et enfin à la description d'un synonyme pour un concept ;
- *Opération de comparaison* : alors que la plupart des algorithmes de classification utilisent l'opération de subsomption structurale, la formation par intégration utilise une opération plus générale qui tient compte des niveaux structural et sémantique des structures à comparer et qui tient compte également des différents cas de figure qui peuvent résulter de la comparaison (la nouvelle structure peut être "plus général que", "identique à", "plus spécifique que", "partage des éléments en commun" ou "n'a rien en commun" avec une structure existante de l'ontologie) ;
- *Stratégie d'intégration* : différentes stratégies ont été proposées pour la classification, notre algorithme d'intégration se base sur une stratégie qui lui est propre;
- *Notion de focus* : l'algorithme utilisé est caractérisé par le fait qu'il tienne compte de la notion de focus (certains éléments d'une structure conceptuelle sont plus importants que d'autres) et utilise une forme de compression et d'explicitation des connaissances.

Le processus d'intégration le plus intéressant concerne celui d'une nouvelle situation ou définition. Après avoir

donné la main à l'utilisateur pour déterminer les concepts qu'il considère pertinents, ces concepts deviennent des focus de la nouvelle situation. Le processus d'intégration consiste alors à propager cette nouvelle situation dans toute l'ontologie mais en prenant comme point d'entrée les éléments spécifiés comme focus. Le processus commence par localiser le concept focus dans l'ontologie et propage ensuite la nouvelle situation à travers les fils du focus en question. Arrivé à un nœud donné, le processus compare (comparaison en termes de GCs) la nouvelle situation avec ce nœud. Le résultat de la comparaison renseigne sur la manière avec laquelle la mise à jour de la mémoire doit s'effectuer. Plusieurs cas de comparaison existent. Par exemple, si la nouvelle situation et le nœud courant ont des éléments en commun, un nouveau nœud est alors créé contenant les informations en commun et devient père aussi bien de la nouvelle situation que du nœud courant. Les détails du processus d'intégration sont fournis dans [7].

5.1 Exemple d'intégration dynamique

A partir du contenu de l'ontologie décrite à la figure 2 (et leurs synonymes en Français), l'utilisateur demande au module d'intégration dynamique d'intégrer les deux situations suivantes: S1 [Mammifère] <- agnt- [Manger] et S2 [Mammifère] -obj-> [Manger] avec comme focus les concepts Mammifère et Manger. S1 et S2 deviennent alors deux situations filles (descendants directs) de ces deux concepts (Figure 5).

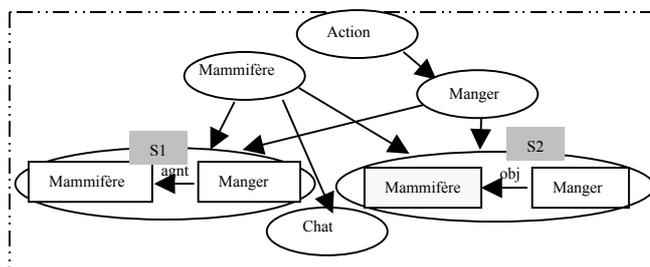


Figure 5 : Intégration dynamique de S1 et S2

L'utilisateur demande ensuite au module d'intégrer S3 [Chat] <- agnt- [Manger] -obj-> [Souris] avec comme focus Manger et Chat. En propageant S3 à partir de Manger, le module compare S3 à S1 et à S2. S3 étant plus spécifique que ces deux dernières, elle en devient situation fille (Figure 6). En propageant S3 à partir de Chat, elle devient simplement une situation fille.

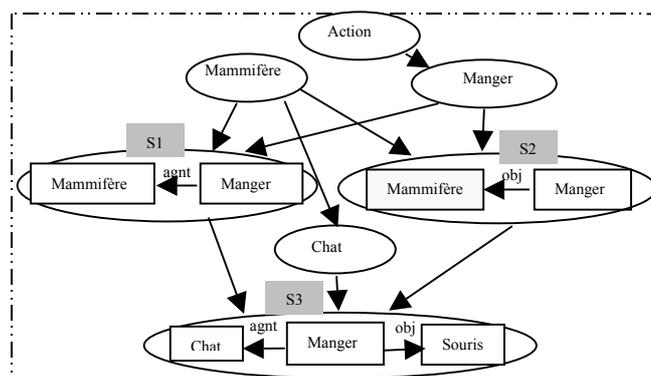


Figure 6 : Intégration dynamique de S3

Enfin, considérons l'intégration de S4 [Chat]<-agnt-[Chasser]-obj->[Souris] avec comme focus Chat et Chasser. Le module propage S4 aux descendants directs de Chat, en l'occurrence S3. Après comparaison de S4 avec S3, il constate qu'elles ont des éléments en commun. Le module crée alors un nouveau nœud S5 dont le contenu est le résultat de la comparaison : [Chat]<-agnt-[Action]-obj->[Souris]. Etant donné que S5 est une généralisation de S3 et S4, l'ontologie est réorganisée en conséquence (Figure 7). Finalement, à partir de Chasser, S4 devient simplement situation fille.

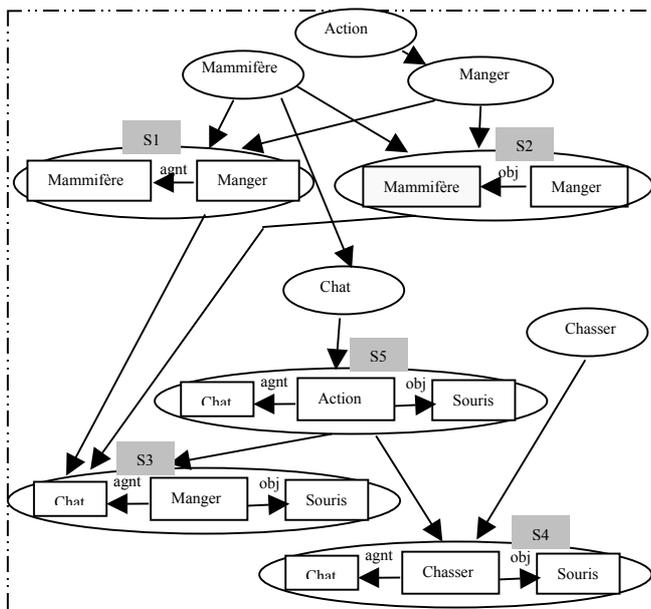


Figure 7 : Intégration dynamique de S4

Nous avons illustré à travers cet exemple un cas d'intégration dynamique de connaissances dans l'ontologie. En effet, l'exemple montre comment le processus d'intégration (par propagation) conduit à la formation incrémentale et à la réorganisation de l'ontologie.

6. CONCLUSION

Les applications en IA présentent divers besoins d'utilisation du concept d'ontologie à savoir l'utilisation d'une ontologie existante, la création et la mise à jour de sa propre ontologie, le besoin d'utiliser aussi bien des ontologies monolingue que multi-lingue et enfin la possibilité d'une formation dynamique et incrémentale d'une ontologie. Le présent article illustre comment la composante ontologie de l'environnement de développement Amine Platform répond à ces besoins.

BIBLIOGRAPHIE

[1] AminePlatform-site : www.insea.ac.ma/CGTools/AminePlatform/AminePlatform.htm
 [2] Berners-Lee T. et al, 2001, *The Semantic Web*, Scientific American.

[3] Biébow B., Szulman S., 1999, Une approche terminologique pour catégoriser les concepts d'une ontologie, in *Ingénierie des connaissances, évolutions récentes et nouveaux défis*, Charlet et al. Ed.
 [4] Bouzouba K, 1998, Modélisation des interactions multi-agents basée sur le point de vue interne, Ph. D. Thesis, Univ. Laval, Canada.
 [5] George A. et al, 1990, Introduction to WordNet: an on-line lexical database, In *International Journal of Lexicography* 3 (4), pp. 235 - 244.
 [6] Kabbaj A., 1995, Self-Organizing Knowledge Bases: The Integration Based Approach, in the Proc. Of the Int. KRUSE Symposium: Knowledge Retrieval, Use, and Storage for Efficiency, Santa Cruz, CA, USA, p. 64-68.
 [7] Kabbaj A., 1996, Un système multi-paradigme pour la manipulation des connaissances utilisant la théorie des graphes conceptuels, Ph. D. Thesis, Univ. De Montreal, Canada.
 [8] Kabbaj A., 1999, Synergy: a conceptual graph activation-based language, in Proc. Of the *International Conference on Conceptual Structures ICCS'99*.
 [9] Kabbaj A. et al, 2001, Uses, Improvements and Extensions of Prolog+CG: Case studies, in Proc. Of the *International Conference on Conceptual structures ICCS'01*, San Francisco, August.
 [10] Lenat, D. B., 1995, Cyc: A Large-Scale Investment in Knowledge Infrastructure, Communications of the ACM 38, no. 11.
 [11] Peirce C., 1960, *Collected Papers of Charles Sanders Peirce*, edited by A. W. Burks, Harvard University Press, Cambridge.
 [12] Sowa J., 1984, *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, USA.
 [13] Sowa J., 1999, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole Publishing Co, Pacific Grove, USA.