

UNE APPROCHE D'INTEROPERABILITE SEMANTIQUE VIA LES SERVICES WEB DANS LES PROGICIELS DE GESTION INTEGREE

MEROUANE ZOUBEIDI, OKBA KAZAR, SABER BENHARZALAH, NADJIB MASBAHI

Laboratoire d'Informatique Intelligente, Département Informatique,
Université Mohamed Khider de Biskra, Algérie

ABSTRACT

Aujourd'hui, le PGI (Progiciel de Gestion Intégrée) est devenu la pierre angulaire des systèmes d'information des entreprises. Cet outil permet une gestion homogène et cohérente du système d'information (SI) de l'entreprise. Notre objectif est de rendre ce PGI interopérable avec des solutions applicatives périphériques, offrant ainsi plus de fluidité pour l'information et avec les autres solutions du marché, à savoir les CRM (Customer Relationship Management) et les SCM (Supply Chain Management). L'interdépendance s'accroît et entraîne des besoins forts d'intégration en assurant un partage et échange d'informations réel et efficace au niveau sémantique dans un environnement ouvert et dynamique. Le travail que nous proposons a pour finalité l'intégration de la sémantique au niveau des ERPs grâce aux techniques proposées par le système Multi-Agents, et le recours aux nouvelles technologies pour résoudre la plupart des conflits sémantiques.

KEYWORDS : Progiciel de gestion Intégré (PGI), Systèmes Multi-Agents (SMA), Ontologie, Interopérabilité Sémantique.

1 INTRODUCTION

L'évolution technologique d'une part et les mutations économiques d'autre part ont bouleversé le domaine des systèmes d'informations. L'accès des entreprises aux nouvelles technologies, à Internet en particulier, tend à modifier la communication entre les différents acteurs du monde des affaires. Notamment entre l'entreprise et ses clients (Business To Consumer, B2C), le fonctionnement interne de l'entreprise (Business To Employees, B2E) et la relation de l'entreprise avec ses différents partenaires et fournisseurs (Business To Business, B2B). On appelle aussi "e-Business" l'intégration au sein de l'entreprise d'outils basés sur les technologies de l'information et la communication, en l'occurrence les Progiciels de Gestion Intégrée (PGI).

La coopération de ces progiciels est devenue aujourd'hui plus qu'une nécessité vu le besoin d'échanger des informations. Cependant, ceux-ci ont été développés indépendamment et ne partagent pas ainsi la même sémantique pour la terminologie de leurs modèles d'applications. Pour que cette entreprise puisse communiquer et coopérer de manière efficace, fiable et performante, il est utile de rendre son PGI interopérable [1] en premier lieu entre eux même, puis avec des solutions applicatives périphériques (module de gestion non couvert par un PGI en place), offrant ainsi plus de fluidité pour l'information et enfin avec les autres solutions du marché, à savoir les CRM (Customer Relationship Management) et les SCM (Supply Chain Management). L'interdépendance s'accroît et entraîne des besoins forts d'intégration en assurant un partage et échange d'informations et de services dans un environnement ouvert et dynamique [2]. Donc, il

s'agit d'une interopérabilité intra-entreprise et interentreprises. Dans ce cadre, le traitement de l'interopérabilité est une tâche coûteuse parce qu'il doit considérer l'hétérogénéité des données, de services (application et fonction) et de processus.

Les paradigmes d'architecture orientée service [3], et des services Web [4] se sont imposés rapidement, et ont fortement changé l'orientation des entreprises du fait des avantages qu'ils fournissent. Ils offrent certes des solutions d'interopérabilité des applications des entreprises au niveau technique, mais ne résolvent pas la problématique liée à l'interopérabilité sémantique qui constitue l'enjeu majeur pour la coopération des PGI. En effet, les hétérogénéités des données et les fonctionnalités échangées entre les services Web sont complexes et rendent leur interopérabilité difficile. Par voie de conséquence, il est impératif de capturer la sémantique liée aux informations échangées, et de résoudre les conflits causés par les hétérogénéités sémantiques.

Cet article présente une architecture à base d'agents et ontologies pour l'interopérabilité sémantique des ERPs. Il propose un bon usage de l'information et de services disponibles. L'idée que nous défendons s'articule autour de contribution associées aux deux dimensions principales : L'amélioration de notre GAACSM [5] pour supporter un cas précis de l'ERP et l'extension de notre travail [6] pour fournir la médiation sémantique de données et de services. Le reste de papier est organisé comme suit. Dans la section 2, plusieurs travaux connexes seront évoqués avec leurs contextes, les hypothèses principales et les propositions. La section 3 dresse un panorama sur la technologie émergente liée à notre contexte et présente la justification de notre

choix. Dans une quatrième section, nous présentons notre approche ainsi que le processus de l'interopérabilité sémantique dans les ERPs fondés sur les ontologies. Ensuite, la section 5 met en relief les aspects techniques de l'implémentation. Enfin, la section 6 conclut l'article en présentant les perspectives de recherches.

2 TRAVAUX CONNEXES

Nous trouvons dans la littérature une panoplie de travaux qui traite l'interopérabilité sémantique. Pour les présenter, il est indispensable de scinder ces approches en deux catégories selon les techniques utilisées, d'évaluer sur un certain nombre de critères relatifs à la modélisation de la sémantique, de dresser un bilan et de proposer une nouvelle architecture qui prend en considération la modélisation de la sémantique pour l'interopérabilité des ERPs. Nous présentons en premier lieu, les travaux basés sur les systèmes multi-agents et ontologies ainsi que ceux relatifs aux services web et ontologies.

Une première catégorie s'articule sur les systèmes multi-agents et ontologies. Pour rappel, le paradigme agent donne une nouvelle vue pour le développement des systèmes de nature complexe, hétérogène, distribué et/ou autonome [7]. Plusieurs projets d'interopérabilité sémantique utilisent celui-ci, à savoir [8, 9, 10, 11]. Le projet Infosleuth [12] a pour but d'implémenter un ensemble d'agents coopératifs qui découvrent, intègrent et présentent l'information en fonction des besoins d'un utilisateur ou d'une application pour lesquels ils produisent une interface simple et cohérente. L'architecture du projet Infosleuth est constituée d'un ensemble d'agents collaboratifs communiquant par l'ACL KQML. Les utilisateurs expriment ses requêtes sur une ontologie spécifique en utilisant KIF et SQL. Les requêtes sont ensuite expédiées aux agents spécialisés (agent broker, ontologique, planner,...) pour la recherche des données sur des sources distribuées et pour l'intégration. La résolution de nombreux conflits sémantiques reste guidée par l'utilisateur [13]. Ils emploient des agents spécialisés pouvant être considérés comme des threads, qui sont loin de la définition de l'agent cognitif comme défini en intelligence artificielle distribuée. Les auteurs [9] proposent un système multi-agents pour assurer l'interopérabilité sémantique et résoudre les conflits sémantiques liés à l'évolution des ontologies de domaine. Dans cette approche, le traitement des requêtes et la validation des mappings sont complètement liés aux utilisateurs. Kim W. et al. [10] présentent des agents pour la méta-recherche intelligente des produits par la considération de multiples attributs, en employant des mappings d'ontologies et les services web. Ce travail est dédié au champ du E-commerce. Leclercq et al. [14] exposent une solution de médiation basée sur une architecture multi-agents adaptée aux spécificités des SIG (Systèmes d'information géographiques). L'architecture est de type médiation de contextes, mono-domaine et basée sur une ontologie commune pour représenter la sémantique d'une manière explicite. Les auteurs [5] présentent une architecture GAACSM (Generic Architecture based Agent

for Context and Schema Mediation). Elle est basée sur une approche multi-agent pour la médiation sémantique des systèmes d'information dans un contexte multi-domaines. Elle combine l'approche de la médiation de schéma et de celle basée sur le contexte. Elle assure une meilleure transparence, un partage sémantique fiable de données et une meilleure scalabilité.

Les approches proposées ci-dessus ont pour points communs la volonté d'apporter des outils, des architectures, des méthodologies au problème de l'interopérabilité sémantique des systèmes d'information. En dépit du nombre considérable des solutions déjà proposées, certains aspects sont imprécis. Chacune des solutions existantes fournit des avantages aux dépens d'autres, et chacune d'elles ne traite qu'une partie des conflits de données sans prendre en considération le partage de services (Applications et fonctions). Concernant le niveau des conflits sémantiques, la plupart des projets ne résolvent que les conflits de nommage. Aussi, la résolution des conflits des valeurs est guidée par l'utilisateur [12]. Elle possède également une faible scalabilité face à l'évolution du nombre de participants. Dans tous ces cas sus-évoqués, dans le cadre de la coopération de systèmes d'information hétérogènes, la quasi-totalité des architectures d'interopérabilité ont recours au paradigme objet [14] sans prendre en compte le paradigme agent.

Une deuxième catégorie se focalise sur les web services et ontologies. Les auteurs [15] présentent une architecture orientée services sémantiques WSMX. Cette dernière est un environnement d'exécution des services Web basée sur le Web Services Modeling Ontology [16]. Elle permet la découverte dynamique, l'invocation et la composition des services Web sémantiques, elle est utilisée par les fournisseurs pour enregistrer et offrir leurs services, et par les clients pour découvrir dynamiquement et invoquer les services sélectionnés. WSMX assure l'interopérabilité des systèmes B2B (Business to Business), elle-même se présente en tant que service Web. Izza [17] présente une architecture ODSOIA (Ontology Driven Service Oriented Architecture), qui permet l'intégration des applications des systèmes d'information industriels. Elle constitue une solution intra-entreprise qui augmente le bus de services d'entreprise ESB (pour Entreprise Service Bus) de deux couches architecturales (la couche sémantique et la couche d'intégration) en plus de la couche orientée services. L'architecture sémantique comprend un ensemble d'ontologies globales, de domaines et locales, des ontologies de services et de mappings concernant les processus, les fonctions et les données. Pour décrire sémantiquement les services Web, les auteurs utilisent OWL-S [18]. Par ailleurs, Mrissa [19] propose une architecture de médiation sémantique orientée contexte pour la composition de services Web. Il s'agit d'une médiation de données prise en charge par un service Web médiateur inséré entre le service Web émetteur et le service Web récepteur lors de l'exécution de la composition. Ces services sont préalablement décrits sémantiquement en utilisant les éléments d'extensibilité du WSDL et une ontologie de domaine à laquelle sont associées des

ontologies contextuelles.

Les approches sus-indiquées recourent à une description sémantique des services soit par langage de description sémantique [15, 17] ou annotation des langages existants en exploitant leurs éléments d'extensibilité [19]. Ces annotations présentent plus d'avantages que les langages de description car elles sont relativement simples et exploitent les normes existantes des services Web. Notons qu'à part les travaux de [19], tous les autres supposent l'adaptation de la sémantique locale des services Web à la sémantique de l'ontologie partagée, pour pouvoir effectuer la médiation.

Pour notre part, afin de permettre à une entreprise de coopérer d'une manière très fiable avec les autres organisations sans échec, nous optons pour les annotations des descriptions des services Web à l'aide du langage récent SAWSDL [21] qui constitue une recommandation du W3C. L'approche que nous proposons et l'architecture qui en découle sont inspirées en premier lieu des travaux de Mrissa [19]. Puis, nous étalons nos travaux [6, 20] pour intégrer la médiation sémantique de service dans le cadre de l'interopérabilité inter-ERP.

3 TECHNOLOGIE EMERGENTE

Dans cette section, nous présentons certaines notions qui sont directement liées au contexte de notre de travail en mettant en relief leur champs d'intervention.

3.1 Progiciel de gestion intégrée

La référence [2] définit une solution de gestion intégrée, également dénommée ERP (Entreprise Resource Planning) ou PGI comme étant :

- Une solution logicielle qui permet de gérer plusieurs processus fonctionnels ou opérationnels d'une organisation (gestion des RH, gestion comptable et/ou financière, des achats, des approvisionnements, de la production, des stocks, des ventes, de la relation client, de la chaîne logistique, etc.).
- Les processus gérés sont intégrés de manière modulaire (module fonctionnel par module fonctionnel), tout en partageant une base de données unique et commune.
- La solution doit couvrir au moins deux domaines fonctionnels différents (par exemple, RH et comptabilité/finance, ou encore comptabilité/finance et gestion commerciale...).
- Elle peut constituer le socle du système d'information de l'organisation si elle couvre la quasi-totalité des processus fonctionnels clés de celle-ci.

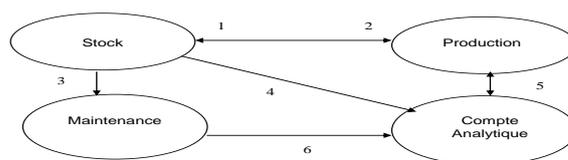


Figure 01 : Un progiciel de gestion intégrée

- Transférer les quantités produites vers l'application Stock.
- Utiliser les quantités de matière première pour la production des produits finis.
- Utiliser les pièces de rechange pour la réparation des équipements.
- Génération des charges de consommation stock pour analyse des coûts.
- Utiliser la production pour calculer le résultat des différents ateliers de production
- Génération des charges de maintenance pour analyse des coûts des équipements.

En outre, le PGI est caractérisé d'un autre principe fondateur permettant l'usage de ce qu'on appelle un moteur de Workflow et qui permet, lorsqu'une donnée est enregistrée dans le SI, de la propager dans les modules qui en ont l'utilité (cf. Figure 1), selon une programmation prédéfinie.

3.2 Justification de choix (Ontologie & Agent)

Pour résoudre les conflits issus de l'hétérogénéité sémantique, l'utilisation de l'ontologie [20], à ce niveau, permet la description de concepts et de relations qui peuvent exister entre eux dans un domaine donné. Le langage OWL est un standard très utilisé pour la description des ontologies sur le Web (Dans le cadre de la relation client ou de la relation citoyen, qui renforce le besoin de communication entre les systèmes back-office -tel que le PGI- et les applications de front-office -tel qu'un site de commerce en ligne-). Par ailleurs, l'utilisation de l'agent, à ce niveau, permet la construction d'une architecture ouverte et évolutive. Les agents instaurent un environnement flexible (agents capables de répondre à temps, proactifs, communicants) dans le but de traiter les requêtes adressées à l'architecture coopérative. Dans ce cadre, les agents disposent de connaissances sémantiques qui sont décrites sous forme de l'ontologie locale de données pour chaque module fonctionnel. La constitution des ontologies et agents est explicitée dans la prochaine section.

4 ARCHITECTURE PROPOSEE

L'objectif de notre solution est la réalisation d'une médiation sémantique ayant les caractéristiques suivantes :

- Permettre aux différents systèmes participant à la coopération de ne pas modifier leurs fonctionnalités habituelles en respectant l'autonomie du PGI ;
- Permettre à un utilisateur, client et fournisseur d'accéder de manière transparente à des ressources fournies par les PGIs ;
- Construire un système de médiation sémantique flexible, permettant l'ajout de fonctionnalités et de services ;
- Résoudre la majorité des conflits sémantiques par le recours aux nouvelles technologies, à savoir les ontologies ;
- Permettre à un nouveau module fonctionnel -non couvert par le PGI- ou bien un autre PGI qui fournit son contexte, de trouver le PGI, les données et les services partagés par le module fonctionnel, de les intégrer dynamiquement au PGI et de les utiliser de manière transparente.

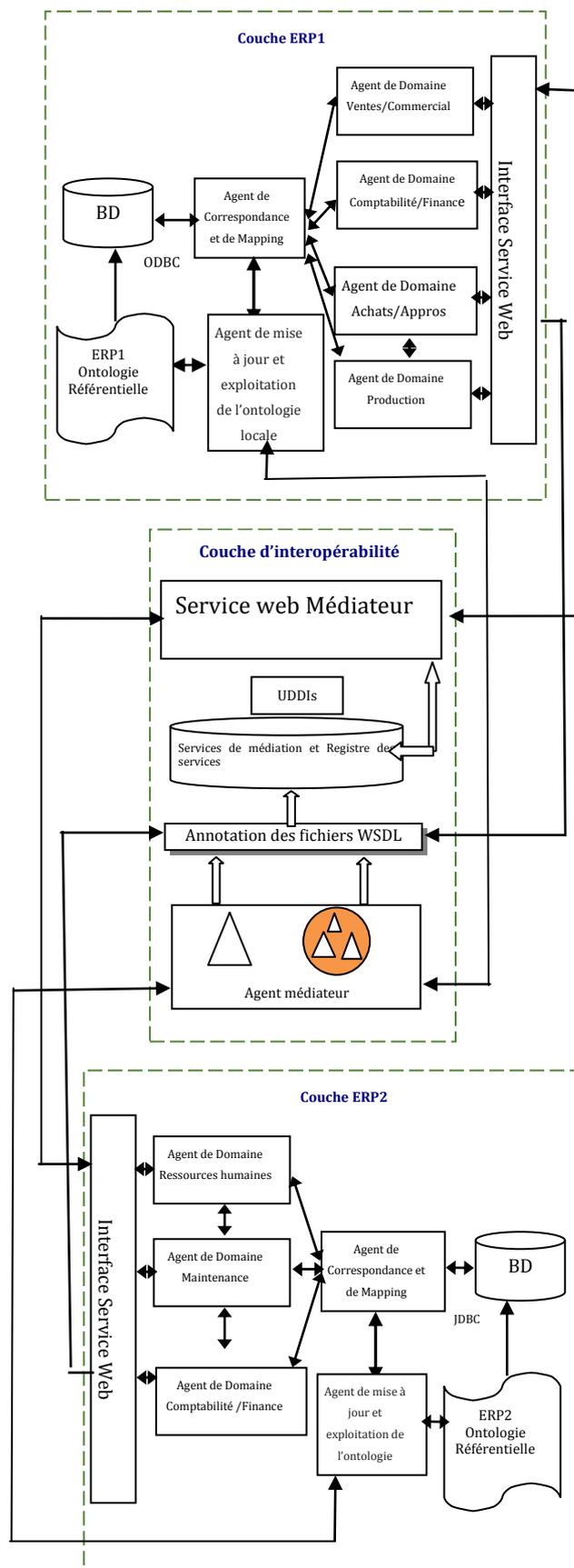


Figure 02: Architecture Générale

4.1 Composants de l'architecture

L'architecture générale proposée (Cf. Figure2) est en multi couches, dont chaque couche possède ses propres fonctionnalités avec des responsabilités précises. Elle a pour objet essentiel de résoudre le problème d'échange d'informations dans des organisations possédant des ERPs hétérogènes. Voici une description plus précise des différentes couches et de leurs relations.

4.1.1 Couche ERP

Est une couche de base, composée d'un ensemble de métiers et une base de données ERP unique et centrale, ayant une variété des données collectées à partir de n'importe quel domaine de l'ERP. Elle assure la communication et la collaboration entre les différents métiers tels que le métier de l'entreprise Achats/Appros, le métier Vente/Commercial, le métier Comptabilité/Finance...etc. Cette couche joue aussi un rôle fondamental pour la communication par échange d'information avec les autres couches des différents ERP via la couche d'interopérabilité à partir de la base de données de l'ERP. Cette couche comporte un ensemble de composants illustré comme suit :

- Base de données unique et commune.
- Ensemble d'agents situés : dans notre architecture comporte une liste d'agents cognitives. Ces agents se communiquent entre eux à travers l'envoi des messages.
- Ontologie propre à l'ERP : regroupe une collection de termes et de définitions du métier de l'entreprise. Elle regroupe un nombre de sections principales, à savoir, les activités et processus, l'organisation, la stratégie et le marketing. Elle est construite par fusion des ontologies de domaines fonctionnels de l'ERP.
- Une interface inhérente aux services web. L'intégration des Web services pour offrir tout ce qui est nécessaire et accessible à travers le Web.

4.1.2 Couche de l'interopérabilité

Cette couche modélise le processus d'interopérabilité entre les ERP, qui passe par un ensemble des étapes successives. Elle comporte un ensemble de composants, à savoir :

- Un agent de médiation qui englobe l'ontologie de domaines propres aux ERPs, et celle des aspects conflictuels.
- Un registre contient les services web. Il regroupe aussi l'ensemble de services de médiation publiés pour chaque aspect conflictuel. La création de service peut être intuitive pendant la publication de service web. Elle peut être basée sur les relations existantes entre les classes instanceable d'un aspect conflictuel dans l'OAC (Ontologie des

Aspects Conflictuels).

4.2 Fonctionnement globale de l'architecture proposée

Pour comprendre le fonctionnement de l'architecture, il est utile de présenter un cas d'école réel. Dans notre contexte, le Groupe ENSP est une société de services pétroliers et parapétroliers. Il regroupe plusieurs filiales et participations éloignées en partenariats avec des entreprises étrangères, à savoir HESP, BJSP, MESP, WESP et BASP. Chacune d'eux possède un ERP et un réseau informatique autonome. Afin de présenter un reporting fiable à la Direction Générale du groupe, il est indispensable de rendre ces progiciels interopérables entre eux et résoudre les différents conflits sémantiques tels que les concepts d'une écriture comptable dans deux ERPs.

Format de date : dd/mm/yyyy ou yyyy/mm/dd.

Nature : Investissement / Charge (montant <30000 DA).

Classe : 2/6.

Compte : 2xxxx/6zzzzzzz.

Montant : DA/Dollars.

TVA Incluse : True/False

Entreprise : ENSP/HESP

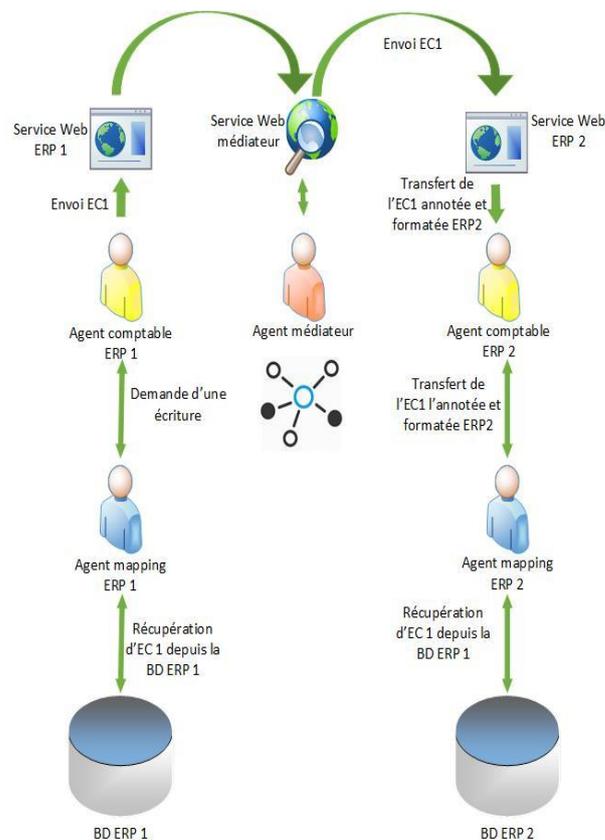


Figure 03 : Articulation de système d'information du Groupe ENSP (Scénario)

Dans la figure 3, l'agent comptable ERP 2 lance une requête via un service web vers un agent comptable 1, après la réception de la requête par l'agent cible. Ce dernier interroge la base de données de l'ERP1 pour récupérer les informations nécessaires via l'agent de mapping et de correspondance. Ensuite, il envoie ses informations vers l'agent de médiation qui lance le service web médiateur pour une médiation sémantique fiable. Puis, il achemine ses informations vers l'agent cible, qui à son tour procède à une vérification selon son domaine de compétences. En cas d'échec, il relance une autre requête. Dans l'autre cas, il transmet les informations à son agent de mapping et de correspondance pour mettre à jour la base de données.

L'interaction entre les agents sources et cibles est faite par le biais des services web, un service médiateur est utilisé pour résoudre les conflits entre ces services par les annotations écrites en langage WSDL proposées par l'agent de médiation. Celui-ci utilise une ontologie de domaine et une autre de gestion des conflits. L'approche SMA permet de lancer plusieurs agents au même temps en cas de plusieurs conflits.

Afin de résumer les rôles de chaque agent mentionné dans les paragraphes précédents, on a opté pour une présentation AUML (Cf. Figure 4) pour clarifier les différentes interactions entre les agents du système.

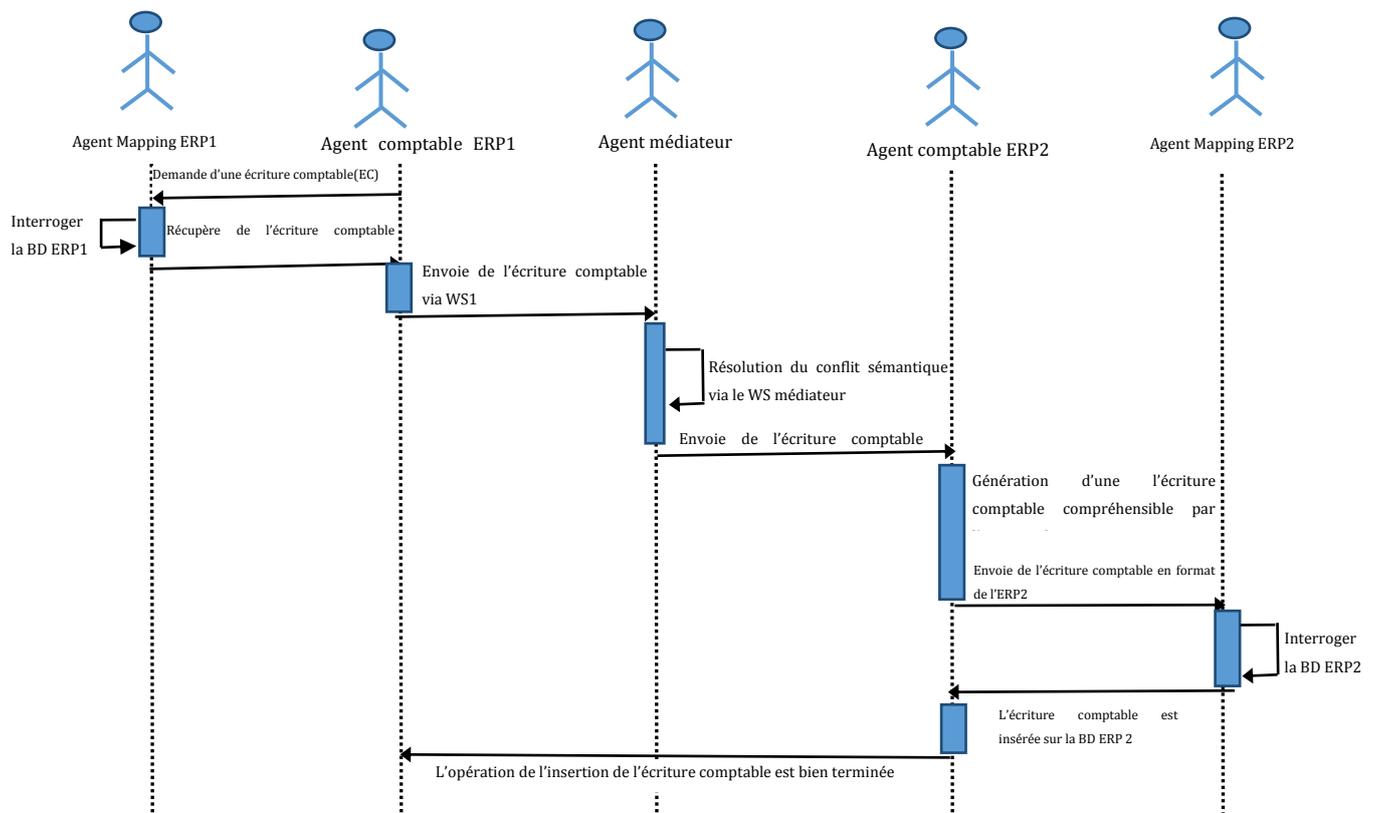


Figure 04 : Diagramme de séquence AUML

4.3 Processus de l'interopérabilité sémantique

Afin de décrire les différents rôles de chaque composant de l'architecture précédente,

il est utile de présenter la démarche préconisée et illustrée dans la figure 5.

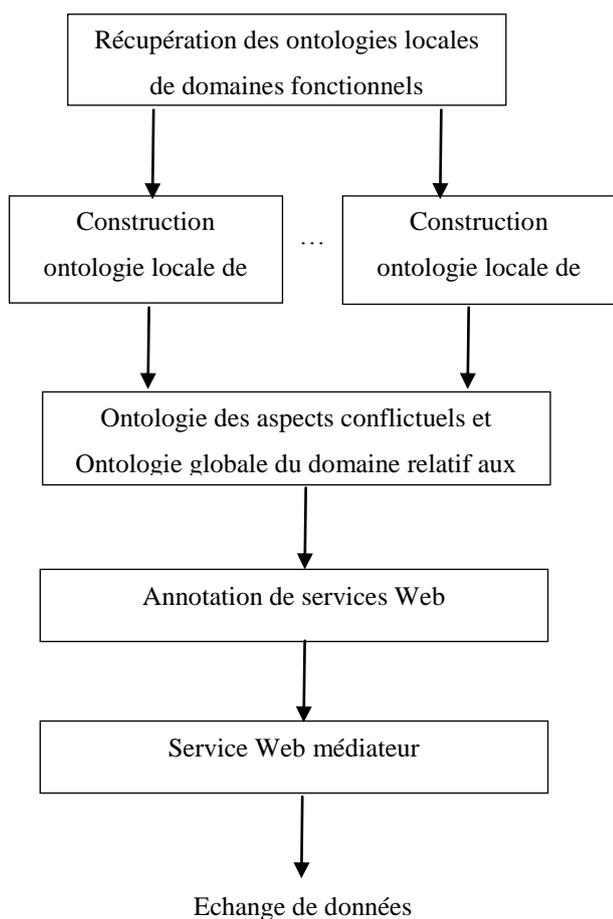


Figure 05 : Processus de l'interopérabilité sémantique

4.3.1 Récupération des ontologies locales de domaines fonctionnels

Dans cette étape, nous reprenons en premier lieu notre travail [6] où nous avons modélisé l'ERP par un ensemble d'agents. Avant de créer un agent intelligent, il faut tout d'abord créer sa base de connaissance ; qui contient des informations nécessaires pour que ce dernier puisse effectuer ses buts. Un agent intelligent est un intermédiaire entre l'ERP et l'environnement de médiation via un service WEB. Principalement, la BC d'un agent de domaine contient le contexte du module fonctionnel dédié. La BC contient l'ontologie locale qui décrit le domaine fonctionnel.

4.3.2 Construction de l'ontologie locale de l'ERP

Après avoir créé des ontologies de domaines fonctionnels, nous avons utilisé deux agents situés pour concrétiser cette phase.

Agent de mise à jour et exploitation de l'ontologie locale de l'ERP : Il exploite et met à jour l'ontologie locale de l'ERP en utilisant les ontologies locales de chaque domaine fonctionnel. Il gère tout accès aux ontologies des domaines fonctionnels. Ces dernières construisent par fusion

l'ontologie de l'ERP et celles des applications externes de l'ERP. Elle regroupe une collection de termes et de définitions du métier de l'entreprise. Il envoie les mises à jour vers l'ontologie globale de l'agent médiation.

Agent de correspondance et de mapping (Agent de routage) : Il permet de connecter entre les agents et la base de données ERP. Il est apte à établir une conversation avec les différents agents via l'agent de mise à jour et exploitation de l'ontologie locale de l'ERP. Il peut faire toute recherche des agents pertinents pour une coopération.

Exemple 1 : Pour illustrer l'architecture que nous avons proposé, nous commençons en premier lieu par la construction d'une partie des ontologies du module fonctionnel de la comptabilité relative à l'ERP1 et ERP2 respectivement figure 06(a) et figure 06(b).

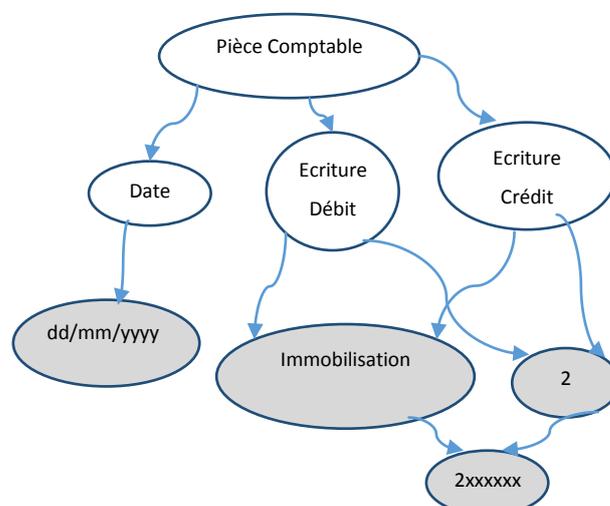


Figure 06(a) : Extrait de l'ontologie de l'ERP1 (les concepts à l'intérieur en zones grisées représentant le contexte du concept nature)

4.3.3 Construction de l'ontologie globale du domaine relatif aux ERPs et celle des aspects conflictuels

La construction d'une ontologie de médiation a pour objectif essentiel d'explicitier tous les concepts des organisations. Elle sera utilisée pour annoter les différents services Web et constitue la base sur laquelle les conflits sont résolus. Nous proposons de scinder l'espace conceptuel en deux niveaux, à savoir, l'ontologie de domaine (OD) et l'ontologie de contexte relatif aux aspects conflictuels (OAC).

Ontologie du Domaine : L'OD définit les concepts génériques et leurs relations couvrant les ERPs. Les concepts définis, à ce niveau, ne doivent pas faire l'objet de conflits entre les différents services web.

Ontologie des aspects conflictuels : Elle est créée pour enrichir la description de concepts génériques décrits au niveau de l'ontologie des ERPs. Cet enrichissement a pour but de mettre en évidence la connaissance nécessaire à la détection d'éventuels conflits sémantiques survenant lors de

toute interaction entre les services WEB. En effet, les OAC permettent de décrire les différents aspects conflictuels caractérisant le contexte de chaque paramètre de service Web.

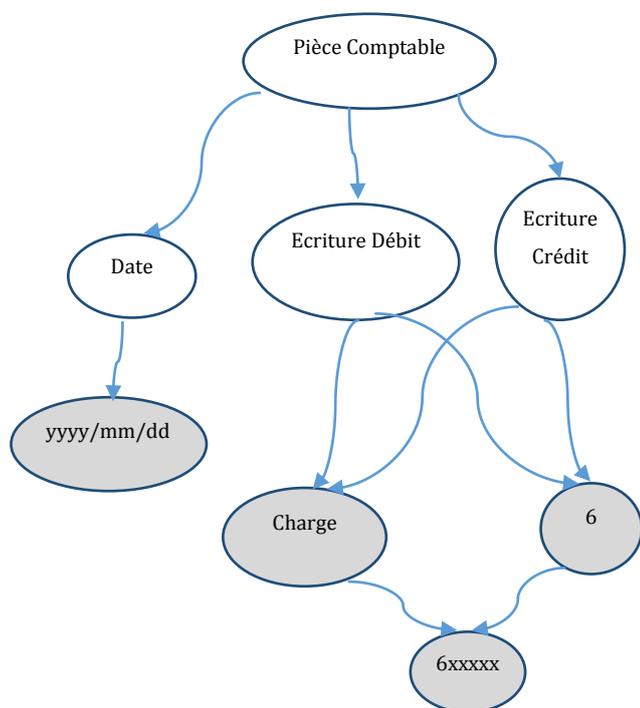


Figure 06(b) : Extrait de l'ontologie de l'ERP2 (les concepts à l'intérieur en zone grisées représentant le contexte du concept nature)

Définition 1 : Ontologie du Domaine est un sextuplet $\langle C;D;OP;DP; SC; SP \rangle$ où :

- C est un ensemble de classes ;
- D est un ensemble de types de données ;
- OP est un ensemble de propriétés d'objet. Chaque propriété objet a un ensemble d'origine (domaine) et un ensemble d'arrivée (range) dans C ; les classes sont reliées par des relations non taxonomiques.
- DP est un ensemble de propriétés de données. Chaque propriété de données a un ensemble d'origine (domain) dans C et un ensemble d'arrivée (range) dans D ;
- SC est une relation dans $C \times C$, représentant une relation de sous-classe ;
- SP est une relation dans $(OP \times OP) \cup (DP \times DP)$ représentant une relation de sous-propriété entre des propriétés homogènes.

Exemple 2 : Dans la figure 7, nous utilisons en premier lieu OWL comme modèle de représentation des ontologies. Puis, nous illustrons (Cf. Figure 8) une partie de l'ontologie commune qui constitue un accord concernant les noms de concepts sémantiques utilisés pour permettre

l'interopérabilité. Chaque ERP doit adhérer à cette ontologie du domaine. Elle contient aussi l'ontologie des aspects conflictuels. Elle permet de décrire le contexte de concepts de l'ontologie de domaine. Ceci permet de décrire clairement les hétérogénéités sémantiques. Après avoir adhéré à l'ontologie de domaine, l'expert métier doit mettre à jour les ontologies des aspects conflictuels associées à sa sémantique locale. Ceci permet d'établir les correspondances avec les représentations des autres fournisseurs. L'ontologie de conflits fournit les vocabulaires qui permettent de spécifier les différentes représentations structurelles et sémantiques des modificateurs.

Définition 2 : Ontologie des Aspects Conflictuels est un 3-tuplet $\langle ACg, ACi, \mathcal{E} \rangle$, où :

- "ACg" est un ensemble de classes représentant différents aspects conflictuels liés aux concepts de l'OD. Chaque classe "acg", appartenant à "ACg", a une superclasse et un ensemble de sous-classes. Chaque classe "acg" porte une désignation ou un nom représentant un aspect conflictuel ;

- "ACi" est un ensemble distinct de classes instanceables ayant une seule superclasse dans "ACg". Par définition, "aci" n'a pas de sous-classes.

- " \mathcal{E} " représente les relations entre concepts similaires ou de fratrie (sibling relationship) entre concepts "ACi" ou "ACg". Les relations entre les classes appartenant à "ACg" sont de type disjoint. Par contre, les relations entre les classes de "ACi", d'un "acg" donné, sont des relations entre pairs (Peer relationship) qui indiquent des conflits entre des données ayant la même sémantique.

```
<?xml version="1.0"?>
```

```
<rdf:RDF
xmlns="http://www.semanticweb.org/hp/ontologies/2017/9/
Ontologie_de_conflit_ERP#"
xml:base="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP"
```

```
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
```

```
xmlns:owl="http://www.w3.org/2002/07/owl#"
```

```
xmlns:xml="http://www.w3.org/XML/1998/namespace"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
```

```
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

```
<owl:Ontology
```

```
rdf:about="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP"/>
```

```
<owl:ObjectProperty
```

```
rdf:about="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Bénéficiaire">
```

```
<rdfs:domain
```

```
rdf:resource="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Pièce_Comptable"/>
```

```
<rdfs:range
```

```
rdf:resource="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Tiers"/>
```

</owl:ObjectProperty>

<owl:ObjectProperty
 rdf:about="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Concerne">

<rdfs:domain
 rdf:resource="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Ecriture">

<rdfs:range
 rdf:resource="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Classe">

<rdfs:range
 rdf:resource="http://www.semanticweb.org/hp/ontologies/2017/9/Ontologie_de_conflit_ERP#Nature">

</owl:ObjectProperty>

Figure 07 : Format OWL de l'ontologie de médiation

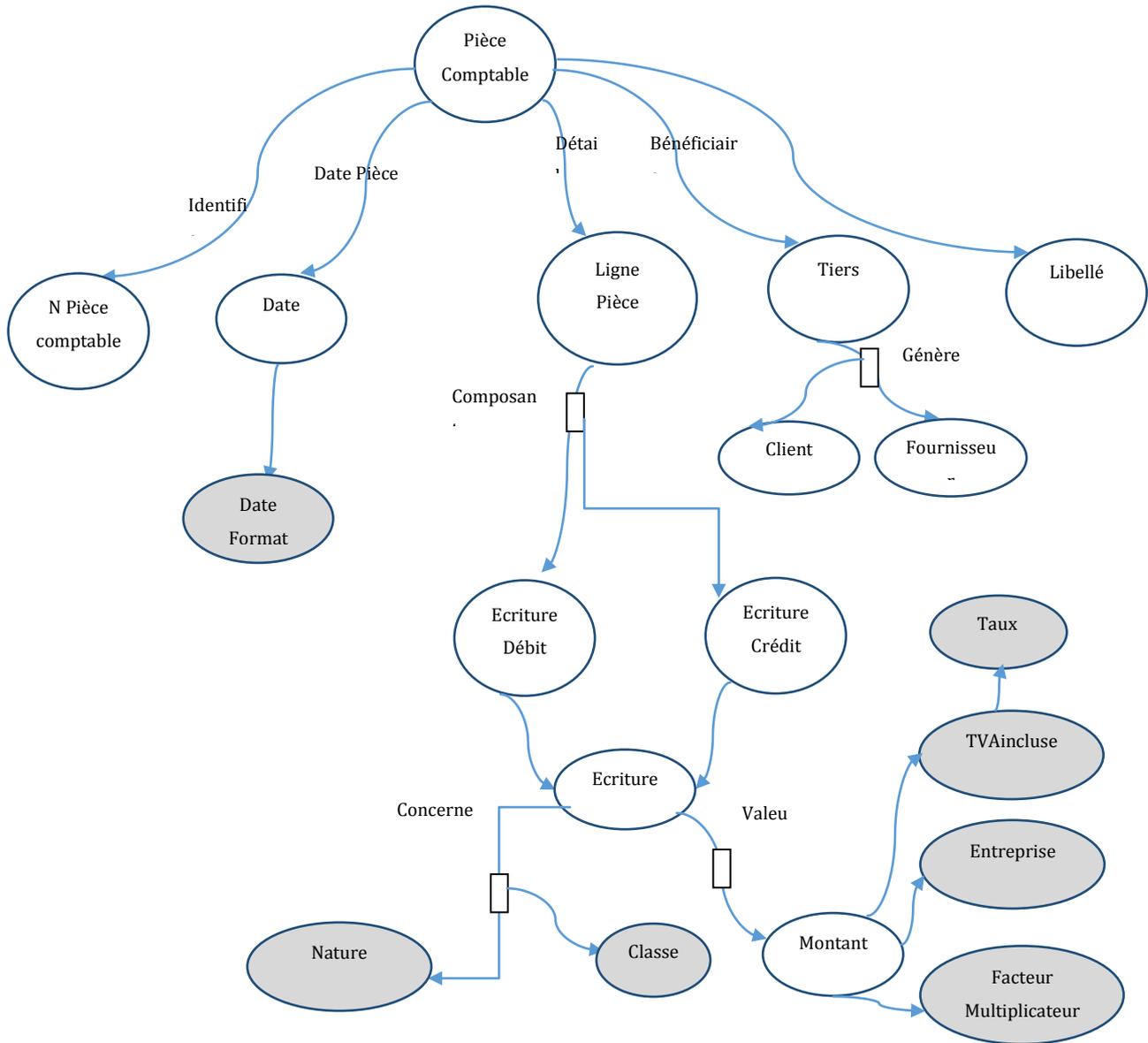


Figure 08 : Extrait de l'Ontologie de médiation

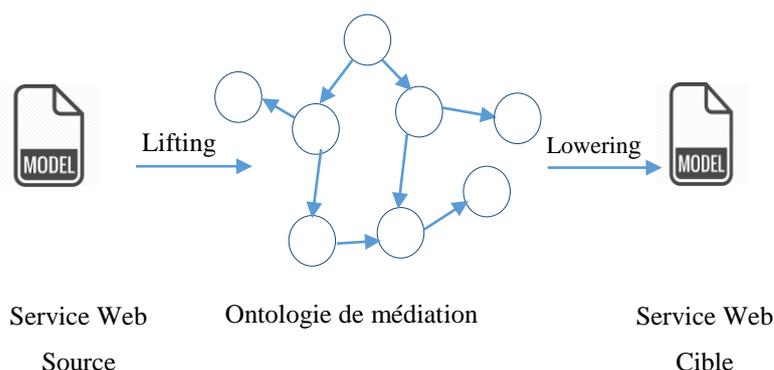


Figure 09 : Annotation des descriptions relatives aux services web de l'exemple.

Définition 3 : Le contexte "Ct" est un ensemble de couples dimension-valeur identifié de manière unique ayant la forme suivante : $\{(D_i, V_i) | i \in [1; m] \text{ où } D_i \in AC_g \text{ et } V_i \in AC_i\}$.

Définition 4 : La relation entre deux contextes "Ct" et "Ct'" où $Ct = \{(D_i, V_i)\}$ et $Ct' = \{(D_i, V_i')\}$, existe si et seulement si pour chaque $i \in \{1..n\}$ il y a $v_i R_{D_i} v_i'$ où $R_{D_i} \in \mathcal{E}_{AC_i}$.

4.3.4 Annotation des services Web

L'enrichissement des données échangées via l'information sémantique est nécessaire pour le bon déroulement du processus de médiation. Cela se fait à partir d'une annotation des descriptions WSDL des services. Le modèle d'annotation des services est exprimé en termes d'une OD. Néanmoins, une OD est censée conceptualiser les connaissances du domaine des ERPs. On note également la présence de trois attributs d'extensibilité qui sont définis par défaut : l'attribut <Montant > permet l'association entre un composant WSDL et un concept d'une ontologie, et les attributs <liftingSchemaMapping> et <loweringSchemaMapping> sont ajoutés aux définitions de

du schéma des données et l'information sémantique de l'ontologie (Cf. Figure 9).

Comme le montre la figure 8, des attributs SAWSDL sont ajoutés aux descriptions des services Web source et cible, il s'agit des schémas XML. Ces annotations font référence aux définitions de mapping des données XML de et vers les ontologies locales de l'ERP dans laquelle nous trouvons les modificateurs et leurs valeurs reliés au concept concerné Montant.

4.3.5 Service Web Médiateur

Notre architecture comprend plusieurs types de services. Les services Web broker, permettant de superviser et orchestrer les autres services d'interopérabilité qui sont les services Web : de publication, de découverte, de médiation et d'exécution. La médiation sémantique concerne les services sélectionnés, elle est prise en charge par un service Web de médiation, qui s'active une fois la requête suivante soit reçue, émanant du service Web broker :

Request (#Mediate, DescriptionWS1, DescriptionWS2) ;

Où : DescriptionWS_i désigne les URLs des descriptions WSDL. Il télécharge les fichiers WSDL des SW source et SW cible, et appelle les définitions de mapping relatives aux annotations liftingSchemaMapping pour exécuter les transformations sémantiques, des schémas via l'ontologie de médiation. Chaque service Web médiateur généré propose une opération appelée mediateX2Y, où X et Y sont remplacés par les noms des messages d'entrée et sortie des opérations. Ces dernières sont spécifiées dans les fichiers WSDL des services mis en relation via le service Web médiateur. Les entrées de l'opération de médiation sont les valeurs qui doivent être transformées dans la représentation requise, lesquelles sont spécifiées par les annotations des fichiers WSDL. Les sorties de l'opération sont les valeurs transformées, qui ont la signification désirée par l'opération cible.

Définition 5 : Le service de médiation a la forme suivante $MS_j(\$I_j ; ?O_j)$:

$G_I \rightarrow G_O$, où :

- "\$I_j" et "?O_j" sont respectivement un ensemble de variables d'entrée et de sortie de MS_j ;

```

XML Schéma du service Web source
...
<xs : element name = «Montant» type=double
  SawSDL :
  liftingSchemaMapping='http://MK/AOSS/
  mapping/Montant2OLD.xslt'>
  ...
  /xs :sequences>

XML Schéma du service Web cible
...
<xs : elementname = «Montant» type=double
  SawSDL :
  liftingSchemaMapping='http://MK/AOSS/
  mapping/Montant2OLD.xslt'>
  SawSDL
  :loweringSchemaMapping='http://MK/AOSS/
  mapping/OLD2Montant.xslt'>

```

types pour spécifier les correspondances entre les éléments

- G_I et G_O sont un ensemble de triplets RDF représentant les paramètres d'entrée et de sortie contextualisés.

4.4 MISE EN ŒUVRE SUR LA PLATE-FORME MULTI-AGENTS JADE

Dans notre proposition, nous avons utilisé le protocole d'ACL de la FIPA comme un langage de communication entre les agents. Le FIPA-ACL [23] sur KQML a des avantages sémantiques plus rigoureux ainsi qu'un important effort de standardisation. De plus, nos agents utilisent un protocole de coopération. Ce dernier a été établi de la part de framework JADE (Java Agent Development Framework) qui fournit un API pour développer un système multi-agents. En outre, ce framework a été sélectionné parce qu'il propose une bibliothèque riche de protocoles de coopérations. L'ajout de nouveaux modules fonctionnels implique la création de nouveaux agents sur le framework JADE. L'ajout des modules ou des applications externes en grande quantité concerne la propriété d'extensibilité. Le nombre important d'agents nécessite plus de communication qui va affecter la demande du système de transport des messages JADE. L'extensibilité des performances du système de transport des messages JADE sont traitées dans [24, 25]. Les résultats obtenus assurent que JADE a exploité l'importance de l'extensibilité selon différents scénarios.

5 CONCLUSION

Nous avons présenté dans le cadre de cet article une approche de médiation sémantique et dirigée par le contexte pour l'interopérabilité de progiciels de gestion intégrée. Principalement, notre contribution utilise un mécanisme permettant de fournir les informations sémantique nécessaire à l'effet de rendre explicite l'interprétation de données échangées entre les ERPs. Elle déclenche automatiquement un mécanisme de médiation, basé sur les services web, pour opérer la transformation requise. En ce sens, l'interprétation des données est explicitée à travers une ontologie à deux niveaux. En effet, au lieu d'une seule ontologie faisant office d'un schéma de médiation, on a proposé de scinder l'espace sémantique en une ontologie du domaine et une ontologie des aspects conflictuels. L'annotation de services est basée sur la notion de contexte. Le contexte, comme connaissance, précise l'interprétation des données que chaque service requiert ou retourne. Le contexte est exprimé en termes de l'ontologie des aspects conflictuels. Il précise les différents aspects conflictuels potentiels pouvant susciter une médiation lors des échanges des données. Notre approche utilise le service web médiateur comme règle de correspondance ou de transformation assurant la conversion des paramètres de services.

Nous avons l'intention de poursuivre ce travail d'abord, à travers une validation formelle de la solution, ensuite par la recherche des possibilités de l'étendre en définissant des algorithmes pour assurer le bon fonctionnement de notre architecture. Enfin, il peut être généralisé sur les autres métiers de l'ERP comme les RH pour prendre en charge la

totalité des conflits dans les SIs entreprises à travers l'orchestration sémantique des différents modules fonctionnels de l'ERP.

REFERENCES

- [1] Durgansh, S., Naveen, K., (2009). Interoperability Maneuver of the Enterprise's Dimensions with ERP system. Global Journal of Enterprise Information System, Vol. 01, N°01, pp. 42-52.
- [2] MARKESS International (2013). Attentes des entreprises pour les solutions de gestion intégrée ERP/PGI face aux nouveaux enjeux, Référentiel de pratiques (2011-2013), <http://www.markess.fr>.
- [3] Bonnet, P., (2005). Cadre de référence Architecture SOA. Meilleures Pratiques. Montreuil, 23 Février 2005. [http://pie.bonnet.ifrance.com/ON-guideSOA-2005-02-23%20\(part1\).pdf](http://pie.bonnet.ifrance.com/ON-guideSOA-2005-02-23%20(part1).pdf).
- [4] Monfort, V., Goudeau, S., (2004). Web services et interopérabilité des systèmes d'information. Dunod, Paris, 2004.
- [5] Bennharzallah, S., Kazar, O., Caplat, G., (2011). Intelligent query processing for semantic mediation of information systems, EIJ: Egyptian Informatics Journal, Vol. 12, pp. 151-163.
- [6] Mesbahi, N., Kazar, O., Zoubeidi, M. and Benharzallah, S., (2014). An agent-based modeling for an enterprise resource planning (ERP). the 6^{ème} conference on intelligent information and database systems, ACIIDS 2014 Bangkok, Thailand LNCS springer, vol. 551, pp. 225-234.
- [7] Carbonell, J., G., and Siekmann, J., (2006). Agent-Oriented Information Systems. 7th International Bi-Conference Workshop, AOIS 2005 Utrecht, ISBN-10 3-540-48291-1 Springer Berlin Heidelberg NewYork.
- [8] Purvis, M., Cranefield, S., Bush, G., Carter, D., McKinlay, B., Nowostawski, M., Ward, R., (2000). The NZDIS Project: an Agent-Based Distributed Information Systems Architecture.
- [9] Talens, G., Boulanger, D., and Séguran, M., (2007). Domain Ontologies Evolutions to Solve Semantic Conflicts. M. Collard (Ed.): ODBIS 2005/2006, LNCS 4623, pp. 51-67, 2007., Springer-Verlag Berlin Heidelberg 2007.
- [10] Kim, W. et al., (2008). Agent based intelligent search framework for product information using ontology mapping. J Intell Inf Syst (2008) 30:227-247 DOI 10.1007/s10844-006-0026-8, Springer Science + Business Media, LLC 2008.
- [11] Gal, A., Segev, A., (2005). Agent oriented data integration. In: Akoka J et al., editors. R Workshops 2005, LNCS 3770, Springer-Verlag, Berlin, Heidelberg, 2005. p. 98-108.
- [12] Bayardo, Jr., R., J., et al., (1997). InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments. Microelectronics and Computer Technology Corporation.
- [13] Jouanot, F., (2001). DILEMMA : vers une coopération de systèmes d'informations basée sur la médiation

- sémantique et la fusion d'objets, Thèse de Doctorat, Université de bourgogne, Novembre 2001.
- [14] Leclercq, E., Benslimane, D., Yétongnon, K., (1999). ISIS: A Semantic Mediation Model and an Agent Based Architecture for GIS Interoperability, in Proceedings of the International Database Engineering and Applications Symposium (IDEAS'99), Montreal, CANADA, August 2-4, 1999. Published by IEEE Computer Society Press, ISBN 0-7695-0265-2, pp. 81–92.
- [15] Haller, A., Cimpian, E., Mocan, A., Oren, E., and Bussler, C., (2005). Wsmx - a Semantic service-oriented architecture. In I. C. Society, editor, ICWS, pages 321–328. IEEE Computer Society, 2005.
- [16] Arroyo, S., and Stollberg, M., (2004). WSMO Primer. WSMO Deliverable D3.1, DERI Working Draft. Technical report, WSMO, 2004. <http://www.wsmo.org/2004/d3/d3.1/>.
- [17] Izza, S., (2006). Intégration des systèmes d'information industriels, une approche flexible basée sur les services sémantiques. Thèse de doctorat de l'Ecole Nationale Supérieure des Mines de Saint – Etienne, France, 20 Novembre 2006.
- [18] Martin, D., L., Paolucci, M., McIlraith, S., A., Burstein, M., H., McDermott, D., V., McGuinness, D., L., Parsia, B., Payne, T., R., Sabou, M., Solanki, M., Srinivasan, N., and Sycara, K., P., (2004). Bringing Semantics to Web Services: The OWL-S Approach. In J. Cardoso and A. P. Sheth, editors, SWSWPC, volume 3387 of Lecture Notes in Computer Science, pp. 26–42. Springer, 2004.
- [19] Mrissa, M., (2007). Médiation Sémantique Orientée Contexte pour la Composition de Services Web. Thèse de doctorat de l'université Claude Bernard Lyon I. 15 Novembre 2007.
- [20] Benharzallah, S., Kazar, O., (2010). A Scalable and Efficient Query Answering for a Context and Schema Mediation, JCIT: Journal of Convergence Information Technology, Vol. 5, No. 1, pp. 23-32.
- [21] Hausberger, T., (2009). Semantic Annotations for WSDL. Departement of Computer Science Leopold Franzens University Innsbruck. Semantic Web Services PS. 5 May 2009.
- [22] Gruber, R., T., (1993). A Translation Approach to Portable Ontology Specification. In Knowledge Acquisition, 5:199-220, 1993.
- [23] Online document: FIPA Communicative Act Library Specification, 2000.
- [24] Cortese, E., Quarta, F., Vitaglione, G., (2002). Scalability and Performance of Jade Message Transport System. Presented at AAMAS Workshop on AgentCities, Bologna, 16 th July, 2002.
- [25] Séguran, M., (2003). Résolution des conflits sémantiques dans les systèmes d'information coopératifs : proposition d'un modèle d'interaction entre agents, Université Jean Moulin, Lyon3, 2003.