

---

# Evolution des ontologies de domaine dans le projet ACSIS

**Guilaine Talens\*, Danielle Boulanger\*, Magali Séguran\*\***

\*MODEME, Université Jean Moulin,  
6, cours Albert Thomas – BP 8242  
69355 Lyon cedex 08, France  
{talens,db}@univ-lyon3.fr

\*\*SAP Labs France, SAP Research  
805, avenue du Dr. Maurice Donat  
06254 Mougins Cedex  
magali.seguran@sap.com

---

*RÉSUMÉ. Pour résoudre les conflits sémantiques issus de la coopération de sources d'information hétérogènes, nous proposons un système multi-agents basé sur l'utilisation d'ontologies de domaine qui suivent l'évolution des sources locales et évoluent en fonction du dialogue entre les agents. Lors de l'ajout d'une source locale par un expert, l'ontologie correspondante est créée de façon automatique, l'expert peut la compléter. Cette nouvelle ontologie est diffusée aux autres agents. Lors de l'exécution de requêtes utilisateurs, les ontologies peuvent également être infirmées ou validées partiellement. Toutes ces interactions font évoluer les ontologies.*

*ABSTRACT. To solve the semantic conflicts in the cooperation of heterogenous databases, we propose a multi-agent system which contains domain ontologies. They evolve with the update of the local bases but also with the interactions between the agents. When a local source is added by an expert, the relevant ontology is dynamically created by the system. The expert can increase it. This new ontology is broadcasted to the other agents. When an end user performs a query, ontologies can be partially validated or invalidated. All the agent interactions imply ontology evolution.*

*MOTS-CLÉS : Création d'ontologies, évolution des ontologies, interopérabilité sémantique dans les systèmes d'information.*

*KEYWORDS: ontology creation, ontology evolution, semantic interoperability in information systems .*

---

## 1. Introduction

La mise en œuvre de processus coopératifs entre applications requiert la résolution des conflits issus de l'interopérabilité de sources d'information distinctes et hétérogènes. Ces conflits découlant de l'hétérogénéité peuvent se situer au niveau *technique* (disparités entre systèmes d'exploitation et de plate-formes), *syntactiques* (différences au niveau des modèles de données) ou *applicatif*. L'hétérogénéité applicative concerne les différences au niveau *des schémas*, au niveau *structurel* (conflits de généralisation/spécialisation) et *sémantique*.

Les conflits sémantiques sont composés :

- de conflits de synonymie (deux entités qui se ressemblent sémantiquement peuvent avoir différents noms) et d'homonymie (deux entités qui sont différentes sémantiquement peuvent avoir le même nom),
- de conflits de représentation et de précision des données (deux attributs qui se ressemblent sémantiquement peuvent être représentés selon des valeurs ou des précisions différentes) et les conflits d'échelle (deux attributs qui se ressemblent sémantiquement peuvent être représentés selon différentes unités).

Nous proposons un système multi-agents qui gèrent ces conflits grâce à des ontologies de domaine. Lors de l'ajout d'une nouvelle source locale un agent la représentant est créé. A partir du schéma de la base, une ontologie est créée automatiquement. L'expert qui demande le rattachement de cette nouvelle base peut également compléter par ses connaissances cette nouvelle ontologie. Une fois le rattachement de la source terminée, le nouvel agent se rattache ensuite aux autres agents et l'ontologie contenue dans ce dernier leur est communiquée.

Les utilisateurs finals demandent l'exécution de requêtes. Ces dernières sont diffusées à l'ensemble des agents. Ils répondent en totalité, partiellement ou pas du tout en fonction des connaissances qu'ils détiennent. Lors de cette étape des liens temporaires de synonymie, de similarité et de différence d'échelle sont créés ; ils seront par la suite validés ou invalidés par les utilisateurs. Lors de cette phase, les ontologies évoluent donc aussi.

Tout d'abord en partie 2, nous décrirons les ontologies encapsulées dans les agents, ensuite en partie 3 nous détaillerons l'évolution des ontologies grâce à des protocoles d'interaction entre les agents. Dans la partie 4, nous comparerons nos travaux à des recherches similaires. Enfin nous concluons en partie 5.

## 2. Ontologies dans les agents

L'architecture ACSIS (Agents pour la Coopération de Systèmes d'Information Sécurisées) propose un système multi-agents pour la résolution des conflits structurels et sémantiques (Boulangier *et al.*, 2001), (Couturier *et al.*, 2003), (Séguran, 2003).

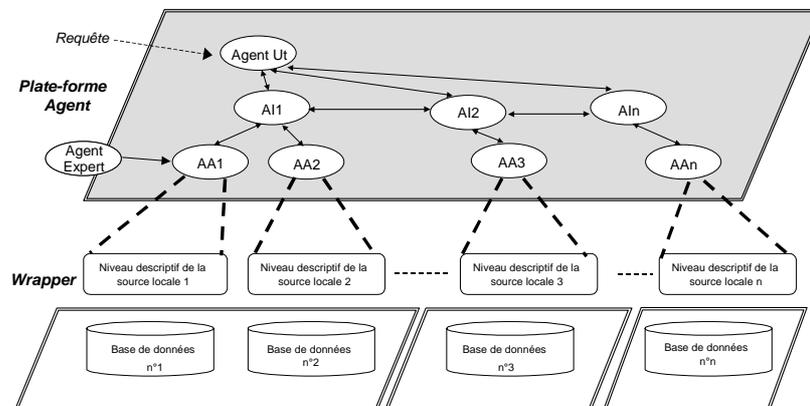
## 2.1. Un modèle d'agent

De la définition d'un agent donnée par Jacques Ferber (Ferber, 1995), ressortent les propriétés clés : autonomie, perception, action, communication. Ainsi, les agents de l'architecture ACSIS disposent des modules suivants : les modules ontologie (connaissances et liens créés à partir de la base locale), accointances (liste des agents proches sémantiquement), raisonnement, communication, comportement.

Le système proposé est composé de plusieurs types d'agents :

- les *Agents d'Accès aux sources locales* (AA) assurent la participation des données locales au sein des processus coopératifs. Ils traduisent la requête Q (provenant des Agents Informationnels) en requête Q' exprimée dans le langage propre de manipulation de données de la base locale et exécutent cette requête Q'.

- les *Agents Informationnels* (AI) regroupent les AA proches sémantiquement dans un réseau d'accointances, découpent les requêtes en sous-requêtes (cf. section 3.2), récupèrent les résultats pour obtenir une solution globale. Les AI communiquent entre eux lorsqu'ils n'ont pas la connaissance suffisante pour répondre à une requête. Un AI est un agent cognitif qui a accès à des sources d'information hétérogènes et qui acquiert, négocie et maintient une information pertinente afin de répondre à des requêtes d'utilisateurs ou d'autres agents (Klusck, 2001). Chaque AA ou AI s'enrichit à partir de sa base de connaissances interne (ontologie) et des évolutions provenant des différentes interactions.



**Figure 1.** Les différents types d'agents

- les Agents Utilisateur et Expert servent d'intermédiaire entre les utilisateurs et les agents AI ou AA. L'*Agent Utilisateur* est gestionnaire de la requête (saisie de la requête, validation des résultats, re-exécution requête). L'*Agent Expert* sert d'interface avec un expert humain qui dispose d'une expertise sur le domaine de la

base locale (approche objet ou relationnelle, définition de liens intra-base et choix d'un nom de domaine représentatif). Il permet à un humain de communiquer des informations sur une base au travers d'interface prédéfinie (Quel est le domaine ? Est-ce une base relationnelle ? Quels types de lien entre quels ODDs ? ...). Les différents types d'agents et leur rôle au sein du système multi-agents sont présentés dans la Figure 1.

Les agents disposent de connaissances sémantiques qui sont décrites sous la forme d'ontologies de domaine.

## **2.2. Constitution des ontologies de domaine**

Les modèles pour la description des ontologies trouvent leur fondement dans des domaines comme l'intelligence artificielle, la représentation de connaissances ou les bases de données (Culot, 2003). Deux écoles coexistent :

- L'une orientée Intelligence Artificielle propose des logiques descriptives associées aux outils d'inférence. Ce qui permet de poser un cadre formel.
- L'autre orientée Base de Données présente des modèles conceptuels étendus permettant de modéliser l'ensemble des informations.

Nous adopterons la définition des ontologies dans le cadre d'une approche base de données/partage de connaissances. Les ontologies sont spécifiées sous la forme d'une définition du vocabulaire de représentation. Les travaux de Gruber (Gruber, 1993) et ensuite de Guarino (Guarino, 1997) ont défini l'ontologie comme « la description partielle et explicite d'une conceptualisation ». Une conceptualisation étant considérée comme un ensemble de concepts, de relations, d'objets et de contraintes qui définissent le modèle sémantique d'un domaine. D'autres définitions, plus récentes, ont été citées dans le cadre de projets orientés coopération-médiation. Par exemple, dans le cadre du projet Observer (Mena *et al.*, 2000) (Mena *et al.*, 2001) : « une ontologie est une description des concepts et des relations qui peuvent exister pour un agent ou une communauté d'agents ». Les termes d'un vocabulaire, liés entre eux par des relations sémantiques (synonymie, polysémie, etc.), constituent l'ontologie.

Dans le contexte du projet ACSIS, le respect des propriétés d'autonomie, d'hétérogénéité et de distribution des bases locales nécessitent le recours à un niveau de représentation des méta-données issues de sources d'information locales (Objets Descriptifs de Données) mais aussi des liens sémantiques entre ces objets. Les méta-classes conceptuelles et les liens associés constituent une ontologie.

Chaque AA représente une source locale et correspond à un domaine. L'ontologie qu'il détient est constituée d'ODDs (Objets Descriptifs de Données) et des liens intra-base entre ces ODDs. Les AIs eux sont des agents multi-domaines qui regroupent un ensemble de AAs. Les ontologies qu'ils détiennent sont constituées de liens sémantiques inter-bases et d'ODDs.

### 2.2.1. Description des Objets Descriptifs de Données

Chaque ODD décrit un élément du modèle de données local (par exemple, pour le modèle relationnel, une table, un attribut, pour un modèle objet, une classe, une propriété, etc.). On distingue les **ODDs de type relation/objet** et les **ODDs de type attribut** :

- les ODDs attributs du modèle objet (attributs monovalués ou multi-valués qui décrivent tout attribut de classe ou bien les attributs *références* qui décrivent un attribut stockant un pointeur sur un autre objet). L'ontologie suivante (cf. Figure 2) est composée des types d'objet personne (n°SS, nom), salarié (fonction, abonnement\_sportif) et abonnement (n°abonnement, type, durée) : l'ODD de type objet salarié spécialise l'ODD de type objet personne. Cette source est gérée par un système de gestion de base de données orienté objet. Elle présente le cas d'un salarié (spécialisation d'une personne) qui dispose d'un ou plusieurs abonnements sportifs dans le cadre de son entreprise.

- les ODDs attributs de relation : clés de relation, clés étrangères et attributs de relation qui permettent de décrire tout attribut de relation monovalué. L'ontologie de la figure 3 est composée des relations projet (nom, type), employé (n°SS, nom, salaire) et travaille qui est composé de deux ODDs clé de relation nom et n°SS qui référencent respectivement l'ODD nom de l'ODD relation projet et l'ODD n°SS de l'ODD relation employé. Cette source décrit à l'aide d'une base de données relationnelles, des employés travaillant sur des projets.

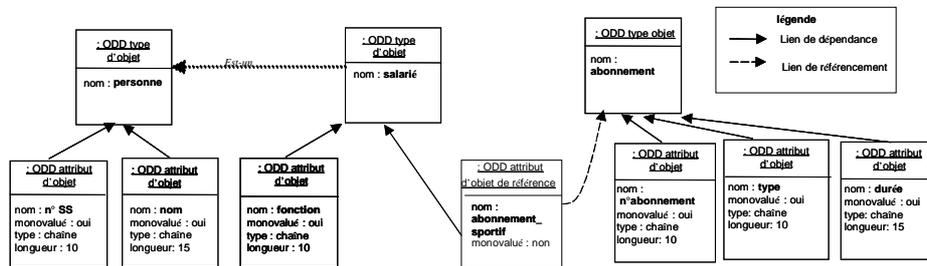


Figure 2. Ontologie créée à partir d'une base de données objet

### 2.2.2. Description des liens

Des liens définissent l'appariement entre deux ODDs en fonction de caractéristiques schématiques, structurelles ou sémantiques.

Les liens de **schéma intra-base** (reliant des ODDs issus d'une même source locale) sont automatiquement extraits des bases locales (cf. Figure 2 et 3) :

- les liens de *dépendance* permettent de décrire toute relation entre un ODD attribut et un ODD type objet/relation,



n'ayant pas besoin de validation, soit des liens validés par l'utilisateur) ou de **type utilisateur** qui sont des liens insérés par l'utilisateur.

Les différents liens sémantiques sont :

- Les liens de **synonymie** permettent de décrire un lien d'équivalence sémantique entre deux ODDs alors qu'ils ne disposent pas du même nom.
- Les liens de **non synonymie** permettent de décrire un lien de différence sémantique entre deux ODDs de noms différents.
- Les liens de **similarité** permettent de décrire un lien d'équivalence sémantique entre deux ODDs de même nom.
- Les liens de **homonymie** permettent de décrire un lien de différence sémantique entre deux ODDs disposant du même nom.
- Les liens de **d'échelle** permettent de décrire un lien d'équivalence au niveau de l'échelle entre deux ODDs de même nom.
- Les liens de **différence d'échelle** permettent de décrire un lien de différence au niveau de l'échelle de deux ODDs de même nom.

Des liens de synonymie, de similarité, d'échelle, de différence d'échelle, ... peuvent être construits manuellement par un expert ou détectés automatiquement. La création de ces derniers va être explicitée dans la partie suivante.

### **3. Evolution des ontologies pour la résolution des conflits sémantiques grâce à des protocoles d'interaction entre les agents**

La résolution des conflits sémantiques par les agents consiste à exploiter les liens sémantiques de leurs ontologies au cours de l'enregistrement d'une base locale dans le système mais aussi lors de l'exécution de requêtes multi-bases. Au cours de ces deux étapes, un ensemble de protocoles d'interaction entre agents permet de faire évoluer automatiquement les ontologies de domaine grâce à la détection de nouveaux liens qui n'ont pas été définis par l'expert du domaine. La mise à jour des sources locales est aussi gérée.

#### **3.1. Enregistrement d'une base de données locale**

La phase d'enregistrement d'une source locale au sein du système débute par la création d'un AA et se poursuit par le rattachement de cet AA à un AI.

##### *3.1.1. Phase de création de l'AA*

L'expert au travers d'un agent expert ajoute une nouvelle base. Il doit spécifier :

- le type de la base (relationnelle ou objet)
- la localisation de cette dernière,

– le domaine, le système lui suggère les domaines existants gérés par les AIs : l'expert peut en choisir un ou en créer un nouveau.

Puis, l'agent expert crée un AA afin de référencer la nouvelle base locale. Le graphe d'objets descriptifs et les liens (liens de référencement, de dépendance et structurels objets (cf. paragraphe 2.2.2)) sont automatiquement créés.

L'expert peut à présent définir des liens sémantiques intra-base ainsi que des liens structurels. Il crée le lien de généralisation/spécialisation entre l'ODD employé et l'ODD chef. Un lien d'homonymie est également défini entre l'ODD nom dépendant de l'ODD employé et l'ODD nom dépendant de l'ODD projet.

Ensuite, l'AA exécute la phase de recherche de liens de similarité, d'échelle et de différence d'échelle intra-base s'ils n'ont pas été définis par l'expert du domaine à partir d'une équivalence sur le nom des ODDs.

Des liens de **similarités et d'échelle intra-base de type permanent** peuvent être créés entre deux ODDs de type attribut si les ODDs dont ils dépendent sont liés entre eux par des liens de spécialisation/généralisation. En effet, on ne peut avoir de différence d'échelle ni de liens d'homonymie lorsque les ODDs sont impliqués dans des liens de spécialisation/généralisation.

Trois liens de similarité permanents sont automatiquement créés grâce au lien de généralisation/spécialisation spécifiés par l'expert entre les ODDs chef et employé (cf. Figure 3 : liens détectés). Un lien d'échelle permanent est également créé entre l'ODD salaire dépendant de l'ODD chef et l'ODD salaire dépendant de l'ODD employé.

Des liens **de similarité et de différence échelle intra-base de type temporaire** sont créés s'il n'y a pas de liens d'homonymie ou d'échelle correspondant.

Si une unité de valeur est détectée au niveau des ODDs de type attribut, des liens de différence d'échelle intra-base temporaires sont supposés. En effet, on ne peut pas repérer les liens d'échelle uniquement sur l'équivalence entre les unités. Par exemple, le lien entre salaire (franc) et salaire (franc) n'est pas un lien d'échelle puisqu'il peut s'agir du franc français ou du franc suisse.

### 3.1.2. *Protocole d'enregistrement de l'AA à un AI*

Les communications entre les agents sont régies par des protocoles. « Un protocole est un ensemble de règles qui guident l'interaction entre plusieurs agents » (Huguet, 2001), (Mamouzi, 2001). Les deux langages de communication entre agents existant KQML<sup>1</sup> (Labrou, 1997) et FIPA ACL (Klusck 2001) présentent une structure de messages identique. Cependant, depuis l'an 2000, FIPA ACL est un standard très utilisé au sein des plate-formes agents et propose, de plus, une bibliothèque de protocoles. Nous nous appuyerons sur celle-ci pour décrire les interactions entre les agents dans ACSIS.

---

<sup>1</sup> KQML : Knowledge Query and Manipulation Language.

Le réseau d'accointances de l'AI est composé d'un ensemble d'AAs proches sémantiquement. Ce protocole détermine l'identité de l'AI qui enregistre l'AA dans son réseau d'accointances en fonction du nom de domaine (Contract Net Protocol : (Smith, 1980)). Si aucun AI ne dispose du nom de domaine de l'AA dans sa liste de domaines proches, le rattachement ne peut se faire et un nouvel AI est créé. Il est demandé à l'expert au travers de l'agent expert de spécifier le domaine de ce nouvel AI ainsi que tous les domaines proches sémantiquement qu'il couvre.

Lorsque l'AA s'est enregistré auprès de l'AI et que le réseau d'accointances est formé de plus d'un AA, l'AI établit une comparaison entre le nom des ODDs provenant de l'ontologie de l'AA rattaché et ceux provenant des ontologies des autres AAs de son réseau d'accointances. Ce protocole se base sur le *Fipa Query Protocol* qui consiste à demander à un agent l'exécution d'actions spécifiques.

Voici les différentes actions générées par le rattachement d'un AA à un AI :

- Envoi de noms d'ODD (*performatif inform*) : l'AA envoie le nom de chaque ODD à l'AI (dans le cas d'ODD attribut d'objet/relation, l'unité est également envoyée si l'attribut unité est non vide).

- Transmission aux AAs du réseau d'accointances (*performatif query\_ref*) : l'AI demande à chaque AA de son réseau d'accointances les noms d'ODDs équivalents aux noms d'ODDs de l'AA créé.

- Acceptation ou refus de traiter la demande de l'AI par chaque AA (*performatif agree ou refuse*).

- Traitement de la demande (*performatif inform ou failure*) : chaque AA renvoie tous les noms (et l'unité) de chacun de ses ODDs qu'il a détectés équivalents. L'AI crée des liens de similarité inter-bases temporaires entre chaque ODD de nom équivalent et des liens de différence d'échelle inter-bases temporaires entre chaque ODD de nom équivalent et disposant chacun d'un attribut unité.

Supposons que l'AA créé correspondant à la figure 2 se nomme divertissement et celui correspondant à la source de la figure 3 se nomme entreprise et que ce dernier se rattache au même AI que l'AA divertissement. Les liens de similarité inter-bases suivants sont créés suite aux rattachements de l'AA entreprise dans l'AI :

- nom de l'ODD employé de l'AA entreprise et nom de l'ODD personne de l'AA divertissement

- n°SS de l'ODD employé de l'AA entreprise et n°SS de l'ODD personne de l'AA divertissement

- nom de l'ODD chef de l'AA entreprise et nom de l'ODD personne de l'AA divertissement

- n°SS de l'ODD chef de l'AA entreprise et n°SS de l'ODD personne de l'AA divertissement

– nom de l'ODD projet de l'AA entreprise et nom de l'ODD personne de l'AA divertissement.

### **3.2. Protocole de traitement de requêtes multi-bases**

Plusieurs étapes importantes balisent le traitement d'une requête multi-bases. Au cours de ces étapes, on a recours à un ensemble de sous-protocoles décrivant les interactions entre les agents (Séguaran et al., 2005).

L'utilisateur au travers de l'interface d'un agent utilisateur saisit sa requête en pseudo SQL. Le mode « select from where » permet d'identifier et de différencier table et attribut. Toutes les tables ne sont pas obligatoirement spécifiées. Les jointures entre les tables ne doivent pas non plus être exprimées. Dans la clause « where » sont exprimées les conditions sur les attributs. Il n'y a pas d'ontologie de l'utilisateur (Mazuel et Sabouret 2006), l'utilisateur s'exprime avec les termes qu'il connaît dans son domaine.

#### **3.2.1. Transmission de la requête**

La requête écrite en pseudo SQL, est transmise de l'Agent Utilisateur à l'Agent Informationnel (*performatif request*). L'AI exploite les liens inter-bases de synonymie et envoie les requêtes à tous les AAs du réseau d'accointances (*performatif query\_ref*).

Soit la requête suivante à l'AI: `SELECT * FROM employé WHERE employé.salaire>1500 euros AND abonnement.type = "année"`. L'AI transmet la requête à chaque AA sans aucune modification car il ne peut exploiter aucun de ses liens.

#### **3.2.2. Evaluation sémantique de la requête**

##### **3.2.2.1. Connaissance sur les éléments de la requête**

Chaque AA évalue les objets descriptifs de données en fonction des éléments contenus dans la requête soumise (nom des attributs et des tables) et des liens de synonymie intra-base. Les AAs tiennent compte aussi des liens d'homonymie intra-base. Chaque AA compare l'unité de valeur des attributs sélectionnés dans la requête avec l'unité des ODDs disposant de la connaissance et crée des liens de différence d'échelle intra-base temporaires entre le terme de la requête et l'ODD. Chaque AA répond en fonction de sa capacité à traiter la demande.

L'AA entreprise détecte la connaissance sur employé (ODD de type relation), salaire (ODD attribut de relation de l'ODD employé) et type (ODD attribut de relation). L'AA entreprise crée un lien de différence d'échelle intra-base entre l'attribut de la requête salaire en euros et l'attribut salaire en francs. L'AA divertissement détecte de la connaissance sur abonnement (ODD type d'objet), sur

type (ODD attribut d'objet de l'ODD abonnement). Le lien de généralisation entre chef et employé ne permet pas de détecter la connaissance sur l'ODD salaire (attribut de relation de l'ODD chef).

#### 3.2.2.2. Recherche des liens de synonymie inter-bases des autres AIs

Si tous les AAs refusent de répondre, l'AI recherche les liens sémantiques d'un autre domaine en interagissant avec les autres AIs. Ces liens temporaires seront à valider in fine par l'Agent Utilisateur.

Nous avons pris l'option de communiquer avec un autre domaine seulement dans le cas où il n'y a pas de réponse dans le domaine concerné afin de ne pas surcharger l'utilisateur d'informations. De plus, il y a une forte présomption pour que des termes identiques employés dans un autre domaine soient des homonymes.

#### 3.2.2.3. Création de nouveaux liens de synonymie intra-base temporaires

Au cours de cette phase, chaque AA crée de nouveaux **liens de synonymie intra-base** temporaires à partir des liens de référencement et de dépendance existant en comparant les éléments de la requête aux noms des ODDs contenu dans son ontologie. Le lien est établi s'il n'y a pas de lien de non synonymie déjà créé entre ces deux termes. La méthode consiste à déceler des liens de synonymie et de similarité temporaires sans requérir l'intervention de l'Agent Utilisateur.

Des ODDs virtuels sont créés dans l'AA, ils représentent les termes de la requête qui n'ont pas été trouvés et qui peuvent être liés à des ODDs existants. Ils sont créés pour matérialiser les liens temporaires détectés. Ils sont virtuels car ils ne représentent aucun concept existant dans la base locale.

Chaque terme de la requête est comparé avec les ODDs de l'AA. Un ensemble est créé, constitué par les ODDs trouvés (ODD ayant le même nom qu'un élément de la requête).

Pour l'AA entreprise, l'ensemble est constitué des ODDs : employé, salaire et type.

Pour l'AA divertissement, il y a seulement l'ODD type.

A partir de l'ensemble des ODDs trouvés, le système va essayer de créer des liens temporaires entre les éléments non trouvés de la requête et des ODDs existants avec la méthode suivante :

1) Sélection des ODDs référencés par les liens de dépendance à partir des ODDs trouvés :

Dans notre exemple, afin de rechercher l'élément abonnement contenu dans la requête, à partir de l'ODD type trouvé : l'ODD projet est retenu car il est référencé par l'ODD type par un lien de dépendance. L'AA entreprise établit donc un lien de synonymie temporaire intra-base entre l'ODD projet et un ODD virtuel abonnement.

2) Si l'étape 1 n'a fourni aucune réponse pour un élément de la requête, les liens de dépendance et de référencement sont parcourus à partir d'un ODD trouvé afin de chercher l'élément. Si l'ODD recherché est de type relation ou objet, l'ODD est retenu si il n'y a pas d'attribut spécifié dans la requête ou si les attributs correspondent en parcourant le lien dépendance. Si l'ODD recherché est de type attribut, l'ODD est retenu si il n'y a pas de table spécifiée dans la requête pour cet attribut ou si la table spécifiée est trouvée en parcourant le lien de dépendance.

Un lien de synonymie intra-base temporaire n'est pas créé entre l'ODD virtuel abonnement et l'ODD équipe malgré les liens de dépendance et de référencement car lorsque un attribut est spécifié dans la requête (ici type), on vérifie si cet attribut est lié à l'ODD atteint. L'AA divertissement n'établit pas de lien de synonymie intra-base temporaire entre un ODD virtuel employé et l'ODD salarié car ce dernier n'a pas un ODD attribut salaire.

Chaque AA renvoie les noms des ODDs qu'il a trouvés ainsi que les liens intra-base de synonymie et de similarité temporaires détectés.

#### 3.2.2.4. Création de nouveaux liens de synonymie inter-bases temporaires

Après avoir reçu les réponses des AAs, l'AI crée **des liens de synonymie inter-bases** grâce à la diffusion des ODDs virtuels dans le réseau d'acointances (chaque AA renvoie alors le nom des ODDs qu'il détecte équivalents). Par exemple, l'ODD abonnement est présent dans l'AA divertissement. Un lien de synonymie inter-bases temporaire est créé entre l'ODD abonnement de l'AA divertissement et l'ODD projet de l'AA entreprise. A chaque création de lien de synonymie et de similarité intra-base ou inter-bases temporaire, il y a instanciation d'un lien à valider correspondant qui sera transmis des AAs ou AIs vers l'Agent Utilisateur.

#### 3.2.3. Découpage de la requête et envoi des sous-requêtes

L'AI ne sélectionne que le sous-ensemble d'AAs à contacter. Les sous-requêtes contiennent les éléments synonymes (grâce **aux liens de synonymie inter-bases temporaires** créés à l'étape précédente) qui remplacent les éléments de la requête. Pour les termes qui ne sont pas remplacés, la sous-requête ne conserve que les éléments de la requête sur lesquels l'AA dispose de connaissance (noms des ODDs renvoyés lors du 3.2.2.1). L'AI envoie les sous-requêtes aux AAs. Chaque AA exécute les sous-requêtes par le biais des ODDs en fonction des liens et répond en fonction de sa capacité à traiter la demande. Chaque sous-requête reçue par l'AA est modifiée en fonction :

- des liens de synonymie temporaires ou permanents,
- du modèle de la base locale (en OQL ou SQL) grâce à des fonctions de traduction dont dispose chaque AA,
- des liens de spécialisation et de généralisation : par exemple, si la requête est « *select \* from employé where employé.nom = 'dupont'* », la requête « *select \* from*

*chef where chef.nom = 'dupont' »*, est également exécutée étant donné le lien de spécialisation /généralisation entre chef et employé.

Chaque AA renvoie les données locales et les liens à valider intra-base. **La récupération des données locales** est réalisée grâce aux requêtes locales exécutées sur les bases dans le formalisme approprié. Au niveau local, lorsque des liens de **différences d'échelle intra-base de type permanent** existent, les données récoltées sont converties directement afin de les présenter dans le format attendu par l'Agent Utilisateur si des fonctions de conversion sont stockées au niveau des ODDs et si une unité est spécifiée dans la requête. Dans le cas d'un lien de **différence d'échelle intra-base de type temporaire**, l'Agent Utilisateur en sera averti et pourra insérer une fonction de conversion après avoir validé ce lien. Les liens de différences d'échelle intra-base créés concernant la différence d'échelle entre le terme attribut soumis dans la requête et l'ODD concerné sont également envoyés sous forme de liens à valider par l'Agent Utilisateur. Les éléments de réponses sont stockés au niveau de chaque AI jusqu'à ce que toutes les réponses des AAs contractants soient parvenues. Chaque AI renvoie à l'Agent Utilisateur les résultats qu'il a obtenus (résultats et liens intra-base à valider) et les liens à valider inter-bases, dont les **liens de différences d'échelle inter-bases temporaires**.

L'AA entreprise reçoit la sous-requête suivante : `SELECT * FROM employé WHERE employé.salaire > 1500 AND projet.type= "année"`. L'AA divertissement reçoit : `SELECT * FROM abonnement WHERE abonnement.type = "année"`. Comme aucune connaissance n'a été trouvée pour employé, le nom de la table retenu est celle se trouvant dans la clause where.

#### *3.2.4. Validation des résultats et mise à jour des liens correspondants au niveau de l'AA ou de l'AI.*

L'Agent Utilisateur organise les résultats pour chaque base locale (pour chaque AA) et les liens sémantiques exploités à valider (intra-base et inter-bases). Les liens de synonymie, de similarité et de différence d'échelle sont mis à jour (de temporaire à permanent) ou supprimés (cf. Figure 4). Dans ce dernier cas, les liens correspondants sont créés (synonymie en non synonymie, similarité en homonymie et différence d'échelle en échelle). Pour chaque résultat de sous-requêtes, l'Agent Utilisateur peut insérer de nouveaux liens de synonymie en fonction des résultats satisfaisants ou non pour permettre une re-exécution de la requête avec des résultats plus satisfaisants.

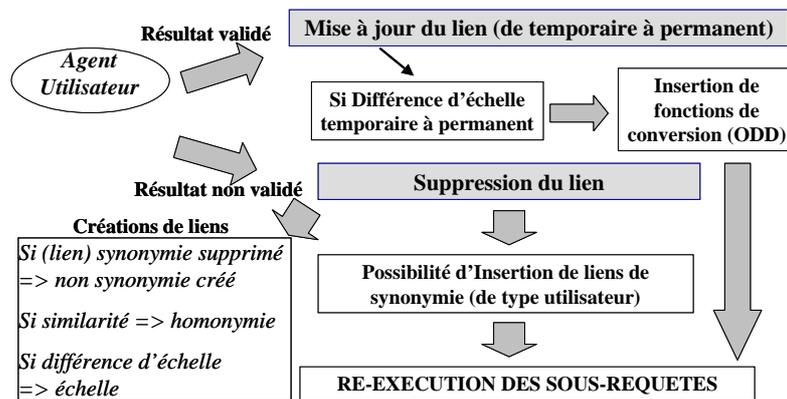
Chaque AA envoie les résultats ainsi que les liens à valider aux AIs. Dans notre exemple l'AA entreprise envoie le lien de synonymie entre projet et abonnement, le lien de différence d'échelle intra-base entre l'attribut de la requête et l'attribut salaire. L'AA divertissement n'envoie aucun lien. L'AI transmet l'ensemble à l'agent utilisateur. Les résultats provenant de l'AA entreprise ne satisfont pas l'utilisateur et les liens de synonymie intra-base et inter-base sont invalidés. Le lien de différence d'échelle est par contre lui validé, il devient donc permanent. L'utilisateur doit insérer une fonction de conversion afin de convertir les francs en

euros. Cette dernière est envoyée à l'AI qui l'envoie ensuite à l'AA entreprise. Elle sera ajoutée à l'ODD salaire dont l'unité est le franc. Suite à l'invalidation du lien de synonymie, un lien de non synonymie est créé entre l'ODD virtuel abonnement et l'ODD projet.

Les résultats concernant l'AA divertissement intéresse l'utilisateur, aucun lien ne l'accompagnant, l'exécution s'arrête.

Lorsque l'Agent Utilisateur insère un nouveau lien, une ou plusieurs créations seront effectuées par l'AA ou l'AI concerné :

- un lien inter-bases de type utilisateur est créé si les deux termes concernés se trouvent dans deux AAs différents,
- un lien intra-base de type utilisateur est créé si les deux termes se trouvent dans le même AA. On crée un lien intra-base à l'aide d'un ODD virtuel dès qu'un terme se trouve dans un AA.



**Figure 4.** Processus de validation des liens

Des sous-requêtes sont re-exécutées dans plusieurs cas (cf. Figure 4) :

- lorsque des liens intra-base ou inter-bases ne sont pas validés, les sous-requêtes sont retournées au niveau des AAs ou des AIs,
- lorsque des liens de synonymie sont insérés par l'utilisateur : seules les parties modifiées de la requête sont renvoyées vers les AAs, les résultats étant conservés au niveau de chaque AI,
- lorsque des fonctions de conversion sont insérées par l'utilisateur (envoyées sous forme de liens validés vers les AIs qui transmettent aux AAs), la sous-requête correspondante doit être re-exécutée.

A la ré-exécution de la requête, l'AA entreprise récupère les employés dont le salaire en franc est supérieur à  $6,55957 \times 1500$  euros.

Supposons que l'utilisateur exprime la requête suivante : `SELECT * FROM employé WHERE abonnement.type = "année"`. L'exécution auprès de l'AA divertissement est différente car à l'aide du lien de référencement le lien de synonymie intra-base entre l'ODD salarié et l'ODD virtuel abonné est créé : aucun attribut n'étant mentionné dans la requête tout ODD référencé peut être utilisé. L'AA entreprise crée également un lien de synonymie intra-base entre l'ODD virtuel abonnement et l'ODD équipe.

### 3.3. Modification d'une base de données locale

Différentes modifications peuvent être apportées à un schéma de base de données, nous allons spécifier les conséquences sur les ODDs.

	Ajout	Suppression	Modification
Relation /Classe	Ajout de l'ODD correspondant	Suppression de l'ODD et mise à jour des liens le contenant	De son nom, Ajout attribut, Suppression attribut
Attribut	Ajout de l'ODD correspondant	Suppression de l'ODD et mise à jour des liens le contenant	De son nom, De son type (modification de l'ODD), De son unité

– Ajout :

Le processus de création de liens de similarité et de différence d'échelle intra-base est exécuté comme lors de l'enregistrement d'une nouvelle base (cf. 3.1).

– Suppression :

Les liens intra-base le concernant sont supprimés car l'ODD n'existe plus dans la base. Les liens inter-bases sont eux conservés car il s'agit de connaissances globales. Le fait qu'un ODD soit un synonyme, un homonyme, ... de tel autre ODD d'une autre base, est une information qui doit être conservée au niveau de l'AI afin de pouvoir toujours l'utiliser lors de l'exécution d'une nouvelle requête. Si cette information est supprimée, le système ne connaîtra, par exemple plus les synonymes d'un terme qui existent encore dans d'autres bases. Donc lors de l'exécution d'une requête si c'est le terme supprimé qui est employé, l'AI ne communiquera pas la requête aux AAs détenant le synonyme car il ne détiendra plus cette information. Une fois le lien supprimé, il est impossible de le retrouver. Par exemple, supposons qu'un AA contienne l'ODD rémunération et que lors de l'exécution de requêtes un lien de synonymie inter-bases ait été établi et validé entre cet ODD rémunération et l'ODD salaire d'un autre AA. Par la suite, l'ODD rémunération est supprimé de la base et donc de l'AA. Lors de l'exécution d'une requête demandant la rémunération d'une personne, si le lien inter-base est conservé l'AI communiquera la requête à l'AA contenant l'ODD salaire. Si le lien inter-bases est supprimé et que l'AA n'a

pas d'ODD personne, ni d'ODD rémunération, la requête ne lui sera pas communiquée.

– Modification :

*Modification de nom d'un attribut, d'une classe ou d'une relation.*

Modification de l'ODD concerné, création d'un ODD virtuel avec l'ancien nom et création d'un lien de synonymie intra-base entre les deux ODDs.

*Modification de l'unité d'un attribut*

Modification de l'ODD et mise à jour des liens intra-base d'échelle et de différence d'échelle.

Les modifications apportées sont envoyées à l'AI et à son tour ce dernier les envoie aux AAs de son réseau d'acointances. Les modifications sont donc répercutées dans tous les AAs et dans l'AI concerné comme lors d'un enregistrement d'une nouvelle base.

#### **4. Travaux voisins**

Plusieurs projets utilisent les ontologies et les SMAs. Dans (Toivonen et Helin, 2003), une ontologie est utilisée pour décrire les protocoles d'interactions. Le protocole des agents évolue dynamiquement avec ces ontologies. Dans notre système multi-agents, les protocoles d'interactions n'évoluent pas, ce sont les ontologies contenues dans les agents qui évoluent.

Le projet DASMAS (Orgun et al., 2005) repose sur un framework de dialogue pour résoudre l'interopérabilité sémantique dans des systèmes multi-agents.

DASMAS travaille avec plusieurs SMAs. A chaque SMA est associée une ontologie de domaine. Le processus de résolution de conflits utilise un protocole de négociation et le lexique WordNet pour trouver les concepts similaires sémantiquement dans les ontologies hétérogènes.

Différents systèmes prennent en compte l'évolution des ontologies, cette évolution est gérée de façon différente :

- soit à l'intérieur de l'ontologie,
- soit à travers des versions d'ontologie.

Les problèmes d'évolution et de versionnement des ontologies sont très différents de ceux traités dans les bases de données relationnelles (Noy et Klein, 2003). Ces auteurs définissent l'évolution et le versionnement d'ontologies comme « la capacité à gérer les modifications d'ontologies et leurs effets en créant et maintenant différentes variantes de l'ontologie ».

Les systèmes gérant des versions d'ontologies sont également différenciables par leur façon de maintenir ou non les différences entre deux versions successives. Le système SHOE (Heflin et Hendler, 2000) ne garde pas traces des modifications

réalisées entre deux versions. Une copie du fichier est effectuée et les modifications sont faites directement dans le nouveau fichier. Au contraire le système Protégé (Liang et al., 2005) enregistre les changements à l'intérieur de l'ontologie et les conserve pour de futures comparaisons. Le système OntoView (Klein et al., 2002) met à disposition des outils pour comparer des versions d'une ontologie afin de proposer les différences entre deux versions. D'autres outils permettent aux utilisateurs d'établir des relations conceptuelles entre des versions.

Dans notre système, les modifications sont insérées directement dans l'ontologie car elles concernent son évolution et est commune à l'ensemble des utilisateurs. Dans de prochains travaux, nous utiliserons les versions d'ontologies afin de prendre en compte les changements apportées par les utilisateurs. En effet, on ne peut pas être sûr de la compétence des utilisateurs à valider ou invalider un lien sémantique. Les changements effectués par ces derniers dans des versions permet de conserver l'ontologie intacte et de prendre en compte le changement. Par la suite, après plusieurs validations ou invalidations le changement peut être intégré directement dans l'ontologie. Nous garderons trace des modifications faites afin de pouvoir effectuer des fusions de versions de la même ontologie. Le système de Auer et Here (Auer et Here, 2006) proche de cette exigence, garde trace des différentes versions d'une ontologie car cela offre la possibilité d'effectuer des opérations de fusions ou de constructions de sous-hiérarchies.

## **5. Bilan et perspectives**

La résolution des conflits sémantiques par les agents consiste à exploiter les liens et les ODDs des ontologies. L'ensemble des protocoles définis permet de faire évoluer les ontologies de domaine en détectant de nouveaux liens sémantiques. La dynamique du système provient non seulement de la détection de nouveaux liens lors du traitement des requêtes mais aussi de sa capacité à suivre l'évolution (modification, ajout, suppression) des bases locales. Les protocoles ont été modélisés à l'aide de AUML (Agent UML) (Odell *et al.*, 2000). Le prototype réalisé sur JADE est opérationnel et a permis de valider notre proposition en implantant les principaux protocoles conformes au standard de la Fipa (Séguran, 2003). JADE (Java Agent Development Framework) fournit une API pour développer un système multi-agents (Bellifemine, 2002). Pour le transfert d'objets (des liens et des requêtes) dans le contenu des messages échangés entre les agents, nous avons utilisé les ontologies de description de messages proposées par JADE. La description d'ontologie au sens de Jade est une solution donnée pour le transport d'objets et n'a pas de rapport avec les ontologies que nous avons décrit précédemment. Nos agents s'enregistrent ainsi auprès de cette ontologie nommée RequeteOntology. L'ontologie Jade est une instance de jade.content.onto.ontology dont les schémas définissent les prédicats et les concepts pertinents. Cette collection de schémas constitue la structure commune d'interaction entre les agents et n'évolue pas au cours du cycle de vie des agents. La classe RequeteOntology contient tous les objets

nécessaires lors de l'envoi des performatifs : liens à valider, requête, sous-requête, .... Le prototype est opérationnel et a permis de valider nos hypothèses et notre système : un système multi-agents dont les différents protocoles font évoluer les ontologies de domaine. Il est composé d'une quarantaine de classes et contient environ 5000 lignes de codes. Il n'a été validé que sur des petites bases, une validation sur un cas réel est envisageable mais demanderait au préalable une évolution des interfaces utilisateur car ces dernières sont peu conviviales et non graphiques. Face à de nombreuses informations à afficher, cela ne sera plus lisible et peut être même incompréhensible.

Les réponses provenant des différents AAs et ensuite des AIs sont proposés à l'utilisateur au travers de l'agent utilisateur sans aucune consolidation. Les différentes réponses sont fournies en explicitant les liens temporaires à valider ou à invalider en précisant le nom de l'AA et de l'AI d'où elles proviennent. Nous envisageons un processus d'homogénéisation et de restructuration des résultats provenant de différentes sources locales au niveau de l'AI afin d'essayer de gérer des incohérences et/ou contradiction provenant de AAs différents.

Pour l'instant, notre agent utilisateur n'a pas d'ontologie, il ne conserve aucune information provenant des ses anciennes demandes. Nous envisageons de lui associer une ontologie afin qu'il détienne de la connaissance. De ce fait, le contenu de ses requêtes sera plus ciblé et plus pertinent.

Nous pensons appliquer le concept de version d'ontologie afin de ne pas prendre en compte directement les modifications des ontologies faites par les utilisateurs comme nous le faisons actuellement. Les modifications proposées par l'utilisateur deviendront publiques seulement après leurs validations par un super utilisateur ou un administrateur. Les versions d'ontologie devront garder traces des différentes modifications pour mieux suivre leurs évolutions et afin de propager les différentes modifications à appliquer sur les ontologies liées.

## **6. Bibliographie**

- S. Auer and Herre H., « A Versioning and Evolution Framework for RDF Knowledge Bases », In Proceedings of Ershov Memorial Conference, 2006.
- Bellifemine F., « Jade and beyonds ». Presentation at AgentCities Information Day 1, Lausanne, February 2002.
- Boulanger D, Dubois G., « Advanced Object Approach for Information System Cooperation ». *Encyclopedia of microcomputers*, Vol. 27, supplement 6, A. Kent, J.G. Williams Executive (editors), published by Marcel Dekker Inc. New York, 2001.
- Couturier V., Séguran M., « Patterns and Components to Capitalise and Reuse a Cooperative Information System Architecture », *International Conference on Enterprise Information System Architecture ICEIS 2003*, Angers, April 23-26 2003.

- Cullot N., Parent C., Spaccapietra S., Vangenot C., « Des SIG aux ontologies géographiques », *Revue internationale de géomatique*. Vol. 0, n° 13/ 3, 2003.
- Ferber J., « Les systèmes multi-agents : vers une intelligence collective », InterEditions, 1995.
- Gal A., « Semantic Interoperability in Information Services: Experiencing with CoopWARE ». *In ACM Sigmod record* vol 28, n°1, 1999.
- Gruber T.R., « A translation approach to portable ontologies ». *Knowledge acquisition*, vol. 5, n°2, , p. 199-220, 1993.
- Guarino N., « Understanding, Building, And Using Ontologies ». *International Journal of Human-Computer Studies*, vol. 469, n°2-3, Feb./March, 1997.
- Heflin J. and Hendler J., « Dynamic Ontologies on the Web », In Proceedings of the seventeenth national conference on Artificial Intelligence, AAAI 2000, Austin, 2000.
- Huget M.P., « Une ingénierie des protocoles d'interaction pour les systèmes multi-agents », Thèse de doctorat, Université Paris IX Dauphine, Paris, 2001.
- Kahng, J., MacLeod, D., « Dynamic Classificational Ontologies ». In *Computing the Brain : A Guide to Neuroinformatics*. Michael A. Arbid, and Jeffrey Grethe, Editors, San Diego. Academic press, 2000.
- Klein M, Noy N., « A Component-Based Framework for Ontology Evolution ». Technical Report, IR-504, CS Dept, Vrije Universiteit, Amsterdam, 2003.
- Klein M., Kiryakov A., Ognyanov D. and Fensel D., « Ontology versioning and change detection on the web », In 13 th International Conference on Knowledge Engineering and Knowledge Management, EKAW02, Spain, 2002.
- Klusck, M., « Intelligent Agent Technology for the Internet : A Survey. Journal on Data and Knowledge Engineering ». Special Issue on Intelligent Information Integration, D. Fensel (Editor), Vol. 36, n°3, Elsevier Science, 2001.
- Liang Y. Alani H. and Shadbolt N.R., « Ontology change management in Protégé », Proceeding of Advanced Knowledge Technologies Doctoral Colloquium, United Kingdom, 2005.
- Mamouzi H., « Ingénierie des protocoles d'interaction : des Systèmes Distribués aux Systèmes Multi-agents ». Thèse de doctorat, Université Paris IX Dauphine, 2001.
- Mazuel L., Sabouret N., « Une interpretation générique de commande en langue naturelle dans les systèmes de dialogues à base d'ontologies », WACA' 2006 Second Workshop sur les Agents Conversationnels Animés, IRIT, Toulouse, octobre 2006.
- Mena E., Illarramendi A., Kashyap V., Sheth A., « OBSERVER : An approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies ». *International journal on Distributed And Parallel Databases DAPD*, vol.8, n°2, 2000.
- Mena E., Illarramendi A., « Ontology-Based Query Processing for global Information Systems ». Kluwer Academic Publishers, June, 2001.
- Nodine M. et al., « Active Information Gathering in Infosleuth ». *IJICIS*, vol. 9, n°1/2, 2000.

- Noy N.F. and Klein M., « Ontology evolution: Not the same as Schema Evolution », *Knowledge and Information Systems*, 5, 2003.
- Odell J., Parunak H., Bauer B., « Extending UML for Agents ». G. Wagner, Y. Lesperance, E. Yu (editors), *Proceedings of Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence*, Icue Publishing, Austin, Texas, 2000.
- Orgun B., Dras M., Nayak A., « DASMAS – Dialogue based Automation of Semantic Interoperability in Multi Agent Systems ». Australian Ontology Workshop, AOW'2005, Sydney, Australia, 2005.
- Séguran M., « Résolution des conflits sémantiques dans les systèmes d'information coopératifs ». Thèse de doctorat, Université Jean Moulin, Lyon, 2003.
- Sheth A., « Changing Focus on Interoperability in Information Systems: from System, Syntax, Structure to Semantics », In *Interoperating Geographic Information Systems*. M F. Goodchild, M.J. Egenhofer, R. Fegeas, C.A. Koffman (editors), Kluwer Academic Publishers, 1999.
- Smith R.G., « The Contract Net Protocol : High-level Communication and Control in a Distributed Problem Solver », *IEEE Transactions on Computers*, vol. 29, n°12, p. 1104-1113, 1980.
- Séguran M., Talens G., Boulanger D., « Domain Ontologies Evolutions To Solve Semantic Conflicts », ODBIS 2005, VLDB Workshop on Ontologies-based techniques for DataBases and Information Systems, Trondheim, Norway, September 2-3, 2005.
- Toivonen S. and Helin H., « Representing Interaction Protocols in DAML », In van Elst, L., Diagnum, V., and Abecker, A. (Eds.): *Agent-Mediated Knowledge Management: Selected Papers from AAAI 2003 Spring Symposium*, volume 2926 of Lecture Notes in Artificial Intelligence, Berlin, Germany, 2003.