Modélisation de l'Hétérogénéité Sémantique entre Agents Dialogiques en Logique de Descriptions

Erika Valencia Jean-Paul Sansonnet
LIMSI-CNRS, Université Paris-XI, Orsay
{erika, jps}@limsi.fr

Dans le cadre de l'étude de la communication entre agents logiciels, nous nous intéressons à l'interaction dialogique entre agents. Cet article présente les problèmes d'hétérogénéité sémantique qui surgissent entre agents dialogiques, et en propose une modélisation à l'aide de la logique des descriptions. Les agents sont ainsi vus comme des bases de connaissances, et l'hétérogénéité sémantique se ramène alors soit à une hétérogénéité terminologique, soit à une hétérogénéité référentielle.

Mots-clé : Dialogisme, Hétérogénéité Sémantique, Base de Connaissances, Terminologie, Adaptation.

1 L'hétérogénéité Sémantique entre Agents Dialogiques

1.1 Motivations

Le paradigme des systèmes multi-agents est de plus en plus utilisé pour décrire, étudier et développer des applications faisant entrer en interaction différents *composants logiciels*. En effet, au delà de la simple gestion de bases hétérogènes, il y a un besoin grandissant de prendre en compte les aspects dynamiques et évolutifs des applications, et donc de les considérer comme des *agents* disposant d'un niveau de représentation des connaissances et de capacités de raisonnement pour entrer en interaction avec leur environnement.

Notre étude se place dans le domaine des recherches sur la *communication* entre agents logiciels. Notons que dans ce cadre, un premier pas avait été établi avec la proposition des actes de discours [22], et les langages d'interaction qui s'en inspirent comme KQML [9]. Cependant, afin de ne pas réduire l'interaction à un médium de transport d'information "à la Shannon", la *conversation* [24] a ensuite été considérée comme une suite d'interactions co-construites et non pas indépendantes. C'est dans cette lignée que s'est inscrite l'introduction de la notion de dialogisme [12].

C'est précisément dans ce contexte que se situe le projet InterViews [20], qui vise à mettre en interaction dialogique des agents logiciels vus comme des applications dynamiques intégrées dans des pages HTML, via Internet. L'idée est de doter chaque application de capacités de traitement symbolique leur permettant d'avoir une représentation interne ainsi que des capacités de raisonnement sur leur fonctionnement et celui des autres agents.

1.2 Dialogisme

Avant de considérer l'interaction hétérogène entre agents dialogiques, nous définissons le dialogisme. Selon [12], le dialogisme doit être considéré comme présentant deux principales caractéristiques :

Un potentiel de sens Le *sens* est co-construit par les inter-actants. Un message n'a pas un sens *a priori*, mais seulement un potentiel de sens. En effet, l'agent émetteur exprime un message porteur d'une signification locale, mais qui n'est pas universellement définie. Le récepteur doit alors en trouver une *interprétation* selon ses connaissances locales sur lui-même, sur l'émetteur et sur le contexte. Le sens interprété par le récepteur peut alors être différent de celui que l'émetteur avait affecté à son message.

Un suivi de conversation C'est le mécanisme qui permet de maintenir la compréhension dans l'interaction malgré les différences potentielles de sens accordées au même message par deux inter-actants. En effet, l'émetteur doit s'assurer que le sens qu'il a affecté à son message et celui qui a été trouvé par le récepteur sont compatibles. Il s'agit alors pour les inter-actants de détecter les incohérences, révélatrices d'incompréhensions.

1.3 Hétérogénéité Sémantique entre Agents Dialogiques

Dans *World and Object*, Quine [19] présente le problème de *Gagavaï*. Ce problème résulte de la *rencontre* d'un ethnologue avec une tribu dont le langage lui est inconnu. Si un natif dit "Gagavaï" tout en pointant un lapin blanc qui court, il n'y a pas de moyen immédiat pour l'ethnologue de savoir ce que Gagavaï veut dire. En effet, cela pourrait vouloir dire blanc, lapin, courir, partir, ou blanc lapin, ou faim, des centaines d'autres choses. Notre étude se concentre sur la gestion de ce type d'hétérogénéité, due au phénomène de potentiel de sens : c'est ce que nous appelons l'hétérogénéité sémantique.

La gestion du suivi de conversation peut être un élément utile à la résolution de l'hétérogénéité sémantique, comme par exemple dans le jeu du "guessing game" du projet Talking Heads [23].

Les agents considérés dans InterViews sont des applications logicielles développées par des utilisateurs différents dans des buts différents avec des *termes* différents. Ainsi, les agents qui en résultent n'auront pas le même vocabulaire, ni le même bagage de sens. L'hétérogénéité sémantique est donc un aspect crucial à traiter dans ce projet. Nous ne désirons en aucun cas gommer cette hétérogénéité en nous ramenant à une homogénéité entre agents, comme ce serait le cas dans l'optique d'une intégration de bases hétérogènes [28].

Un agent dialogique va donc émettre des messages, que nous qualifierons dorénavant de requêtes, qui utilisent des connaissances locales contenues dans sa propre base de connaissances. Nous dirons que la requête émise doit être *cohérente* avec la base de l'envoyeur.

Nous modélisons les agents A_i dans InterViews comme des agents computationnels composés de :

- Une *ontologie* o_i : qui est une spécification explicite d'une *conceptualisation* [11]. Une conceptualisation est une *abstraction* à l'aide de concepts de l'univers connu de l'agent. Elle est une *spécification explicite* en ce qu'elle est représentée à l'aide d'un formalisme de représentation des connaissances.
- Une *mécanique* de fonctionnement f_i : qui est un ensemble de données *exécutables*.
- Un *comportement* c_i : qui est la *manifestation* de l'exécution des procédures de l'agent en utilisant les connaissances représentées dans l'ontologie.

Nous allons étudier les phénomènes d'hétérogénéité sémantique au niveau des définitions de sens pour les termes connus par les agents, c'est-à-dire entre les bases de connaissances ontologiques. Ainsi, pour traiter les problèmes d'hétérogénéité sémantique (HS) entre agents dialogiques (AD), nous proposons de les considérer au niveau de la *représentation* des *bases* de *connaissances* (KBase) des agents [27].

1.4 Modélisation en Logique de Descriptions

Il apparaît que la Logique des Descriptions (LD) [8], [18] est adaptée pour traiter les problèmes de gestion de *requêtes* entre deux représentations des connaissances sémantiquement hétérogènes. En effet, les LD sont déjà considérées comme un candidat naturel pour décrire les schémas en bases de données et les questionner à l'aide de requêtes.

De plus, elles sont étudiées en tant que formalisme de représentation pour les systèmes de gestion de bases de données [21]. Enfin, nous savons que les LD sont aussi adaptées lorsque les requêtes sont formulées par une personne qui n'est pas familière avec la base interrogée ou bien par une personne qui ne sait pas bien quelles questions sont appropriées [1].

Nous commençons donc par formaliser les notions d'agents dialogiques et d'hétérogénéité sémantique. Nous examinons ensuite les hypothèses nécessaires pour considérer l'hétérogénéité sémantique. En effet, si l'hétérogénéité est trop radicale, l'interaction n'aura pas lieu [6], et à l'opposé, si l'hétérogénéité est trop faible, il n'y a plus d'hétérogénéité à "gérer".

Ces hypothèses sont proches de celles imaginées par Minsky [17], lorsqu'il imagine un cas de *rencontre* entre deux "aliens". Tout d'abord, on ne pourra s'intéresser à l'interaction que si l'on suppose que les deux aliens sont *motivés* par cette interaction. Par ailleurs, Minsky fait une seconde hypothèse concernant *l'environnement physique* duquel est forcément issu cet alien. Il propose de considérer qu'il est soumis aux mêmes *contraintes* aux limites en termes de *spatialité*, de *temps* et de contraintes matérielles : les deux aliens sont contraints par leur *corporalité*. Nous dirons donc que les agents doivent partager un domaine. Enfin, l'alien devra donc posséder un moyen de *représenter* les situations physiques et qu'il devra posséder des moyens de *manipuler* cette représentation.

Nous dégageons ensuite deux types de HS entre AD modélisées en LD, la HS terminologique et la HS référentielle. Dans ce document, nous nous focalisons sur la HS terminologique, et nous proposons de la résoudre avec des outils d'adaptation issus d'une modélisation du raisonnement par analogie. Nous illustrons cette proposition par les algorithmes appropriés et leur application sur un exemple.

1.5 Outils de Logique de Descriptions

Une des principales caractéristique des LD est de *séparer* les connaissances *terminologiques* concernant les concepts (dans une TBox), des connaissances *assertionnelles* concernant les individus effectifs du monde (dans une ABox). Cette distinction présente des avantages du point de vue de la représentation des connaissances et de la complexité. De plus, les logiques de descriptions possèdent une sémantique dénotationnelle propre, qui va nous permettre de formaliser la notion d'hétérogénéité au niveau sémantique entre les agents.

Les LD sont des descendantes du formalisme SIN, introduit par Brachman en 1978 [3] pour donner une sémantique formelle aux formalismes issus des réseaux sémantiques. Nous utilisons dans ce document le langage terminologique appelé TF, qui est assez proche du langage introduit par Brachman et Levesque en 1984 [4], il est tiré de la thèse de Nebel [18].

Par exemple, si nous voulons écrire la terminologie suivante :

- Un Homme est un Humain
- Une Femme est un Humain
- Aucun Homme n'est une Femme et vice versa
- Une Équipe est (définie comme) un Ensemble avec au-moins 2 membres qui sont des Humains
- Une Petite-Equipe est (déf.comme) une Equipe avec au-plus 5 membres

Voici les concepts primitifs à introduire (avec la syntaxe présentée en annexe, table 1 en notations abstraites) :

$Humain \leq \top$

$Homme \leq Humain$

$Femme \leq Humain$

$Ensemble \leq \top$

Les deux rôles dont il est question dans cet exemple sont celui de membre et de chef :

$chef \leq membre$

Et nous introduisons enfin les concepts atomiques plus complexes par des équations. Une équipe est un ensemble mais moyennant certaines restrictions. Une restriction de valeur concernant le rôle *membre* :

$Equipe \doteq (Ensemble \wedge (\forall membre : Humain)(\geq 2\ membre))$

Une petite équipe est alors une équipe moyennant une restriction supplémentaire : il faut au maximum 5 membres :

$PetiteEquipe \doteq (\ Equipe \wedge (\leq 5\ membre))$

2 Définition de l'Hétérogénéité Sémantique en Logique de Descriptions

2.1 Agents Dialogiques

Pour résoudre l'hétérogénéité au niveau sémantique entre des agents dialogiques, nous nous intéressons à la partie qui *définit* la sémantique des *termes* connus de l'agent : l'ontologie o_i des agents. Ces ontologies sont modélisées par des bases de connaissances, notées KB_i (pour "knowledge base"). Ces bases de connaissances sont écrites en logique des descriptions, dont nous connaissons la sémantique formelle (voir annexe).

Dans ce document, chaque agent dialogique A_i est circonscrit à la Kbase :

$$KB_i = \{Tb_i, Ab_i\} \text{ c}$$

qui modélise son ontologie, avec Tb_i qui représente la Tbox, et Ab_i qui représente la Abox de la KBase.

A partir de cette hypothèse et en utilisant la logique de descriptions présentée ci-dessus, nous allons formaliser toutes les hypothèses du dialogisme, puis définir ce que nous entendons par hétérogénéité sémantique (HS) entre agents dialogiques dans InterViews. Nous dégageons alors deux types de HS, la HS terminologique et la HS référentielle. Nous terminons par une proposition, qui consiste à utiliser les outils du raisonnement par analogie pour résoudre ces deux types de HS.

2.2 Interaction entre Agents

2.2.1 Pas d'Interaction

Une interaction I est une séquence finie de n pas d'interaction I^i, elle est notée :$I = [I_1, ..., I^n]$ où $n \in N$ avec $n < \infty$

Un pas d'interaction I^i entre deux Kbases KB_1 et KB_2 est le couple $I^i = \{Q^i, R^i\}$ formé d'une requête Q^i et sa réponse *associée* R^i.

La KBase qui *émet* la requête, est qualifiée *d'envoyeur*. La KBase qui *reçoit* la requête est qualifiée de *receveur*. Elles sont notées respectivement KB_e et KB_r.

2.2.2 Requête et Réponse

Dans notre modélisation en LD du problème, la requête et la réponse sont des *expressions* de logique des descriptions, tout comme les descriptions dans les KBases modélisant les agents. Cependant, la requête contient en plus des *patterns* pour questionner à propos d'une connaissance. Ces patterns sont exprimés à l'aide de variables libres [2]. Si bien qu'une requête Q est une expression de LD utilisant l'alphabet usuel de la LD, plus trois symboles de pattern pour désigner les question à propos des individus (pattern $\#x?$), à propos des concepts (pattern $X?$), ou des rôles (pattern $x?$).

Enfin, d'après la définition donnée aux requêtes dialogiques dans InterViews, une KBase ne peut formuler des requêtes que dans le vocabulaire qu'elle connaît. Le vocabulaire connu par un agent A_i est celui qui est défini dans la terminologie Tb_i de la KBase KB_i qui le modélise. Ainsi, à un pas d'interaction quelconque i, l'envoyeur KB_e ne peut formuler vers le receveur KB_r, que les requêtes $Q^i_{e \to r}$ qui sont *consistantes* avec sa *propre* terminologie, ce qui se note : $KB_e \models Q^i_{e \to r}$.

2.3 Hétérogénéité Sémantique

Une interaction I entre deux Kbases KB_1 et KB_2 est qualifié *d'hétérogène* si la requête produite par KB_1 est *sémantiquement hétérogène* par rapport à KB_2.

Considérons la définition que Weinstein [27] propose pour l'hétérogénéité sémantique pour une terminologie unique : 1) la terminologie comprend des symboles qui ont des dénotations *inconsistantes* le même symbole référence des objets différents ; 2) la terminologie comprend des dénotation *redondantes* : des symboles différents référencent le même objet ; 3) la terminologie comprend les *deux*.

L'interaction sera donc qualifiée de HS dès lors qu'elle fait intervenir deux KBase hétérogènes sémantiquement au sens de Weinstein. Qu'une requête $Q^i_{1 \to 2}$, considérée à un pas d'interaction i particulier, soit hétérogène ou pas, n'est pas pertinent. Une interaction hétérogène peut être constituée de requêtes qui ne sont pas toutes strictement hétérogènes au sens de Weinstein. Mais nous verrons que le fait que les deux Kbases soient hétérogènes, aura des conséquences sur la gestion des requêtes par la KBase receveur, même si la requête elle -même n'est pas strictement hétérogène au pas i.

Cette inconsistance de $KB_{1 \cup 2}$ aura lieu ssi les Kbases :

1. sont *différentes :* $KB_1 \neq KB_2$
2. et qu'elles *partagent* un domaine.

Ces deux conditions impliquent une *intersection* non nulle entre les domaines d'interprétation des Kbases KB_1 et KB_2 :

$$\epsilon_1(KB_1) \cap \epsilon_2(KB_2) \neq \emptyset$$

où $\epsilon_i(KB_i)$ désigne le domaine d'interprétation de la Kbase.

2.4 Échec d'une Interaction Sémantiquement Hétérogène

Les problèmes sémantiques peuvent être classés en deux catégories, selon qu'ils sont *détectés* ou non par l'agent qui en est à l'origine [14] :

Le Not-Understanding Problem (NUP) est l'échec à un pas d'interaction i du receveur KB_r à trouver *une* interprétation *valide* à la requête $Q^i_{e \to r}$. Comme nous l'avons dit, il se peut qu'il n'en trouve *aucune* ou bien qu'il en trouve *plusieurs* et qu'il soit incapable de décider laquelle est valide. Dans tous les cas le participant qui échoue, en est conscient : le problème est *détecté* par lui.

Il se formalise par :

$$NUP(i) ::= Card\{\epsilon_r(Q^i_{e \to r})\} \neq 1$$

où ϵ_r est la fonction d'interprétation de KB_r.

Le Misunderstanding Problem (MUP) arrive quand un participant KB_e ou KB_r trouve *une* interprétation qu'il considère comme valide par rapport à sa KBase locale, pour la requête $Q^i_{e \to r}$. Si la validité de cette interprétation doit être remise en cause par l'envoyeur de la requête KB_e, cela ne pourra se faire que si KB_e détecte une anomalie. Mais cela ne sera détectable que si cela se traduit par un échec à un pas d'interaction ultérieur, j ; c'est-à-dire si le MUP a engendré un NUP. Cela se formalise par :

$$MUP(i) ::= \exists j > i \mid NUP(j)$$

Dans ce cas, nous sommes ramenés à traiter le NUP à un pas i.

2.5 Gérer une Interaction Sémantiquement Hétérogène

2.5.1 Notion de Point de Vue pour la Gestion

L'hétérogénéité sémantique entre deux Kbases KB_1 et KB_2 peut se résoudre selon deux approches :

- Utilisation d'un point de vue global sur les deux KB_i ;
- Utilisation d'un point de vue local dans chaque Kb_i.

2.5.2 Point de Vue Global

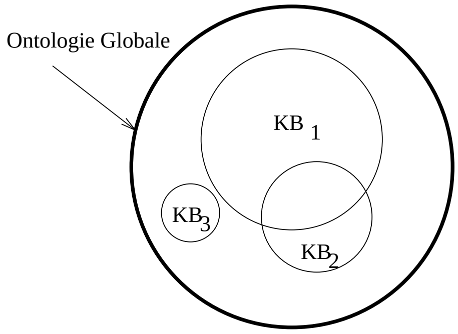
Un point de vue *global* est un point de vue *externe et unique* sur les deux Kbases à la fois. Dans ce type de point de vue, on "gomme" la HS en la ramenant à de *l'homogénéité* : on *créer* un Kbase *abstraite*, plus *large*, *englobant* de manière cohérente les deux Kbases des agents. Cette KBase, notée $KB_{(1,2)}$, est qualifiée de *globale*, par opposition aux Kbases KB_1 et KB_2 qui sont alors qualifiées de *locales*. La Kbase globale peut être : construite, ce qui correspond à la proposition d'une ontologie globale [15], ou bien simulée par le biais d'un dictionnaire, ce qui correspond à la proposition *d'agents médiateurs* [5]. Notons que dans le cas d'un point de vue global, la HS est *résolue*. En effet, la mise en oeuvre du point de vue global implique un traitement avant l'interaction proprement dite. Ce traitement consiste en la création soit de l'ontologie globale, soit du médiateur. Mais dans les deux cas, les Kbases hétérogènes sont ramenées à une KBase globale *homogène*.

2.5.3 Point de vue Local

Gérer la HS en utilisant un point de vue *local*, correspond à *s'interdire* de recourir à un point de vue global. Cette solution peut être adoptée pour plusieurs raisons :

1. La Kbase globale est difficile, coûteuse ou impossible à construire [15] : pour des raisons d'incohérence entre les données, de temps, de taille.

FIG. 1 – **Ontologie Globale** : Cas d'intersection de domaines correspondant à la proposition d'un ontologie globale.



2. Le dictionnaire est impossible à écrire : pas de correspondance un-à-un possible par exemple.
3. L'application *requiert* un traitement au niveau local.

Dans InterViews, on se trouve toujours dans le troisième cas. En effet, les agents computationnels ne seraient plus libres *d'évoluer* ou d'être modifiées (par d'autres agents ou par l'utilisateur) *dynamiquement* ; la *taille* de la Kbase globale nécessaire est imprédictible, puisque le nombre d'applets interagissant à un instant donnée n'est pas contrôlé ; et la *cohérence* d'une telle Kbase globale serait toujours remise en cause.

Dans le cas d'un point de vue local, il n'y a pas de traitement préalable avant l'interaction proprement dite. La KBase qui reçoit la requête, reçoit une expression qui est issue d'une autre KBase *hétérogène*.

3 Les Classes d'Interaction Sémantiquement Hétérogènes

3.1 Intersection de Domaines

En considérant l'intersection $\epsilon_1(KB_1) \cap \epsilon_2(KB_2)$, nous donnons une classification des situations que l'on peut rencontrer en quatre cas.

3.1.1 Ontologie Globale

Le cas illustré par la figure 1 correspond à la proposition d'une l'ontologie globale. Dans ce cas, nous avons :

$$\exists KB_{globale} \mid \forall i,\, \epsilon_i(KB_i) \subseteq \epsilon_{globale}(KB_{globale})$$

Et il en suit que l'intersection de domaines utile pour résoudre la HS, est toujours l'intersection d'une KBase avec cette $KB_{globale}$; et l'on a :

$$\forall i,\, \epsilon_i(KB) \cap \epsilon_{globale}(KB_{globale}) = \epsilon_{globale}(KB_{globale}).$$

Ce cas, qui correspond à construire $KB_{globale}$ en intégrant toutes les KB_i, et en gommant l'hétérogénéité.

3.1.2 Pas de Communication

Le cas illustré par la figure 2 correspond à toutes les situations où :

$$\forall i, j, \text{ avec } i \neq j,\, KB_i \cap KB_j = \emptyset$$

Ce cas mène donc à une impossibilité d'interaction, comme décrit dans [6].

FIG. 2 – **Pas de Communication** : Cas d'intersection de domaines correspondant à une interaction ne pouvant mener à aucune communication.

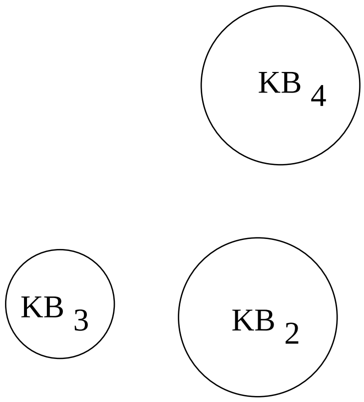
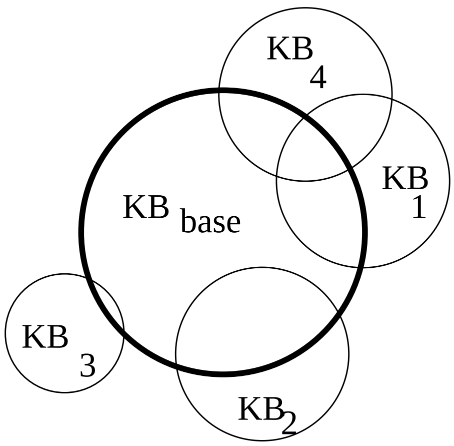


FIG. 3 – **Base Commune** : Cas d'intersection de domaines correspondant à la proposition d'une Kbase commune : la Base.



3.1.3 Une KBase Commune : La Base

Le cas illustré par la figure 3 correspond à une Kbase commune à toutes les KBases des agents. On appelle cette KBase, la Base, elle est notée KB_{base}. C'est elle qui sert de *médiateur* dans le cas de la construction d'un dictionnaire. On a :

$$\forall i,\ KB_i \cap KB_{base} \neq \emptyset$$

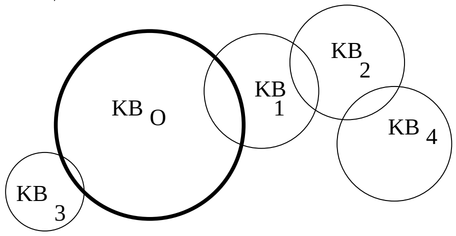
Dans ce cas, on pose l'existence d'une terminologie de base qui représente le *sous-domaine* d'expertise *commun* dans le cas où les agents sont orientés vers le *même* domaine d'application. Mais cette condition est en général *trop forte* lorsque les agents ne sont pas strictement orientés vers la même application. Enfin, constituer une telle base est difficile à mettre en oeuvre en pratique.

3.1.4 Une KBase initiatrice : Le Germe

Le cas illustré par la figure 4, correspond à la proposition d'un agent *germe*, noté KB_0, qui implique une interaction qui se déroule comme suit :

1. Au premier pas d'interaction, il existe un agent-germe ;

FIG. 4 – **Germe Initiatrice** : Cas d'intersection de domaines correspondant à la proposition d'une Kbase initiatrice : le *germe*, noté KB_0.



2. A chaque pas d'interaction suivant, tout agent entrant dans le système pour interagir partage au moins un sous-domaine avec :
   - soit le germe,
   - soit avec n'importe quel autre agent déjà enregistré dans le système , ce qui s'écrit :
   $\forall i$, $i > 1$ soit $(KB_i \cap KB_0 \neq \emptyset)$, soit $(\exists j,\, j \neq i,\, KB_i \cap KB_j \neq \emptyset)$

3.1.5 Différences de Domaines

Le fait que $\epsilon_1(KB_1) \cap \epsilon_2(KB_2) \neq \emptyset$, implique qu'il existe des différences, et des ressemblances entre les domaines des KBases.

Les sortes de *différences* entre ontologies ont déjà été analysées par Campbell [5] qui distingue entre les différences de :

Nommage hétérogénéité provenant de l'utilisation d'un même nom pour des objets différents ou de noms différents pour un même objet. Ce problème est recouvert par la question plus générale de la référence. Nous parlerons maintenant d'hétérogénéité sémantique de type référentielle.

Structure hétérogénéité provenant de la syntaxe. Cette hétérogénéité est écartée par le fait que nous utilisons le même formalisme pour exprimer toutes les KBases ainsi que les requêtes et les réponses.

Taxonomie hétérogénéité provenant de la structuration en hiérarchie du domaine. Nous parlerons désormais d'hétérogénéité sémantique de type taxonomique.

Expressivité hétérogénéité provenant du langages de représentation. Comme pour la structure, nous écartons ce type d'hétérogénéité grâce à l'utilisation d'un même langage de formalisation pour les KBases et les requêtes.

Les différences de structure syntaxique et d'expressivité ne sont pas pertinentes ici puisque nous avons supposé que les deux ontologies étaient exprimées sou formes de terminologies LD, ce qui nous ramène à n'envisager que les deux cas d'hétérogénéité sémantique :

1. HS Taxonomique
2. HS Référentielle

4 Algorithmes et Exemples

4.1 Algorithmes pour la Gestion de l'interaction

A1 : Gestion de l'Interaction. L'interaction est gérée de manière basique selon le schéma suivant :

	`GererInteraction(KB1,KB2)`
pas d'interaction :	`incr i`
initialisations :	`initi(KBe,KB1)`
	`initi(KBr,KB2)`
appel de fonction :	`GestionPasInteraction(i,KBe,KBr)`
fin :	`Validation`

A2 : Gestion d'un Pas d'Interaction Un pas d'interaction est géré selon l'algorithme suivant :

	`GererPasInteraction(i,KB1,KB2)`
émission de requête :	`Q=EmissionRequete(i,Requete(KBe))`
gestion de requête :	`R=GererRequete(Q,KBr)`

A3 : Gestion de la Requête Une requête est gérée par la KBase réceptrice selon l'algorithme suivant :

	`GererRequete(Q,KBr)`
fct sur le nb de patterns de Q :	`for a=0 to Nb_patterns(Q)`
test de type sur patterns Q :	`T=TypePattern(a,Q)`
cas terminologique :	`if T in (concept,role) P=GererPatternTbox`
cas référentiel :	`else if in (individu) P=GererPatternAbox`
production d'une réponse	`R= ProduireReponse(P)`

Les patterns sont gérés par des fonctions qui appliquent soit des algorithmes de matching, soit des algorithmes d'adaptation dans le cas de termes hétérogènes.

4.2 Hétérogénéité Sémantique de Type Terminologique

La HS terminologique concerne les TBoxes des KBases en LD. Les TBoxes des deux KBases présentées ci-dessus sont hétérogènes. Mais la requête envoyée par une KBase à un pas d'interaction particulier peut être :

1. Non hétérogène dans les termes qu'elle contient (pouvant conduire à un MUP) ;
2. Hétérogène explicitement dans les termes qu'elle contient (au sens de Weinstein).

Ces deux cas font cependant partie de l'étude de l'interaction terminologiquement hétérogène. Nous allons voir que même dans le cas de la gestion d'une requête non hétérogène, dès lors que les TBoxes sont hétérogènes, il y a des conséquences pour la TBox de l'agnet récepteur.

Les KBases et les requêtes présentées ci-dessous respectent respectivement la syntaxe des langages de description et de requête définis pour les LD dans le chapitre 2.

Soient Tb_1 et Tb_2 les deux terminologies des deux Kbases KB_1 et KB_2 :

$$
\begin{array}{ll|ll}
Tb_1 = & \{A \doteq E \wedge C & Tb_2 = & \{F \doteq \forall s.C \\
& D \doteq \forall r.C & & G \leq C \\
& B \leq E & & E \leq G \wedge F \\
& A \leq C\} & & A \doteq B \wedge C
\end{array}
$$

4.2.1 Requête Non-Hétérogène : Réduction des Mondes Possibles

Plaçons-nous à un pas d'interaction i quelconque. La fonction d'initialisation de `GererInteraction`, fixe le rôle des KBases envoyeur et récepteur :

- pour KB_e l'agent A_1 (qui est donc l'envoyeur pendant ce pas),
- pour KB_r l'agent A_2 (qui est donc le receveur pour i)

La fonction de gestion du pas d'interaction `GererPasInteraction` débute avec les arguments (i, KB_1, KB_2). Soit alors l'émission de requête $Q^i_{1\to 2}$:

$$KB_1 \models Q^i_{1\to 2} = (A \doteq E \wedge ?X)$$

La fonction de gestion de requête `GererRequete` doit retourner une interprétation de $Q^i_{1\to 2}$ dans la KBase KB_2. Détaillons le raisonnement qui est nécessaire pour trouver cette interprétation :

1. On constate que A est un concept connu de KB_2, il n'est donc pas sémantiquement hétérogène.
2. La définition locale de A dans Tb_2 est : $A \doteq B \wedge C$. Cette définition est pourtant différente de celle que KB_1 possède localement dans Tb_1 pour le même symbole de concept : A.
3. On matche l'expression $A \doteq E \wedge ?X$ avec $A \doteq B \wedge C$. Ce matching se note : $E \wedge ?X :: B \wedge C$; où E est aussi un concept connu de Tb_2 avec la définition $E \leq G \wedge F$ où $G \leq C$ et donc $E \leq C$.
4. Pour répondre à $Q^1_{1\to 2}$, il faut trouver un concept X tel que son intersection avec E, $X \cap E$, dénote *exactement* le même ensemble que celui dénoté par $B \cap C$, en sachant $E \leq C$.
5. Le concept X est B ssi $B \leq E$.
6. L'agent KB_2 donne pour réponse à $Q_{1\to 2}$, la substitution valide suivante : $R^i_{2\to 1} = [X \to B]$

Après la production de cette réponse, KB_2 peut *informer* KB_1 de l'inférence établie : $B \leq E$. Si cette communication a lieu, l'information $B \leq E$ peut être utilisée à un pas d'interaction ultérieur pour détecter des MUP éventuels.

Dans tous les cas, l'effet de l'interaction de KB_2 avec KB_1 est *d'ajouter* une *information* dans Tb_2 et donc de *réduire* les nombre de modèles *possibles* pour Tb_2.

4.2.2 Requête Hétérogène : Utiliser l'adaptation

Considérons à présent le pas d'interaction j avec $j > i$, on a toujours $KB_e = KB_1$ pour ce pas.

Gérons la nouvelle requête posée par KB_1 à KB_2 :

$$Q^j_{1\to 2} = ?X(\forall r.C)$$

La fonction de gestion de la requête essaiera d'abord de matcher de pattern de la requête avec la définition locale :

$$F \doteq \forall s.C$$

Ce matching ne pourra réussir avec la substitution $X \to F$, que si nous supposons *aussi* que $r \to s$. Mais nous savons que r est sensé être un rôle *constant* dans la requête telle qu'elle est envoyée, et non un pattern.

Dans ce cas, nous proposons *d'adapter* ce terme dans la requête pour lui donner un sens connu pour KB_2 dans Tb_2. On note une telle adaptation avec l'opérateur $\rightharpoondown$. Cette adaptation sera effectuée par une autre fonction que nous présenterons dans la troisième partie, comme mentionné plus haut.

Ainsi, la substitution valide donnée par KB_2 comme réponse sera :

$$R^j_{2\to 1} = (X \to F, r \rightharpoondown s)$$

L'adaptation est un pas classique du raisonement par analogies, et a inspiré beaucoup de modèle informatiques [13]. En effet, pour résoudre un nouveau problème, on a souvent recours à la construction d'une solution par analogie avec un cas connu, c'est le raisonnement par cas. Notre idée est que l'interpétation de termes dans des requêtes peut se faire selon le même mécanisme, à partir de "cas", de termes, connus. Les algorithmes utilisés pour réaliser cette adaptation sont directement inspirés par nos précédentes recherches dans ce domaine [25] [26].

4.3 Hétérogénéité de type Référentielle

Les problèmes d'hétérogénéité référentielle font intervenir les identifieurs d'individus des ABoxes Ab_1 et Ab_2 de KB_1 et KB_2. Soient les Aboxes suivantes pour les deux agents :

Les identifieurs dans les ABoxes des KBases en interaction peuvent être *locaux* ou *globaux*.

4.3.1 Identifieurs Globaux

Dans ce cas, les identifieurs dénotent un même objet qu'ils soient dans l'une ou l'autre des ABoxes. Ceci est vrai pour tout agent enregistré dans le système. Ainsi chaque Kbase est réduite à être simplement une vue différente sur la même base [7]. La question est alors de savoir s'il est possible d'avoir différentes théories du même domaine. Mais ce cas est à mettre en correspondance avec les traitements globaux de la HS entre domaines.

4.3.2 Identifieurs Locaux

Dans ce cas, chaque agent utilise des identifieurs locaux pour référer à des objets de l'environnement *commun* en fonction de leur terminologie personnelle. Dans ce cas, il ne peut pas y avoir de *conflit* entre ABoxes.

Cependant, il y a alors un problème de référence aux objets. Comment savoir quel élément une KBase désigne ?

Nous proposons d'avoir recours à une référence basée sur un *accès* par le *contenu*. C'est-à-dire que les identifieurs utilisés pour désigner des éléments du domaine commun sont remplacés par leurs descriptions, c'est-à-dire des assertions au sujet de leurs propriétés. Traiter ensuite cette expression de LD contenant une telle description revient à traiter un problème d'hétérogénéité terminologique.

5 Conclusions et Perspectives

5.1 Synthèse

Nous avons modélisé les notions d'agent dialogique, d'interaction, de requête et de réponse en logique des descriptions. Un agent dialogique est vu comme une base de connaissances LD, une requête comme une expression de LD contenant des patterns typés, et une interaction une suite finie d'échanges de telles expressions avec une interprétation en termes de sémantique ensembliste.

Dans ce contexte, nous avons proposé une définition et une classification des types d'hétérogénéité sémantique qui émergent entre deux agents dialogiques. En considérant les hypothèses propres au projet InterViews, nous avons restreint ensuite le problème à celui de la gestion de la HS par la Kbase récepetrice au niveau d'ue seule requête (au cours d'un seul pas d'interaction) et de façon locale.

Il apparaît alors que nous devons résoudre deux types de HS : la HS terminologique au niveau des TBoxes, et la HS référentielle au niveau des ABoxes.

Au cours d'une interaction sémantiquement hétérogène pour les terminologies, on peut rencontrer deux cas de figure pour un pas d'interaction quelconque :

1. La requête n'est pas hétérogène par rapport à la base réceptrice ;
2. La requête est hétérogène par rapport à la base réceptrice.

Dans le cas d'un requête non-hétérogène, nous proposons de procéder à un pattern matching comme pour le traitement classique des requêtes dans les bases de données en LD, à cela près que des déductions supplémentaires peuvent être possibles, ayant des conséquences au niveau de la mise à jour de la TBox de l'agent récepteur.

5.2 Proposition

La proposition que nous faisons pour résoudre l'hétérogénéité sémantique des *termes* contenus dans les requêtes, à un pas d'interaction, est donc la suivante :

- *Matcher* les termes *inconnus* avec les termes de la Kbase *locale*, comme on le fait pour des *variables* libres dans une requête normale à une base de données simple [2]. Cela peut entraîner un raisonnement, et même nécessiter des hypothèses supplémentaires. Ces hypothèses supplémentaires sont considérées comme des informations déduites par l'agent receveur et réduisant ainsi le nombre de modèles possibles de sa KBase KB_r.
- *Adapt*er les termes matchés à l'aide de transformations élémentaires qui respectent le type des termes appariés. Cette adaptation se fait en utilisant des *analogies* avec la *structure* des connaissances contenues dans la Kbase locale.

Ces deux opérations sont des opérations de base classiquement considérées dans le modèles de raisonnement par analogies [10] [16]. Les outils que nous souhaitons développer pour résoudre la HS entre AD doivent donc s'inspirer des algorithmes de traitement automatique des problèmes d'analogies que nous pouvons étudier en IA.

5.3 Travaux Futurs

Pour résoudre complètement la HS entre agents dialogiques au sein d'InterViews, il faut ensuite intégrer pleinement les algorithmes ci-dessus au cycle de traitement des requêtes propre à notre projet. Les agents dans InterViews sont décrits avec le langage déclaratif de description, appelé VDL[1], dont le sous-ensemble gérant l'échange et l'interprétation des requêtes est appelé VQL[2]. Nos travaux futurs incluent donc l'écriture des algorithmes d'adaptation et de matching en VQL, pour une expérimentation *in situ*.

Références

[1] Alexander Borgida. Description logics in data manageent. Portion of the paper based on invited talk at IFIP'92, available on the web at Borgida's homepage, 1995.

[2] Alexander Borgida and Deborah L. McGuiness. Asking queries about frames. In *KR'96*, 1996.

[3] R. J. Brachman. *A structural paradigm for representing knowledge.* PhD thesis, Division of Engineering, Harvard University, 1977.

[1] VDL pour View Design Language, car dans InterViews les agents logiciels sont appelés des Vues (View).
[2] View Query Language.

[4] Ronald J. Brachman. "i lied about the trees" : Or, defaults and definitions in knowledge representation. *The AI Magazine*, 6 :80–93, 1985.

[5] Alistair E. Campbell, Hans Chalupsky, and Stuart C. Shapiro. Ontological mediation : An analysis. DRAFT, OM web site, 1995.

[6] John Casti. *Paradigmes Perdus.* InterEditions, 1991.

[7] Alon Y. Levy Catriel Beeri and Marie-Christine Rousset. Rewriting queries using views in description logics. In *Proceedings of the 16th ACM Symposium on Principles of Database Systems.* PODS-97, 1997.

[8] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. *Principles of Knowledge Representation*, chapter Reasoning in Description Logics, pages 193–238. Studies in Logic. CSLI Publications, 1996.

[9] Tim finin and Richard Fritzson. KQML as an agent communication language. In *CIKM'94*. ACM Press, November 1994.

[10] Dedre Gentner. Structure-mapping a theoretical framework for analogy. *Cognitive Science*, 7(2) :155–170, April-June 1983.

[11] Nicola Guarino. Formal ontology, conceptual analysis and kowledge representation. *IJHCS*, 43(5/6), 1995.

[12] Sylvie Pesty Yves Demazeau Guillaume Chicoisne, Pierre-Michel Ricordel. Outils et pistes pour la pratique du dialogisme entre agents. In *Systèmes Multi-Agents, de l'interaction à la sociabilité.*, pages 16–176. JFIADSMA'98, Hermès, 1998.

[13] Rogers P. Hall. Computational approaches to analogical reasoning : A comparative analysis. *Artificial Intelligence*, 39(1) :39–120, may 1989.

[14] Graeme Hirst, Susan McRoy, Peter Heeman, Philip Edmonds, and Diane Horton. Repairing conversational misunderstandings and non-understandings. *Speech Communication*, 15, 1994.

[15] Richard Hull. Managing semantic heterogeneity in databases : A theoretical perspective. In *PODS 97*. ACM Symposium on Principles of Database Systems, 1997.

[16] Bipin Indurkhya. Modes of analogy. In K.P.Jantke, editor, *Analogical and Inductive Inference*, volume 397 of *LNAI*, pages 217–230. Springer-Verlag, 1989.

[17] Marvin Minsky. *Extraterrestrials : Science and Alien Intelligence*, chapter Communication with Alien Intelligence. Cambridge University Press, 1985.

[18] Bernhard Nebel. *Reasoning and Revision in Hybrid Representation Systems.* Number 422 in LNAI. Springer-Verlag, New York, 2nd edition, 1995.

[19] W. Quine. *From a Philosophical Point of View.* Harvard University Press, 3rd edition, 1980.

[20] J.P. Sansonnet. Interviews. Technical report, 1999.

[21] Andrea Schaerf. *Query Answering in Concept-Based Knowledge Representation Systems : Algorithms, Complexity, and Semantic Issues.* PhD thesis, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1994.

[22] J.R. Searle. *Speech Acts.* Cambridge University Press, 1969.

[23] Luc Steels. *The Talking Heads Experiment.* LABORATORIUM, Antwerpen, 1999.

[24] F. Flores T. Winograd. *Understanding Computers and Cognition : A New Foundation for Design.* Ablex, Norwood, 1986.

[25] Erika Valencia and Jean-Louis Giavitto. Algebraic topology for knowledge representation in analogy solving. In ECCAI, editor, *European Conference on Artificial Intelligence (ECAI98)*, pages 88–92, Brighton, UK, 23–28 August 1998. Christian Rauscher.

[26] Erika Valencia, Jean-Louis Giavitto, and Jean-Paul Sansonnet. Formalisme simplicial pour la représentation des connaissances et application à l'analogie. In *JNMR99*, march 1999.

[27] Peter C. Weinstein and William P.Birmingham. Comparing concepts in differentiated ontologies. In *KAW'99*, 1999.

[28] B.J. Wielinga and A. Th. Schreiber. Reusable and shareable knowledge bases : A european perspective. KADS-II project, ESPRIT.

TAB. 1 – Syntaxe de TF

$\langle terminologie \rangle$	$::=$	$\{\langle termeIntroduction \rangle \vert \langle restriction \rangle\}^*$
$\langle termeIntroduction \rangle$	$::=$	$\langle conceptIntroduction \rangle \vert$
		$\langle roleIntroduction \rangle$
$\langle conceptIntroduction \rangle$	$::=$	$\langle atomicConcept \rangle \doteq \langle concept \rangle \vert$
		$\langle atomicConcept \rangle \leq \langle concept \rangle \vert$
		$\langle atomicConcept \rangle \leq ANYTHING$
$\langle roleIntroduction \rangle$	$::=$	$\langle atomicRole \rangle \doteq \langle role \rangle \vert$
		$\langle atomicRole \rangle \leq \langle role \rangle \vert$
		$\langle atomicRole \rangle \leq ANYROLE$
$\langle concept \rangle$	$::=$	$\langle atomicConcept \rangle \vert$
		$(AND \langle concept \rangle^+) \vert$
		$(ALL \langle role \rangle \langle concept \rangle) \vert$
		$(ATLEAST \langle nombre \rangle \langle role \rangle) \vert$
		$(ATMOST \langle nombre \rangle \langle role \rangle)$
$\langle role \rangle$	$::=$	$\langle atomicRole \rangle$
$\langle restriction \rangle$	$::=$	$(DISJOINT \langle atomicConcept \rangle \langle atomicConcept \rangle)$
$\langle nombre \rangle$	$::=$	$\langle entierNonNegatif \rangle$
$\langle atomicRole \rangle$	$::=$	$\langle idetinfieur \rangle$
$\langle atomicConcept \rangle$	$::=$	$\langle identifieur \rangle$

Annexe : Le langage Terminologique TF

Syntaxe Générale

Le langage terminologique permet de définir une *terminologie*. Cette terminologie est la représentation structurée et déclarative d'un vocabulaire d'un domaine. Une terminologie est définie comme une séquence de *déclarations*, donc d'introduction de définitions, de *termes* ou de *restrictions* sur des définitions déjà déclarées :

$$\langle terminologie \rangle ::= \{\langle termeIntroduction \rangle | \langle restriction \rangle\}^*$$

$$\langle termeIntroduction \rangle ::= \langle conceptIntroduction \rangle | \langle roleIntroduction \rangle$$

Nous présentons maintenant plus en détail la syntaxe des rôles, des concepts et des constructeurs autorisés.

Les rôles

Un **rôle** est l'entité qui exprime une relation entre deux membres d'une paire d'individus, c'est donc un prédicat *binaire* en LPO. Rappelons que les LD sont historiquement issues des SIN qui sont des graphes. Ainsi, un rôle est la représentation d'un arc de graphe, il est donc limité à deux extrémités et les rôles sont donc limités à exprimer des relations binaires[3].

De même que pour les rôles, un rôle atomique est en fait un simple identifieur, de plus on peut définir un rôle par :

[3]Certains chercheurs se sont d'ailleurs penchés sur cette limitation, et il a été prouvé que toute relation n-aire peut s'exprimer à l'aide de relations binaires ; mais il n'a pas été dit que cela soit naturel, agréable, expressif, etc. Le débat reste ouvert dans la communauté.

– son introduction en tant que *sous-rôle* d'un autre rôle : $\langle atomicRole\rangle \leq \langle role\rangle$;
– son introduction en tant que sous-rôle du rôle quelconque, qui *subsume* tous les rôles ($ANYROLE$) : $\langle atomicRole\rangle \leq ANYROLE$;
– son *identification* (signe $\doteq$) avec une expression : $\langle atomicRole\rangle \doteq \langle role\rangle$.

Les concepts

Un **concept** exprime la notion de *classe* en tant que groupe d'individus, il correspond à un prédicat *unaire* en logique des prédicats du premier ordre. On définit de nouveaux concepts par une introduction de concept définie comme suit dans TF :

$$\langle conceptIntroduction\rangle ::= \langle atomicConcept\rangle \doteq \langle concept\rangle |$$

$$\langle atomicConcept\rangle \leq \langle concept\rangle | \langle atomicConcept\rangle \leq ANYTHING$$

Ainsi, on note que l'on peut introduire des concepts de trois manières :

– En définissant ($\doteq$ pour la *définition*) un nouveau concept atomique $\langle atomicConcept\rangle$ avec un détail de ses caractéristiques dans $\langle concept\rangle$;
– En introduisant un nouveau concept atomique $\langle atomicConcept\rangle$ et en le positionnant par rapport au reste de la terminologie ; c'est-à-dire en précisant de quel concept $\langle concept\rangle$ il est un sous-concept ($\leq$ pour la relation *d'inclusion*) ;
– En introduisant un nouveau concept atomique $\langle atomicConcept\rangle$ et en le positionnant par rapport au plus grand concept possible de la terminologie ($ANYTHING$, noté aussi $\top$) ;

Introduire un concept atomique revient à introduire simplement un nouvel *identifieur* :

$$\langle atomicConcept\rangle ::= \langle identifieur\rangle$$

Pour introduire un concept complexe, c'est-à-dire une combinaison de concepts, on dispose d'opérateurs de construction. Les opérateurs de construction de TF sont :

– AND ($\wedge$) pour exprimer le “et” logique, il permet d'aligner plusieurs termes pour définir un concept : on l'utilise comme ça : ($description_1$ $\wedge description_2$). On peut bien sûr emboîter plusieurs $\wedge$ et créer une conjonction à plus de 2 termes. Par exemple une moto est un objet-à-deux-roues et un engin-à-moteur, cela s'écrit $moto ::= (AND(deuxRoues, enginMoteur))$.
– ALL ($\forall$) pour exprimer une restriction de *valeur* sur le role-filler d'un rôle. Rappelons qu'un rôle est la représentation d'une relation binaire $R(x, y)$, dans cette notation, le rôle-filler est y (c'est l'argument). Restreindre la valeur de ce filler revient à donner un *type* (un domaine) obligatoire pour y. On utilise le constructeur $\forall$ comme ça : ($\forall role$:*concept*). Cela correspond à dire que tous les concepts reliés au concept défini par le lien *role* doivent obligatoirement être du type *concept*. Par exemple si on écrit ($\forall$ $chef$ $Femme$) cela veut dire que tous les concepts reliés par le lien $chef$ doivent être des $Femme$.
– $ATLEAST$ ($\geq$) et $ATMOST$ ($\leq$) , pour exprimer des restrictions de *nombre*, on l'utilise comme ça : ($\geq$ n *role*). Cela veut dire que le concept doit avoir au moins n concepts reliés à lui par le lien *role*. Par exemple on peut écrire ($\geq$ 4 *Leg*) cela veut dire qu'on définit un concept de trucs qui ont au minimum 4 *Leg*.

Voici la syntaxe générale correspondante pour ces constructeurs :

$$\langle concept\rangle ::= \langle atomicConcept\rangle | ((\langle concept\rangle \wedge)^{+}) | (\forall \langle role\rangle : \langle concept\rangle) |$$

$$(\geq \langle nombre\rangle \langle role\rangle) | (\leq \langle nombre\rangle \langle role\rangle)$$

Enfin, chaque définition de concept peut reposer soit sur un concept prédéfini, soit sur un concept déjà défini par l'utilisateur.

Les constructeurs

Les opérateurs de construction de concepts cités plus haut, sont appelés aussi plus simplement des *constructeurs*[4].

Les constructeurs de concepts

- De *formation* de *concepts* : la conjonction (AND), la restriction de valeur (ALL) la restriction de nombre ($ATLEAST$, $ATMOST$). Mais il en existe d'autres pour construire des concepts, qui ne sont pas utilisés dans TF (par exemple l'égalité,...).
- D'*introduction* de concepts : *définition* d'un concept atomique ($\doteq$) en l'identifiant à une expression, ou bien *spécification* ($\leq$) comme sous-concept d'un autre. Dans les deux cas, on peut utiliser des concepts complexes, construits avec les constructeurs cités en premier item pour l'expression de droite dans ces définitions. Un concept apparaissant à gauche de tels symboles sont appelés : *primitif* s'il est suivi d'un $\leq$ et *défini* s'il est suivi d'un $\doteq$. Un concept apparaissant à droite de tels symboles est appelé *atomique*.

Les constructeurs de rôles

- De *formation* de *rôles* : par exemple inverser le sens de la relation, restreindre la plage de valeurs de l'un des deux individus pris dans la relation, composition de rôles. Nous n'avons pas parlé de ces constructeurs car il ne sont pas autorisés dans TF.
- D'*introduction* de rôles : *définition* d'un rôle atomique ($\doteq$) en l'identifiant à une expression, ou bien *spécification* ($\leq$) comme sous-rôle d'un autre. Dans les deux cas, on peut utiliser des rôles complexes, construits avec les constructeurs autorisés.

Dans ce document nous insistons sur le fait que TF se *restreint*, à des rôles non composés, c'est-à-dire qu'il n'y a aucun constructeur de formation de role autorisé. Le fait d'introduire d'autres opérateurs, comme par exemple ceux du troisième type, permet de définir des *extensions* du langage TF : cela débouche sur d'autres logiques de description.

Ainsi, l'ensemble des constructeurs est différent selon la LD considérée, et un langage de description L donné est *complètement défini* par la donnée des constructeurs qu'il autorise dans sa syntaxe.

Sémantique dénotationnelle

Le sens *intuitif* d'un formalisme de représentation des connaissances doit être appuyé par un modèle *théorique* de sémantique. Un moyen classique d'y arriver est de donner une traduction du formalisme en termes de *logique* du *premier ordre*. Cependant, une sémantique *directe* est plus utile pour les calculs de complexité par exemple. Nous allons donc donner une sémantique *dénotationnelle* pour le langage TF.

Sémantique

Fonction d'extension

Pour formaliser la sémantique que l'on donne pour chaque élément de la terminologie, on introduit la fonction d'extension ε telle que cette fonction appliquée à un concept de la terminologie renvoie l'extension que dénote ce concept dans le domaine D, ainsi :

$$\varepsilon(concept) = extension_D(concept)$$

Pour définir ε plus formellement, nous introduisons d'abord les notations suivantes :

[4] Ils sont écrits dans cette section sous leur forme concrète. En effet, le fait d'écrire le nom du constructeur en toutes lettres suivi de ses arguments s'appelle la forme concrète, elle est plus pédagogique. On voit souvent une forme plus abstraite qui se base sur des notations de la logique du premier ordre. On distingue les constructeurs de concepts, et de rôles.

N_c : ensemble des *concepts atomiques* de la terminologie
N_r : ensemble des *rôles atomiques* de la terminologie
N : ensemble général des termes *atomiques* , il contient donc les concepts atomiques et les termes atomiques.
TF_c : ensemble de *tous les concepts*, pas forcément atomiques
TF_r : ensemble de *tous* les *rôles*, pas forcément atomiques
TF_T : ensemble de *tous les termes* de la terminologie

Concepts et rôles

La sémantique des concepts et des rôles est donnée par la fonction d'extension ε, définie par :

$$\varepsilon : \begin{cases} TF_c \to 2^D \\ TF_r \to 2^{D \times D} \end{cases}$$

$$\varepsilon(ANYTHING) = D$$

$$\varepsilon(ANYROLE) = D \times D$$

En effet, l'extension du plus grand concept correspond au domaine D en entier, et l'extension de la plus vaste relation correspond à $D \times D$ puisque les relations ont pour extension des couples de valeurs dans D.