

Université de Montréal

**L'ingénierie de documents d'affaires
dans le cadre du web sémantique**

par
Jamel Eddine Jridi

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Rapport pour la partie orale
de l'examen pré-doctoral

Juin, 2012

© Jamel Eddine Jridi, 2012.

Université de Montréal
Faculté des études supérieures

Cet examen pré-doctoral intitulé:

**L'ingénierie de documents d'affaires
dans le cadre du web sémantique**

présenté par:

Jamel Eddine Jridi

a été évalué par un jury composé des personnes suivantes:

Bruno Dufour,	président-rapporteur
Guy Lapalme,	directeur de recherche
Philippe Langlais,	membre du jury

Examen accepté le:

RÉSUMÉ

L'échange électronique de documents est le mécanisme de communication entre des applications distantes. Ce mécanisme décrit l'échange et les informations partagées dans un processus commercial. L'approche d'ingénierie de documents est une discipline qui aide à modéliser et structurer ces documents d'affaires en utilisant plusieurs normes et langages. Dans ce rapport, nous présentons les problèmes d'échange de documents et la nécessité d'avoir des méthodes qui aident dans l'ingénierie de documents afin d'améliorer le partage d'information à l'aide d'un modèle générique d'échange entre les différents partenaires d'un processus d'affaire. XML est une solution efficace pour structurer les documents sous forme de modèles. RosettaNet et ebXML sont des solutions d'ingénierie de documents, utilisant XML, pour résoudre certaines limites des standards d'échange traditionnels. Le problème majeur de XML est le fait qu'il ne traite pas la sémantique dans les documents, l'accès aux informations et l'interprétation des relations entre les documents et les concepts qui les composent. XML permet de structurer les documents échangés, mais sans en traiter la sémantique. Nous proposons qu'une ingénierie de documents basée sur les techniques du web sémantique est une solution efficace. Notre proposition est basée sur un support qui préserve la sémantique, qui relie les différents concepts dans les documents et qui dégage de nouvelles informations par l'application de plusieurs types de raisonnement.

Mots clés: ingénierie de documents, web sémantique, processus d'affaire, service web, partage d'informations, ebXML, RosettaNet.

ABSTRACT

The electronic exchange of documents is the mechanism of communication between remote applications which describes the exchange and the information shared in a business process. Document engineering is a discipline to model and structure these business documents by using several standards and languages. In this report, we present the document exchange problems and the need for methods for document engineering. Document engineering approach aims at an efficient information sharing. Through a generic exchange model between business partners. XML based is an effective solution for document structure. RosettaNet and ebXML were documents engineering solutions using XML language to deal with the limits of traditional exchange standards. Their major limitation dealing with the semantics in documents, information finding and interpreting their relations. It is used to structure exchanged documents, but does not deal with the semantics of data. In this report, we discuss the limits of XML and argue that. Semantic web based document engineering is an efficient solution which preserves the semantics, links different concepts between documents and extracts new information by means of several types of reasoning.

Keywords: document engineering, semantic web, web service, business process, information sharing, ebXML, RosettaNet.

TABLE DES MATIÈRES

RÉSUMÉ	iii
ABSTRACT	iv
TABLE DES MATIÈRES	v
LISTE DES FIGURES	vii
LISTE DES SIGLES	ix
CHAPITRE 1 : INTRODUCTION	1
1.1 Contexte général	1
1.2 Définition et domaine d'application	3
1.3 Les avantages de l'ingénierie de documents	4
CHAPITRE 2 : L'INGÉNIERIE DE DOCUMENTS	6
2.1 Introduction	6
2.2 Les phases de l'approche	6
2.2.1 L'analyse de contexte d'utilisation	8
2.2.2 L'extraction des patterns par l'analyse du processus d'affaires	9
2.2.3 L'analyse de documents et de ces composants	12
2.2.4 La conception des modèles de documents (Phase d'assemblage)	14
2.2.5 L'implémentation des modèles (Encodage en XML)	15
2.3 Conclusion	16
CHAPITRE 3 : XML VS. WEB SÉMANTIQUE	17
3.1 Introduction	17
3.2 XML et ces avantages	17
3.3 Web sémantique	21
3.4 La différence entre XML et Web Sémantique	29

3.5	Conclusion	30
CHAPITRE 4 : ÉTAT DE L'ART SUR LES STANDARDS D'AFFAIRE . .		31
4.1	Introduction	31
4.2	Approches basées sur XML	32
4.2.1	xCBL	33
4.2.2	RosettaNet	36
4.2.3	ebXML	39
4.3	Approches basées sur les techniques du web sémantique	43
4.4	Synthèse	47
4.5	Conclusion	49
CHAPITRE 5 : CONTRIBUTIONS		51
5.1	Introduction	51
5.2	Description de l'approche	51
5.3	Les phases de l'approche	55
5.4	Prototype	55
5.4.1	Description du PIP3A4	55
5.4.2	Conception et Implémentation	58
5.5	Conclusion	64
CHAPITRE 6 : CONCLUSION		66
BIBLIOGRAPHIE		68

LISTE DES FIGURES

1.1	L'ambiguïté entre les différentes parties d'un processus de vente des livres [8].	2
2.1	Les étapes d'un processus de vente (la réalisation de la commande, le paiement, le traitement de la commande et la livraison).	8
2.2	Les sous modèles de processus de vente.	11
2.3	Deux exemples de bons de commande envoyés de deux sites différents. Chaque site a sa façon de représentation des informations.	13
2.4	Diagramme de classes de processus de vente en ligne.	14
2.5	Un modèle de document implémenté avec XML Schema en lui associant une instance d'un document XML (en haut à droite).	15
3.1	Exemple d'un document XML.	19
3.2	Les trois sous langages de OWL 1.	26
4.1	Exemple d'un document XML d'une commande conformément à xCBL.	35
4.2	Un scénario d'achat entre deux entreprises dans le cadre de RosettaNet.	37
4.3	Un extrait du 3A4_MS_R02_00_PurchaseOrderRequest.dtd décrivant le RosettaNet PIP3A4 qui représente la demande d'un bon de commande.	38
4.4	L'approche ebXML- Automatisation des échanges interentreprises.	41
4.5	Représentation d'un exemple de la propriété owl:subClassOf dans le registre ebXML.	45
4.6	Architecture de RosettaNet Adapter selon [14].	46
5.1	Architecture de l'approche.	52
5.2	Modèle des concepts ebOWL.	53

5.3	Diagramme de séquence pour la réalisation de commande dans PIP3A4.	56
5.4	La définition de la structure de l'élément <code>SpecifiedFullPartner</code> dans le PIP3A4.	57
5.5	Description de (1) la taxonomie de notre ontologie (liste des concepts), (2) Les <code>Data Property</code> et (3) les <code>Object Property</code>	59
5.6	Un extrait du schéma XML décrivant l'élément <code>AccountDescription</code>	60
5.7	Une instance de la classe <code>OrderItem</code>	64

LISTE DES SIGLES

EDI	Electronic Data Interchange
XML	eXtensible Markup Language
UML	Unified Modeling Language
URI	Uniform Resource Identifier
RDF	Resource Document Format
DTD	Document Type Definition
OWL	Web Ontology Language
SGML	Standard Generalized Markup Language
xCBL	XML Common Business Library
ebXML	electronic business XML
ebRIM	electronic business Registry Information Model
ebRS	electronic business Registry Specification
BP	Business Process
CPA	Collaboration Protocol Agreement
CPP	Collaboration Protocol Profile
CDC	Core Data Component
OASIS	Organization for the Advancement of Structured Information Standards
UN/CEFACT	United Nations/Centre for Trade Facilitation and Electronic Business
TDCC	US Transport Data Coordinating Committee
RNIF	RosettaNet Implementation Framework
PIP	Partner Interface Process
MIME	Multipurpose Internet Mail Extensions

CHAPITRE 1

INTRODUCTION

Dans ce chapitre, nous présentons le contexte général de notre travail. Nous évoquons les problématiques au niveau de l'échange de documents et l'importance d'avoir des méthodes pour aider à la gestion de documents. Ensuite, nous définissons la discipline d'ingénierie de documents et ses domaines d'application. Enfin nous présentons les apports de cette nouvelle discipline.

1.1 Contexte général

De nos jours, il existe plusieurs types de documents ayant différentes structures et contextes. La circulation ou l'échange de documents est très important, ce qui permet de faciliter la communication entre les différentes parties d'un processus dans une entreprise, dans un organisme ou même sur internet. Dans le cadre d'une entreprise, nous pouvons avoir des documents de différentes natures, par exemple ; des relevés de transactions, des formulaires, des contrats, etc. L'apparition et l'évolution des entreprises virtuelles est remarquable sans avoir besoin des personnels ni d'espace physique. Une grande partie des affaires traitées sur le Web aujourd'hui a lieu à travers des échanges d'information rendus possibles par l'utilisation de documents comme interface. Cette évolution est due à l'évolution des techniques offertes par les services web. Mais l'échange de ces documents peut causer des problèmes liés à la compréhension du sens des mots dans les documents échangés. Le sens d'un mot pourrait changer selon les situations ou contextes. Par exemple, le mot *adresse* peut désigner l'adresse du vendeur dans un document et l'adresse de l'acheteur dans un autre. La Figure 1.1 montre l'ambiguïté au niveau des champs, dans les documents échangés entre des partenaires du processus de vente de livres. Nous remarquons que dans les différents services qui gèrent la vente en ligne sur un site fictif *GMBooks.com* : le champ référence de la commande est décrit par des termes ayant des sémantiques différentes :

- Order Reference signifie le numéro de la commande.
- Customer Reference désigne l'identifiant du client sur le site (p.e. nom d'utilisateur, etc), par contre la valeur de ce champ fait référence au numéro de la commande, et
- Details peut contenir une description textuelle de la commande mais pas forcément son numéro.

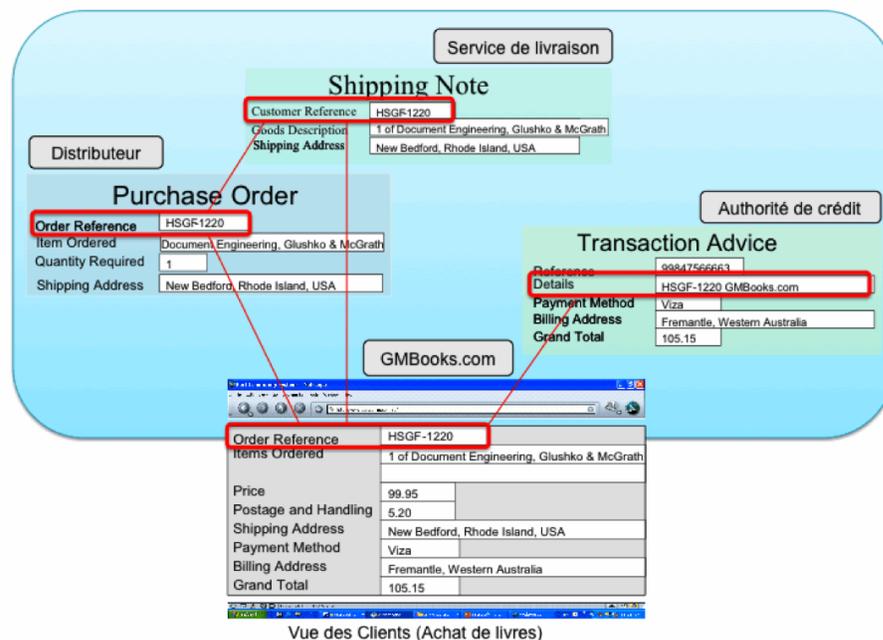


Figure 1.1 – L'ambiguïté entre les différentes parties d'un processus de vente des livres [8].

Pour y remédier, il faut que chaque partenaire utilise les mêmes définitions des informations dans leurs documents, ce qui est difficile. Il est plus pratique que chacun utilise un système permettant de transformer la représentation de l'information reçue dans une autre représentation compréhensible par l'entreprise. Donc, il faut avoir une façon plus précise et adéquate au moment de la définition de contenu des documents. Le but est de déterminer comment l'information et les données sont définies et utilisées, afin de concevoir un modèle décrivant la structure et la sémantique des concepts utilisés. Parmi

les solutions proposées, on retrouve : un dictionnaire de données, une base de données, un diagramme de classe UML, ou un schéma de documents en XML, etc.

Toutes ces recherches font partie du domaine de l'ingénierie des documents, nécessaire pour analyser, concevoir et mettre en œuvre les échanges d'information sur Internet. Elle est basée sur les méthodes d'analyse et de conception de données permettant d'avoir un modèle précis décrivant les informations.

XML et les services Web fournissent une grande flexibilité pour créer, manipuler, modéliser, organiser et partager les informations. Ces techniques ne nous donnent toutefois pas un point de vue sur la sémantique des documents. Il nous faut donc penser aux documents d'une façon plus abstraite et essayer d'utiliser d'autres techniques telles que le web sémantique.

Dans cette étude, nous nous intéressons à l'ingénierie des documents dans le cadre du web sémantique. Ces documents sont de diverses natures et présentés différemment. Nous concevons donc une méthode permettant de gérer ces échanges de documents, en utilisant les techniques du web sémantique.

1.2 Définition et domaine d'application

L'ingénierie de documents est une nouvelle discipline pour la spécification, la conception et la mise en œuvre des documents échangés électroniquement. Cette discipline est appliquée dans plusieurs domaines afin de faciliter la communication et gérer la grande masse d'informations qui circulent dans/et en dehors d'un organisme. Elle est utilisée dans plusieurs secteurs dont celui de la gestion des affaires ou du domaine médical, etc. Dans le domaine des affaires, le but est d'une part, de gérer le processus des échanges et d'autre part, d'identifier et de classer les types de documents, p.e. les bons de livraison, les contrats, les rapports des employés, les ventes, etc. Par contre dans le domaine médical, les documents peuvent être des rapports de médecins, des ordonnances médicales, des résultats d'analyses, etc.

Pour ce faire, l'ingénierie de documents se base sur des méthodes d'analyse et de conception permettant de construire des modèles formels pour décrire l'information se-

lon un domaine d'application particulier. Les méthodes d'analyse de documents sont effectuées dans le but de trouver un modèle abstrait et logique à partir des instances existantes d'un seul type de document, et ensuite d'encoder le modèle en utilisant des techniques comme XML. Un meilleur modèle, décrivant une classe de documents, est celui qui répond le mieux aux exigences des utilisateurs actuels et potentiels pour la réalisation de tâches spécifiques avec de nouvelles instances.

Selon Robert J. Glushko et Tim Mc Grath, l'utilisation des patterns réutilisables pour la gestion de processus des affaires rend la sémantique plus précise. Ceci n'est pas le cas dans le cadre d'échange des documents entre différentes entreprises, puisque chaque entreprise peut avoir un modèle personnel pour gérer ces documents. Par contre pour remédier à ce problème, il faut avoir un modèle d'échange de documents génériques qui gère la communication entre les entreprises afin de conserver la sémantique des documents, ce qui est très difficile.

1.3 Les avantages de l'ingénierie de documents

Dans les organisations, une grande variété de documents circulent à l'intérieur ou à l'extérieur p.e. les feuilles de temps, les bilans de dépenses, les rapports, les bons de commande, les catalogues, et d'autres types de documents. Ces documents sont de différentes natures : narratives comme les manuels d'instructions, ou transactionnelles comme les formulaires ou les relevés de transactions.

Le processus des affaires est géré par l'échange des documents qui sont utilisés comme des interfaces de communication. Il est possible d'avoir des chevauchements dans la sémantique des composants formant ces documents. Ce qui provoque une perte de temps et des dépenses afin de comprendre ces informations.

L'ingénierie de documents est une solution efficace. Elle trouve une façon pour modéliser, décrire l'information, et résoudre les problèmes liés à la sémantique. Dans un processus d'affaire, il y a au moins deux côtés à chaque échange de documents, et toutes les parties doivent s'assurer qu'ils comprennent les documents de la même manière.

L'ingénierie de documents construit un modèle de processus des affaires et de do-

cuments échangés. Cette approche est essentielle parce que les documents narratifs et transactionnels sont souvent étroitement liés, par exemple, la relation entre les formulaires d'immigration et les instructions pour les remplir, ou entre les brochures et les bons de commande.

CHAPITRE 2

L'INGÉNIERIE DE DOCUMENTS

2.1 Introduction

L'ingénierie de documents est une nouvelle discipline pour la spécification, la conception et la mise en œuvre des documents échangés électroniquement [8]. Cette discipline se base sur différents types d'analyse formant le processus d'ingénierie de documents. Dans ce chapitre, nous décrivons ces types d'analyses et les phases de l'ingénierie de documents. Nous utilisons un exemple fictif mais réaliste pour décrire le déroulement de l'approche. Cette partie est inspirée de [8].

2.2 Les phases de l'approche

L'ingénierie de documents s'appuie sur des méthodes d'analyse et de conception permettant de construire des modèles formels pour décrire l'information selon un domaine d'application particulier. Ces types d'analyse sont :

- L'analyse du processus d'affaires décrit le contexte général du processus, pour comprendre la sémantique des informations échangées ainsi que leurs relations.
- L'analyse des tâches identifie les différentes étapes et les informations réalisant une tâche donnée.
- L'analyse des documents commence par l'analyse des instances de documents. Cette technique extrait l'information structurelle, la représentation des données, le contenu de document, etc.
- L'analyse des données traite de façon abstraite les composantes des données, révélées par l'analyse de documents, et présente le domaine d'un point de vue conceptuel.

Ces méthodes d'analyse de documents sont effectuées dans le but de trouver un modèle abstrait et logique à partir des instances existantes d'un seul type de document, et

ensuite d'encoder le modèle en utilisant des techniques comme XML. Un meilleur modèle, décrivant une classe de documents, est celui qui répond le mieux aux exigences des utilisateurs actuels et potentiels pour la réalisation de tâches spécifiques avec de nouvelles instances.

Le processus d'ingénierie de documents est composé de plusieurs phases :

1. L'analyse de contexte d'utilisation.
2. L'extraction des patterns par l'analyse du processus d'affaires.
3. L'analyse de documents et de ces composants.
4. La conception des modèles de documents (Phase d'assemblage).
5. L'implémentation des modèles de processus et de documents (encodage en XML).

Pour illustrer le déroulement du processus, nous utilisons un scénario d'une vente en ligne illustré dans la Figure 2.1. Une entreprise envoie ses promotions à différents sites web qui gèrent la vente de ses produits en ligne. Ces sites reçoivent plusieurs types de produits, de marques différentes. Ces sites web sont utilisés comme une vitrine pour afficher les promotions reçues. Cette stratégie est utilisée pour améliorer la visibilité des produits et augmenter le chiffre d'affaire. Le gestionnaire des sites web gagne un pourcentage des ventes p.e. l'entreprise donne au site un produit à la valeur de 20\$ et ce dernier le revend à 25\$. Les clients commandent les produits à partir de ces sites web. Le gestionnaire du site traite la commande, vérifie les fonds du client et prend une pré-autorisation sur le montant de la commande. Ensuite, il envoie un bon de commande à l'entreprise qui prend en charge la préparation et la livraison des produits. Au moment de la livraison, cette entreprise informe le gestionnaire du site des détails de la livraison et lui autorise le retrait d'argent. Plusieurs organismes participent au processus de vente des produits matérialisé par un échange de documents entre les gestionnaires des sites et l'entreprise : bons de commande, des relevés de transactions, etc. Le problème majeur est lié à la différence de structure, de contenu et de sémantique des éléments dans les documents échangés, d'où l'importance d'une stratégie de gestion de ces documents. Dans le reste de ce chapitre, nous illustrons le processus de gestion de documents avec cet exemple.

2.2.1 L'analyse de contexte d'utilisation

Il faut comprendre les activités principales du processus d'affaire ainsi que les membres participant à ces activités, analyser le contexte d'utilisation et identifier les exigences [8]. L'identification des exigences dans les phases initiales d'un projet est très importante. Elle réduit la probabilité de refaire l'analyse, la conception et la mise en œuvre des étapes. Les exigences pour les documents et les processus d'affaires doivent être précises et vérifiables pour être utiles. Une façon d'y arriver est de les exprimer comme des règles (ou contraintes). Ces exigences seront exprimées avec des règles sur le contenu, la structure et la présentation des documents et de leurs composants. Ces règles formalisent les définitions du contexte dans lequel les documents sont utilisés, et les patterns décrivant les activités principales dans un processus d'affaire. Un pattern est un modèle, abstrait générique et réutilisable, qui peut être adapté à un ensemble d'instances ou contextes. La Figure 2.1 décrit le contexte général d'un processus de vente en ligne, dans lequel, nous avons différents participants : le client, le détaillant (le site web), le livreur, le distributeur (l'entreprise), etc.

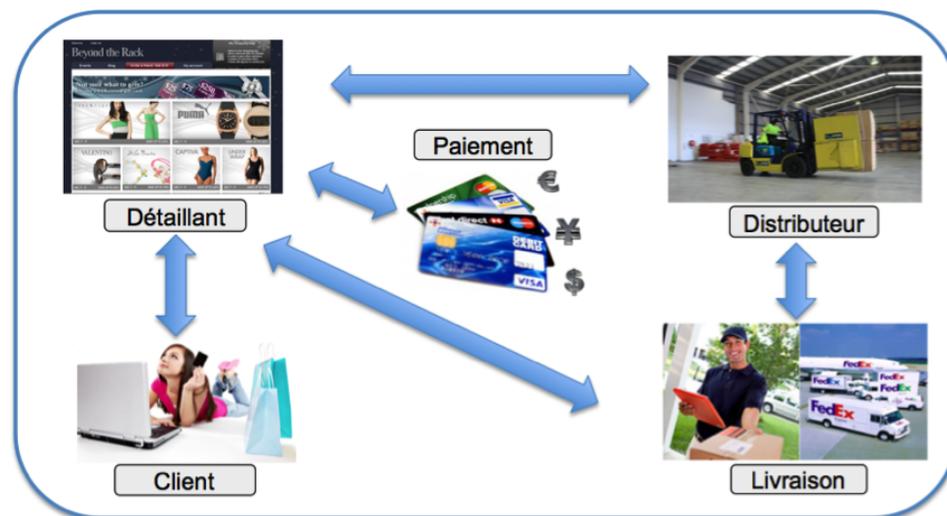


Figure 2.1 – Exemple des sites web qui gèrent la vente et les promotions de produits. Ainsi que la communication entre les différentes parties de ce processus.

2.2.2 L'extraction des patterns par l'analyse du processus d'affaires

Cette étape modélise les processus d'affaire dans une entreprise, à plusieurs niveaux d'abstraction et de granularité. Un processus d'affaire est une chaîne d'activités ou d'événements à réaliser dans un ordre chronologique. Cette chaîne peut produire des résultats qui peuvent être l'entrée d'un autre processus d'affaire. Il existe deux types de processus d'affaire : les processus internes (ou privés) réalisés entièrement dans une même entreprise, et les processus externes (de collaboration, ou d'affaires publiques) sont effectués entre deux ou plusieurs parties d'affaire. La compréhension d'un processus se base sur un ensemble de questions touchant le fonctionnement d'un processus, son but, etc. Par souci de cohérence, nous répondons aux mêmes questions pour chaque processus. Le Tableau 2.I montre les questions à poser ainsi que les réponses possibles dans le cadre de la vente en ligne. Ces questions et leurs réponses forment une base d'information sur les processus interconnectés à partir de laquelle nous pouvons développer des modèles.

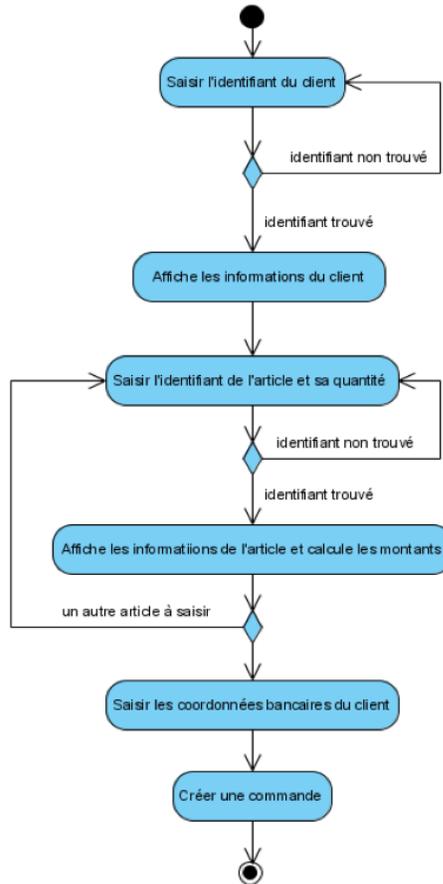
Ces données nous aident à construire le modèle du processus d'affaire pour la vente en ligne. Ce processus est composé de deux parties : la première partie concerne la réalisation de la commande, par contre la deuxième est destinée à la livraison du produit. Il est possible d'avoir un modèle pour la réalisation de la commande, et un autre pour la livraison. Ces modèles peuvent être décrits par les diagrammes UML, comme le diagramme de séquence pour montrer le scénario d'exécution d'une transaction particulière. Dans la Figure 2.2, nous les décrivons par deux types de diagrammes UML : diagramme de collaboration (Figure 2.2a) et diagramme de séquence (Figure 2.2b). Le diagramme de collaboration décrit la réalisation d'une commande. Le client se connecte au site en insérant son identifiant. Le système vérifie les informations saisies et affiche la page personnelle du client. Ensuite, le client choisit les articles et effectue son paiement en saisissant ses coordonnées bancaires. Enfin, le système prend une autorisation des fonds du client sur le prix de la commande et la crée. Le diagramme de séquence montre le scénario de la livraison d'une commande. Ce scénario commence par l'envoi d'un bon de commande du gestionnaire du site au fournisseur de l'entreprise. Ce dernier prépare

la commande et la transfère au service de livraison, lui de son côté valide la sortie de la marchandise. Le fournisseur informe le gestionnaire de la sortie de la commande et lui autorise le retrait de l'argent.

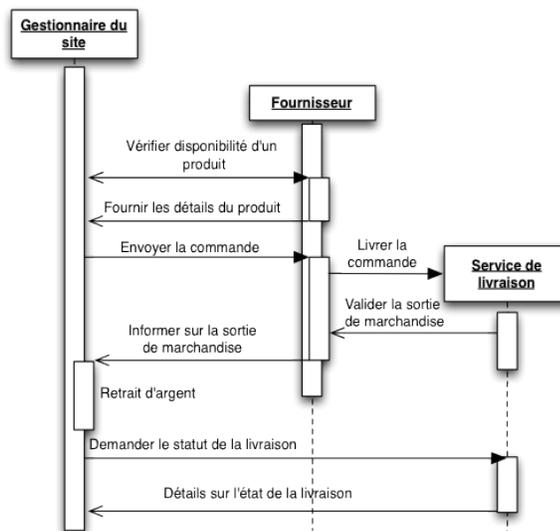
Tableau 2.I – Description de processus de vente en ligne.

Questions	Réponses
Quel est le nom du processus ?	Le processus de vente en ligne.
Quel est l'objectif de ce processus ?	<p>L'objectif est de gérer la vente de plusieurs types de produits. Ce processus est composé de plusieurs sous étapes :</p> <ul style="list-style-type: none"> ● Le choix des produits et paiements. ● Traitement de la commande. ● La livraison de la commande.
Quelles sont les industries, les organisations impliquées dans le processus ?	<ul style="list-style-type: none"> ● Les services de cartes de crédit. ● Le fournisseur. ● Les sociétés de livraison.
Quelles sont les intervenants ou les participants au processus ?	<ul style="list-style-type: none"> ● Le client comme acteur principal. ● Le gestionnaire du site. ● Le distributeur (le fournisseur de l'entreprise) ● Le livreur (FedEx, UPS, etc.)

L'échange ou la réalisation de ces processus se fait en utilisant des documents qui sont utilisés comme des interfaces ou des conteneurs d'information [8]. L'analyse de ces processus permet une connaissance commune et partagée de leur fonctionnement entre les différents partenaires d'affaire. Le plus important est d'avoir un modèle unique qui gère l'échange de documents entre les partenaires, c.-à-d. il est possible d'avoir des changements au niveau de processus interne mais c'est plus difficile pour ceux qui sont externes.



(a) Diagramme de collaboration pour la réalisation d'une commande



(b) Diagramme de séquence pour la livraison d'une commande

Figure 2.2 – Les sous modèles de processus de vente.

2.2.3 L'analyse de documents et de ces composants

Les documents sont les parties les plus visibles d'un processus d'affaire, nous devons donc les identifier et les comprendre. Nous voulons étudier les rôles joués par les documents utilisés comme des interfaces et des supports de communication entre une ou plusieurs parties. L'analyse des documents se base sur plusieurs critères : le contenu, la structure, la présentation, la syntaxe et la sémantique des informations.

Cette phase identifie les composantes sémantiques ou concepts contenus dans les documents sélectionnés, afin de créer une vue conceptuelle cohérente nommée `document component model`. Cette analyse est faite en utilisant des techniques d'analyse de données qui normalisent les composantes dans des structures en fonction de leur dépendance fonctionnelle (sémantique commune). Dans l'exemple du processus de vente en ligne, plusieurs documents sont échangés, dont différents types de relevé de transaction d'achat, des bons de livraison et des bons de commande. Dans notre exemple du processus de vente, l'entreprise reçoit des bons de commande de différents sites web qui gèrent les promotions et la vente de ses produits. La Figure 2.3 montre deux exemples de bons de commande envoyés depuis les organismes des sites à l'entreprise. Nous remarquons l'existence de composants d'information ayant une sémantique commune même si les documents viennent de différentes sources. Mais il existe aussi un chevauchement dans la sémantique de certains composants d'information existants, citons par exemple :

- `Reference` dans le premier document représente le numéro de transaction d'achat, mais celui de la commande dans le deuxième document.
- `Total` dans le premier document représente le montant total de la transaction après l'ajout des taxes, par contre dans le deuxième document, il signifie le total avant l'ajout des taxes.

Les bonnes règles de nommage sont importantes dans notre modélisation. Des noms significatifs aident non seulement à trouver une sémantique commune entre les composants, mais aident également dans l'analyse et la conception de composants de documents réutilisables. C'est le cas des champs `Shipping et Postage and Handling`

qui signifient à la fois le montant de livraison à payer. Aussi les champs Order date et Order Placed signifient la date de la réalisation de la commande.

BON DE COMMANDE			
PROMO. Inc. Adress: Montréal, QC			
Order no 445882	Order date Dec 10, 2010	Sub-total \$198.00	Taxes \$25.75 Shipping \$11.95 Credits \$10.00 Total \$225.70
	Deisha Sunglasses in Gray Size: One Size Price: \$99.00 SKU: GIOGARMANI560SQMB		Deena Sunglasses in Silver and Blue Size: One Size Price: \$99.00 SKU: GIOGARMANI599SVHV
		Name: Alain Adress: 2222, Vigi Avenue, Montréal, QC Paid via : Master Card Reference: 0102986473622	
Reference: 445882 Order Placed: Dec 10, 2010			
Item	Item Code	Description	Price
	GIOGARMANI560SQMB	Deisha Sunglasses in Gray Quantity: 1	\$99
	GIOGARMANI599SVHV	Deena Sunglasses in Silver and Blue Quantity: 1	\$99
Total:		\$198.00	Client: Alain
Postage and Handling:		\$11.95	Shipping Adress:
Taxes:		\$25.75	2222, Vigi Avenue, Montréal, QC
Order Payment:		\$225.70	
Payment Method:		Master Card	
Transaction Number:		0102986473622	

Figure 2.3 – Deux exemples de bons de commande envoyés de deux sites différents. Chaque site a sa façon de représentation des informations.

L'analyse de l'information dans l'ingénierie des documents crée un modèle conceptuel généralisé exprimant les règles de gestion des affaires pour tous types de documents requis dans le contexte d'utilisation. Nous appelons cet artefact un modèle de composants de documents définissant tous les éléments nécessaires pour optimiser la réutilisation et minimiser la redondance lors de la conception des nouveaux modèles de documents.

2.2.4 La conception des modèles de documents (Phase d'assemblage)

Nous passons ensuite de l'analyse à la conception. Nous créons des modèles pour les nouveaux types de documents basés sur les composantes, les structures et associations (par exemple, cardinalités en UML) satisfaisant aux exigences de notre contexte d'utilisation (voir la Figure 2.4). Nous avons plusieurs composants participant à la réalisation de notre processus de vente en ligne. La figure ci-dessous résume les informations ainsi que les relations entre les composantes du processus.

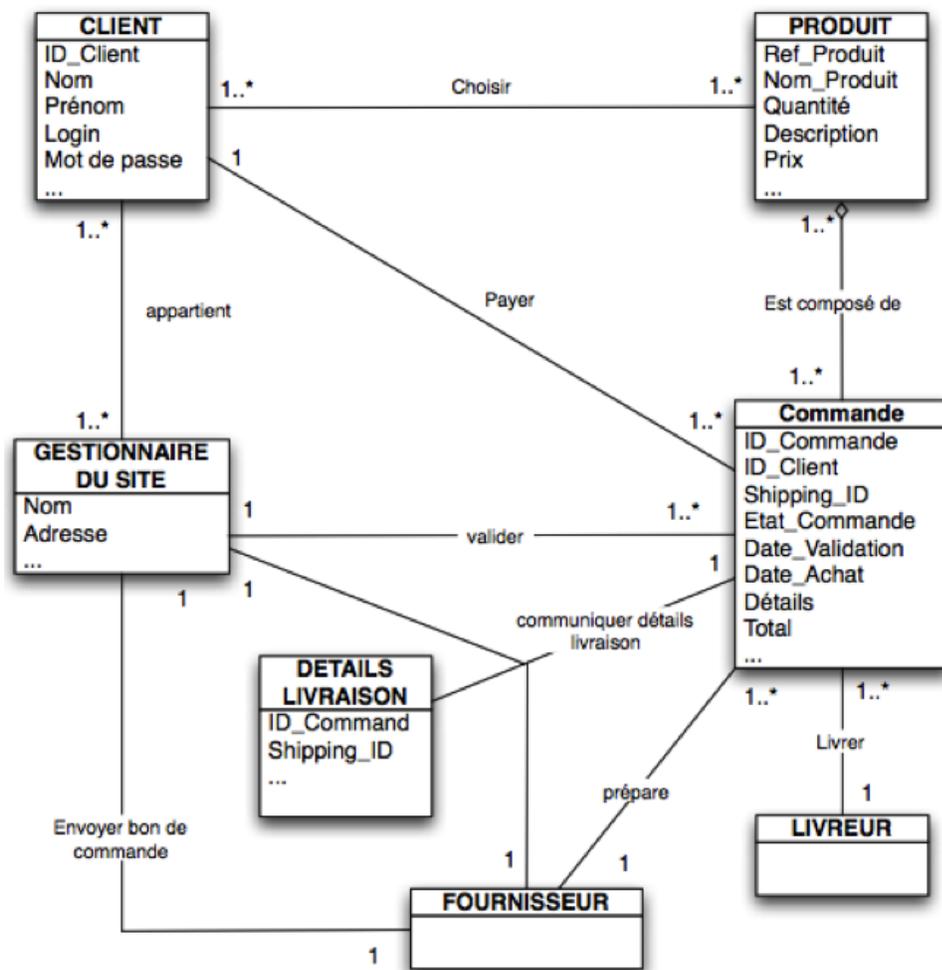


Figure 2.4 – Diagramme de classes de processus de vente en ligne.

Ces nouveaux modèles s'appellent modèles d'assemblage de documents. L'assemblage de documents est basé sur les modèles de composantes d'information. Une concep-

tion efficace des modèles d'assemblage de documents nous oblige aussi à savoir quand les modèles peuvent être réutilisés, quand un nouveau modèle doit être créé, et ce qui distingue un modèle d'un autre.

2.2.5 L'implémentation des modèles (Encodage en XML)

Cette étape transforme le modèle conceptuel, qui met en évidence les relations sémantiques entre les classes (p.e. diagramme de classes), en un modèle physique qui décrit l'ensemble de ces classes (p.e. une base de données relationnelle, un XML Schema, etc).

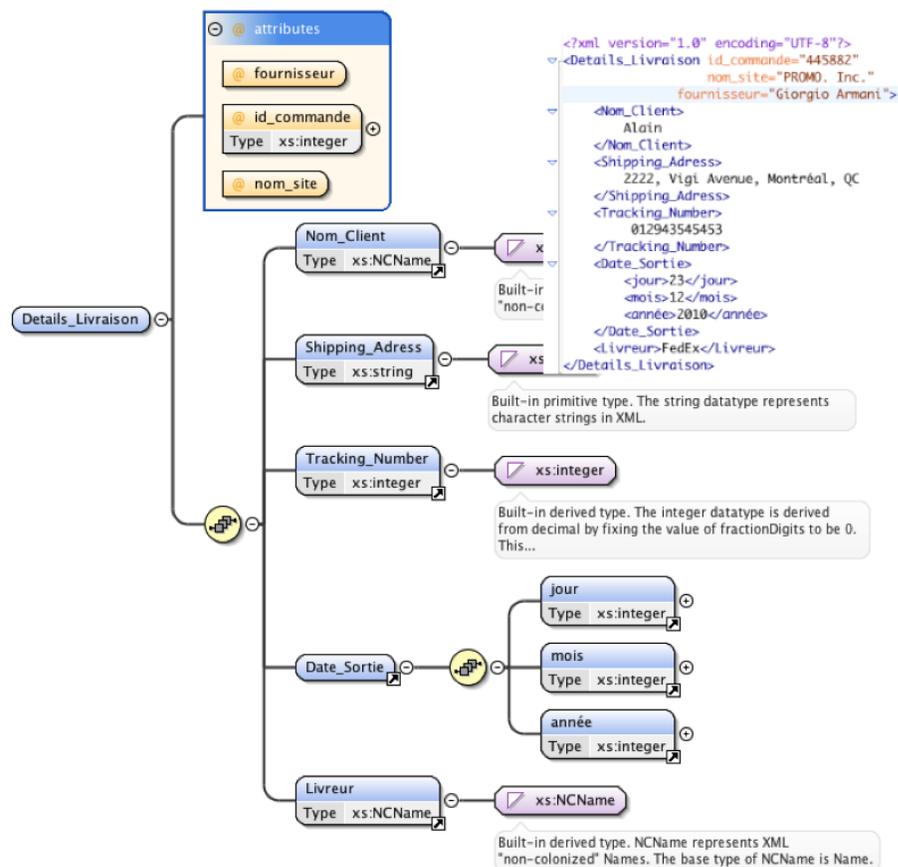


Figure 2.5 – Un modèle de document implémenté avec XML Schema en lui associant une instance d'un document XML (en haut à droite). C'est un exemple de documents échangés entre le fournisseur et le gestionnaire du site qui contient les détails de livraison de la commande.

Le modèle d'implémentation de documents est réalisé dans un langage de balisage comme XML Schema. Le choix est fait grâce aux avantages de XML Schema : la simplicité, la puissance, l'expressivité, et la maintenabilité [8]. La Figure 2.5 décrit un modèle d'un document échangé entre le fournisseur et le gestionnaire du site. Ce document informe le gestionnaire du site de la sortie de la commande ainsi que les détails de la livraison.

Pour les modèles physiques des processus d'affaires, la réalisation est faite en adoptant un méta-modèle approprié pour coder les règles et les exigences du contexte d'utilisation. Cette adaptation permet de créer le modèle de mise en œuvre des processus d'affaires. Parmi les exemples de modèles de processus d'affaire, nous trouvons RosettaNet PIPs [17] encodée en XML.

2.3 Conclusion

Dans ce chapitre, nous avons présenté les différentes phases du processus d'ingénierie de documents. Pour chacune, nous avons donné un scénario d'exécution pour en illustrer le déroulement. La phase d'analyse de contexte d'utilisation ainsi que l'analyse de processus d'affaire sont importantes pour comprendre le déroulement du processus et dégager les participants et les types de documents échangés. L'analyse de ces documents donne une idée sur les composants d'information existants pour en extraire des patterns d'utilisation. Le résultat de ces analyses ainsi que les différentes phases dans le processus d'ingénierie de documents mènent à des modèles de collaboration intra et inter-entreprises. Ces derniers gèrent le processus d'affaire entre partenaires dans ce processus. Dans le chapitre suivant, nous définissons le formalisme XML et ses avantages, le web sémantique, et leurs différences.

CHAPITRE 3

XML VS. WEB SÉMANTIQUE

3.1 Introduction

Dans ce chapitre, nous définissons le formalisme XML et ses avantages, le domaine du web sémantique et son utilité. Nous les comparons et soulignons les avantages apportés par le web sémantique dans le domaine de la gestion des processus d'affaires.

3.2 XML et ces avantages

XML (eXtensible Markup Language, ou langage de balisage extensible), est un “métalangage”, mais aussi une organisation des informations. XML a été recommandé par le W3C en 1998, comme une version simplifiée de SGML (Standard Generalized Markup Language), dont il retient plusieurs principes. Le plus important point commun avec le SGML est le fait que tout document XML peut être basé sur un schéma. Les origines du SGML remontent à la fin des années 1960 avec le langage GML (Generalized Markup Language). Le SGML est devenu par la suite une norme ISO, et il a été adopté comme standard en 1986. Il fournit différentes syntaxes de balisage qui peuvent être utilisées pour définir des langages de balisage plus ou moins spécialisés. Il a servi de base pour la définition de XML et HTML.

Un document XML est structuré sous forme d'éléments imbriqués qui correspondent à une structure arborescente (voir Figure 3.1). Cette arborescence possède différents types de nœuds :

- La racine est le nœud de base d'un document XML. Il est unique et englobe tous les autres éléments (nœuds).
- Un élément est un nœud possédant un nom. Les éléments sont encodés à l'aide de balises, qui sont déclarées entre < et >. Le début d'un élément est délimité par une balise ouvrante <nom_balise> et sa fin par une balise fermante </nom_balise>.

Si un élément est vide, on peut le définir avec la syntaxe `<nom_balise/>`. Ces éléments forment la structure du document : ce sont les branches (les balises) et les feuilles (le contenu textuel des balises) de l'arborescence. Ils peuvent contenir du :

- texte : le contenu d'un élément auteur p.e. `Gilles Brassard`,
 - des attributs : l'élément `livre` contient un attribut `annee`,
 - des éléments imbriqués : l'élément `livre` englobe les deux sous-éléments `titre` et `auteurs`.
 - les éléments peuvent avoir un contenu mixte c.à.d. des données textuelles mélangées avec d'autres éléments, p.e. la balise `description` contient du texte ainsi que la balise `isbn`.
- Les commentaires sont considérés comme des nœuds à part entière dans l'arbre XML.

Le point fort de ce métalangage réside dans sa flexibilité et son interopérabilité entre diverses plates-formes de programmation et systèmes d'exploitation. La plus grande différence entre le HTML et le XML est que, contrairement au XML, le HTML contient des attributs et des éléments prédéfinis dont le comportement est bien spécifié. Par contre pour le XML, les auteurs d'un document peuvent créer leurs propres vocabulaires XML spécifiques à leur application ou à leur besoin. Ces balises sont des identificateurs quelconques à choisir comme étant évocateurs du type de leur contenu.

Un document XML est un document auto descriptif et extensible. Sa structure arborescente permet de modéliser une grande quantité de données informatisées. Il facilite l'échange de contenus entre des systèmes d'information hétérogènes, notamment, sur Internet. Nous distinguons deux sortes de contenus : orientés données et orientés document.

Les contenus orientés données utilisent XML en tant que vecteur de données. Ils sont conçus pour être exploités par une machine. XML est généralement utilisé, comme une base de données, pour le stockage et l'échange de données structurées. Le document

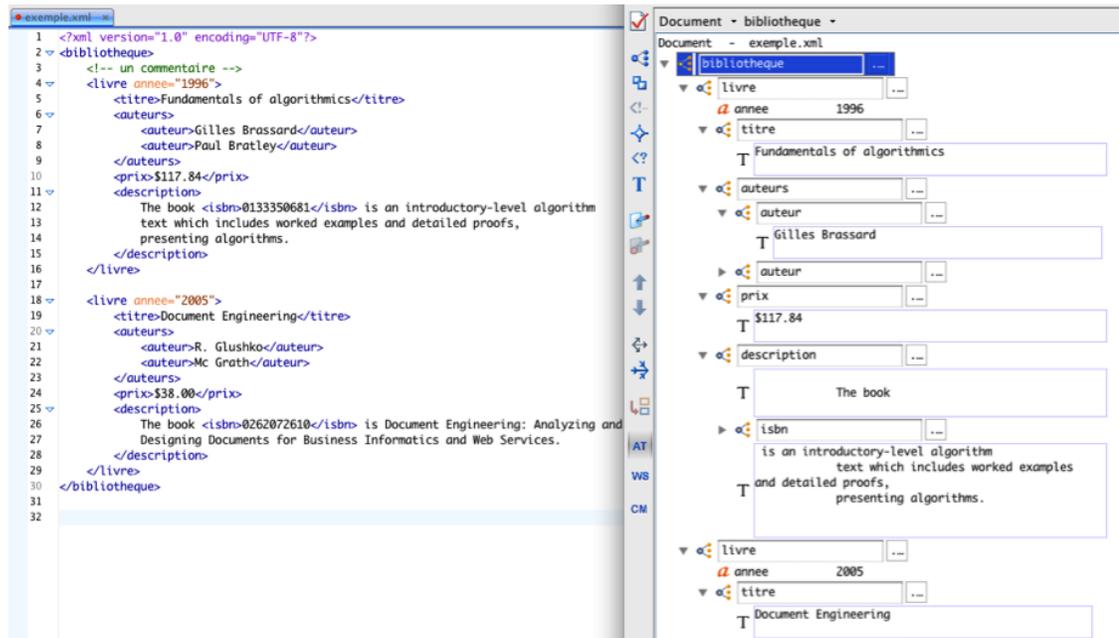


Figure 3.1 – Exemple d’un document XML (à gauche) avec sa représentation sous forme d’une arborescence (à droite). Ce document décrit l’ensemble des livres contenus dans une bibliothèque.

XML est l’unité fondamentale du stockage dans une base de données XML, l’équivalent d’une ligne d’une table dans une base de données relationnelle. Les contenus orientés données sont caractérisés par une structure assez régulière, des données qui présentent une granularité fine, c’est-à-dire que la plus petite unité de donnée est située au niveau d’un élément de type chaîne de caractère ou d’un attribut, il est rare d’avoir du contenu mixte. Il existe plusieurs exemples de contenus orientés données : des relevés des transactions, des factures, des horaires des vols, etc.

La deuxième catégorie d’utilisation XML, généralement connu sous le nom “contenus orientés document”, utilise XML pour séparer la sémantique d’un texte de son contenu, p.e. indiquer que telle partie d’un texte est un titre, un paragraphe, une liste énumérée, etc. Les contenus orientés document sont habituellement conçus pour être utilisés par des humains, comme des livres, des messages électroniques, des annonces, des manuels d’instruction, etc. Ils sont caractérisés par une structure irrégulière, des données qui présentent une granularité plus grande, et beaucoup de contenus mixtes.

Cette structuration permet de séparer l'information (le contenu) de son apparence (le contenant), et donc de fournir plusieurs types de sorties pour un même fichier de données, en fonction de l'utilisateur ou de l'application (un autre document XML, un fichier HTML, un fichier PDF, etc). XML est une partie d'une famille de technologies :

- DTD (Document Type Definition), XML Schema, Relax NG, Schematron : ce sont des langages de définition de documents. La définition d'un langage ou d'un type de document déclare un vocabulaire et des règles supplémentaires : un ensemble d'éléments et d'attributs, un ensemble de valeurs prédéfinies pour certains de ces éléments, une séquence dans laquelle ces éléments peuvent apparaître, la définition des éléments obligatoires, facultatifs, répétitifs, etc. Un document XML, dit bien formé, signifie qu'il respecte les règles syntaxiques de XML garantissant une structure arborescente ; un fichier est valide s'il est bien formé et conforme à un schéma qui lui est associé. Un document XML bien formé est manipulable par un analyseur XML, indépendamment du type de données qu'il contient.
- XPath : est un langage de requête pour sélectionner des parties dans un document XML. En outre, XPath peut être utilisé pour calculer des valeurs (par exemple, des chaînes, des nombres ou des valeurs booléennes) à partir du contenu d'un document XML. Le langage XPath est basé sur la représentation arborescente du document XML. Il offre la possibilité de naviguer dans l'arborescence, et de sélectionner des nœuds en se basant sur une variété de critères.
- XQuery : est un langage de requête pour sélectionner et extraire des éléments et des attributs à partir d'un ou plusieurs documents XML. Il offre aussi un ensemble de fonctionnalités permettant d'effectuer de nombreux types d'opérations sur des données et des documents XML, y compris :
 - La sélection de l'information fondée sur des critères spécifiques.
 - Le filtrage des informations non désirées.
 - Le tri, le groupement et l'agrégation des données.
 - La transformation et la structuration des données XML dans un autre vocabulaire ou structure XML.

- La transformation des données XML en XHTML.

Il est souvent qualifié de “SQL du XML”, puisqu’il est semblable à SQL pour la base de données. Il permet de faire des requêtes complexes sur la structure ou encore les contenus en se basant sur des expressions XPath. Voici un exemple d’une question que XQuery pourrait résoudre en utilisant l’exemple illustré dans la Figure 3.1 : sélectionner tous les livres ayant un prix inférieur à 100\$ de la collection des livres stockée dans le document XML appelé `exemple.xml`.

```
for $book in doc("exemple.xml")bibliotheque/livre
where $book/prix < 100
order by $book/prix
return $book/titre
```

- XSLT : est l’abréviation de Extensible Stylesheet Language Transformation. C’est un langage de programmation qui sert à transformer des documents XML dans divers formats comme HTML, XML, PDF, texte, etc.

3.3 Web sémantique

L’idée d’un Web sémantique a été introduite par Tim Berners-Lee dans les années quatre-vingt-dix [11]. Le contenu des documents HTML est aujourd’hui conçu pour être lu et interprété par les êtres humains. Ce contenu peut difficilement être manipulé et analysé, en tenant compte du sens, par des programmes informatiques. Les ordinateurs peuvent analyser de manière experte les pages Web selon leur mise en forme, mais en général, ils n’ont pas de moyen fiable pour traiter la sémantique. Le défi du Web sémantique est donc de fournir un langage qui exprime des données et des règles de raisonnement sur ces données. L’idée principale du web sémantique est de rendre le contenu accessible et compréhensible par la machine, et de déduire du contenu implicite. Ceci ouvre la voie au développement d’outils sophistiqués susceptibles d’apporter un niveau supérieur de fonctionnalités pour assister les activités humaines sur le web. L’aspect sémantique est exprimé par le langage RDF et encodé par un ensemble de triplets. RDF est un moyen d’identifier les relations entre les ressources sur le web. Il repose sur un modèle de triplet

(sujet, propriété, objet) ou (sujet, verbe, complément) [11] où :

- le sujet est la ressource à décrire.
- le prédicat est un type de propriété applicable au sujet. Le prédicat lui-même est considéré comme une ressource, et identifié par une URI. Un URI est une chaîne de caractères utilisée pour identifier un nom ou une ressource sur Internet.
- L'objet est une donnée ou une autre ressource.

Tableau 3.I – Représentation des triplets RDF sous différentes formes.

Triplets	<ul style="list-style-type: none"> • Jamel {est_rédacteur_de} l'ingénierie de documents. • L'ingénierie de documents. {est_un} rapport.
Format relationnel	<ul style="list-style-type: none"> • est_rédacteur_de (Jamel, l'ingénierie de documents). • est_un (l'ingénierie de documents, rapport).
RDF/XML	<pre><rdf:RDF xmlns:rdf=".../22-rdf-syntax-ns#" xmlns:phd="www.iro.umontreal.ca/jridijam#"> <rdf:Description rdf:about="Jamel"> <phd:est_rédacteur_de> <rdf:Description rdf:about="l'ingénierie de documents"> <phd:est_un> <rdf:Description rdf:about="rapport"/> </phd:est_un> </rdf:Description> </phd:est_rédacteur_de> </rdf:Description> </rdf:RDF></pre>

Le Tableau 3.I représente des Triplets RDF sous différentes formes. Allant de haut en bas, nous voyons : le format en triplet dans lequel chaque élément est une URI ou une donnée dans le cas d'un objet, le format relationnel et le format d'échange RDF/XML. Il existe aussi une notation graphique dont le sujet et l'objet sont liés par une flèche étiquetée. Prenons le texte suivant pour illustrer ce concept :

Jamel rédige un rapport sur l'ingénierie de documents.

Dans la perspective du web sémantique, RDF définit une couche au-dessus de XML. De ce fait, RDF a été doté d'une syntaxe XML. Toutefois, RDF ne fournit aucun mécanisme pour décrire des propriétés ni pour décrire les relations entre ces propriétés et d'autres ressources. C'est le rôle du langage de description de vocabulaire RDF "RDF Schema". RDF Schema est une extension sémantique à RDF. Un schéma est un méta-modèle qui spécifie quels types de ressources peuvent être utilisés dans les triplets RDF. Ces schémas introduisent de nouvelles primitives pour la définition de classes, de propriétés, dont la sémantique formelle s'appuie sur la logique du premier ordre. La définition de classes et propriétés du langage RDFS est similaire aux systèmes de types des langages de programmation orienté objet tels que Java. Par exemple, nous pourrions définir la propriété `author` ayant un domaine `Document` et un type `Person`, alors qu'un système orienté objet classique définirait typiquement une classe `Document` avec un attribut `author` de type `Person`. Les ressources peuvent être divisées en groupes appelés des classes (`rdfs:Class`). Les membres d'une classe sont dits des instances de la classe. Les classes sont elles-mêmes des ressources. Elles sont souvent identifiées par des URI et peuvent être décrites par des propriétés RDF. Il existe plusieurs propriétés qui peuvent exister entre les classes ou ressources ; p.e. les classes `Shoes` et `Clothes` sont des sous classes de la classe `Product`, alors toutes les instances de `Shoes` et `Clothes` seront également des instances de `Product`. La propriété `rdfs:subClassOf` peut être utilisée pour déclarer qu'une classe est une sous-classe d'une autre.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/.../22-rdf-syntax-ns"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema" >
  <rdfs:Class rdf:ID="Product"/>
  <rdfs:Class rdf:ID="Shoes">
    <rdfs:subClassOf rdf:resource="Product"/>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Clothes">
    <rdfs:subClassOf rdf:resource="Product"/>
  </rdfs:Class>
</rdf:RDF>
```

La classe `rdf:Property` est la classe des propriétés RDF. Elle permet d'associer une propriété à une ressource ou classes en utilisant :

- La propriété `rdfs:range` est une instance de la classe `rdf:Property`, utilisée pour déclarer l'ensemble des valeurs possibles d'une propriété (une donnée ou bien une autre ressource),
- La propriété `rdfs:domain` est une instance de la classe `rdf:Property`, utilisée pour déclarer qu'une propriété s'applique à une classe de ressources. Par exemple, la propriété `referencedBy` sera appliquée sur la classe `Product` avec une valeur de type `string`. Par contre dans le deuxième exemple, nous avons la propriété `composedBy` qui s'applique sur la classe `Product`.

Exemple 1 : `Product referencedBy string`

```
<rdf:Property rdf:ID="referencedBy">
  <rdfs:domain rdf:resource="#Product"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</rdf:Property>
```

Exemple 2 : `Order composedBy Product`

```
<rdf:Property rdf:ID="composedBy">
  <rdfs:domain rdf:resource="#Order"/>
  <rdfs:range rdf:resource="#Product"/>
</rdf:Property>
```

La puissance d'expression de RDF et de RDFS est très limitée. Il est impossible de préciser la nature des relations entre ressources (réflexivité, etc.) Ainsi qu'il n'y a pas de contraintes d'existence et de cardinalité.

- RDFS ne permet pas d'exprimer que 2 classes sont disjointes :
 - Exemple : les classes `Homme` et `Femme` sont disjointes.

- RDFS ne permet pas de créer des classes par combinaison ensembliste d'autres classes (intersection, union, complément) :
 - Exemple : on veut construire la classe `Personne` comme l'union disjointe des classes des `Homme` et `Femme`.
- RDFS ne permet pas de définir de restriction sur le nombre d'occurrences de valeurs que peut prendre une propriété :
 - Exemple : on ne peut pas dire qu'une personne a exactement 2 parents.
- RDFS ne permet pas de caractériser des propriétés notamment :
 - Transitivité : `estPlusGrandQue`
 - Unicité : `estLePèreDe`
 - Propriété inverse : la propriété `mange` est inverse de `estMangéPar`.

Ces limites ont donné naissance à un nouveau langage de description d'ontologies, nommé OWL (Web Ontology Language), qui a été recommandé par le W3C début 2004, plus expressif que RDF et RDFS. OWL est réalisé comme une extension de RDF et RDF Schema (RDFS). Il est un langage de description d'ontologies conçu pour la publication et le partage d'ontologies sur le Web sémantique. Une ontologie est un ensemble structuré des termes et concepts utilisés pour décrire et représenter un domaine de connaissance. Les ontologies sont utilisées par des personnes, des bases de données et des applications, qui ont besoin de partager les informations d'un domaine p.e. la médecine, les processus d'affaire, la gestion financière, etc. Les ontologies incluent les définitions des concepts de base dans un domaine, les relations entre eux et la description des connaissances.

OWL permet de définir des classes (concepts) qui représentent des ensembles d'individus (objets, instances) et des propriétés qui définissent les caractéristiques de ces individus et des relations (binaires) possibles entre eux. Aussi, les classes OWL peuvent être définies comme des combinaisons logiques (intersection, union ou complément) d'autres classes. Une ontologie OWL est composée des trois catégories syntaxiques différentes :

- Entités, comme les classes, propriétés et individus : elles constituent les éléments de base d'une ontologie. Par exemple, une classe `Product` représente l'ensemble des produits. De même, la propriété `composedBy` représente la relation entre `Product` et `Order`. Enfin, l'individu `GIOARMANI560SQMB` représente une instance particulière de la classe `Sunglasses` (sous classe de `Accessories`).
- Expressions : représente une combinaison d'entités pour former des descriptions complexes à partir de formes de base.
- Les axiomes sont l'ensemble des énoncés supposés vrais. Par exemple, en utilisant un axiome `SubClassOf`, on peut affirmer que la classe `Shoes` est une sous-classe de la classe `Product`.

Ces trois catégories syntaxiques expriment la partie logique d'une ontologie. Elles sont interprétées dans une sémantique définie avec précision qui permet de tirer des inférences utiles. Par exemple, si l'individu `GIOARMANI560SQMB` est une instance de la classe `Sunglasses`, qui est une sous-classe de `Accessories` (sous-classe de `Product`), on peut dire que `GIOARMANI560SQMB` est aussi une instance de la classe `Product`.

Afin de donner à l'utilisateur un choix entre différents degrés d'expressivité, trois sous-langages de OWL 1 (Figure 3.2) ont été conçus avec des caractéristiques différentes : OWL Lite, OWL LD, et OWL Full.



Figure 3.2 – Les trois sous langages de OWL 1.

- OWL Lite : offre une hiérarchie de concepts simple, facile à programmer, des raisonnements rapides et décidables, mais des contraintes simples de cardinalité 0 ou 1 par exemple une personne a une seule adresse, mais peut avoir un ou plusieurs numéro de téléphone, ce que OWL Lite ne le permet pas.

- OWL DL : se base sur la logique de description, et donne une expressivité plus importante que OWL Lite, en ajoutant l'aspect d'un raisonnement décidable dans un temps fini. Le langage OWL LD comprend toutes les structures de langage de OWL avec des restrictions comme la séparation des types (une classe ne peut pas être en même temps un individu ou une propriété, une propriété être un individu ou une classe).
- OWL Full : offre un haut niveau d'expressivité, OWL tente de trouver un équilibre entre une puissance d'expression et un support logique efficace. La logique (le raisonnement) est un facteur important parce qu'elle permet de :
 - (a) vérifier la cohérence d'une ontologie et des connaissances,
 - (b) vérifier la présence de relations non voulues entre classes, et
 - (c) classer automatiquement les instances en classes.

Depuis la recommandation du W3C en 2004, le Web Ontology Language (OWL 1) a été appliqué avec succès à de nombreux problèmes en informatique. Mais il a aussi révélé des possibilités d'amélioration dans plusieurs domaines. Bien que OWL 1 a eu un succès, certains problèmes ont été identifiés dans sa conception. Une des limitations importantes d'OWL 1 est l'absence d'un ensemble approprié de types prédéfinis, parce que OWL repose sur une XSD (XML Schema) pour la liste des types de données prédéfinis. Depuis Octobre 2007, OWL 1 a été l'objet d'une révision par un groupe de W3C. Ce groupe a commencé à étendre la version proposée en 2004 avec de nouvelles fonctionnalités. La nouvelle version, appelée OWL 2, a été recommandée par le W3C, le 27 Octobre 2009.

OWL 2 est une nouvelle version d'OWL qui améliore considérablement les types de données. Un but dans le développement d'OWL 2 est de fournir un support efficace de modélisation d'ontologies et une plateforme robuste pour le développement futur [9]. OWL 2 ajoute de nouvelles fonctionnalités par rapport à OWL 1. Certaines des nouvelles fonctionnalités sont syntaxiques (par exemple, l'union des classes et les classes disjointes), tandis que d'autres offrent une nouvelle expressivité, y compris : les clés ; les types de données plus riche ; des plages de données ; restrictions de cardinalité et les propriétés asymétriques, réflexives, et disjointes.

Actuellement, il existe plusieurs outils de construction et de vérification d'ontologies p.e. *Protégé-OWL*. *Protégé-OWL* est un éditeur d'ontologies open source. Il offre un support d'interrogation et utilise aussi des moteurs d'inférences qui vérifient la consistance de l'ontologie créée c.à.d. tester la non existence des définitions contradictoires. Il existe plusieurs moteurs d'inférences tels que *Racer*, *Fact*, *Pellet*, etc. La plupart de ces moteurs sont conçus pour raisonner sur les logiques de description mais acceptent en entrée des fichiers OWL. Certains moteurs d'inférence ne peuvent raisonner qu'au niveau des concepts et des propriétés) alors que des moteurs comme *Pellet* et *Racer* permettent de raisonner aussi sur les instances de concepts. De nos jours, il existe plusieurs types de documents et sources d'information assurant la communication entre des parties de différents domaines. Une représentation efficace d'un domaine organise les différents types de documents et dégage les relations inter-documents et inter-domaines. Cette organisation permet d'avoir une recherche plus efficace. Prenons l'exemple du domaine d'affaire, l'automatisation de la gestion de processus d'affaire est encore très limitée [10]. Il n'existe ni une représentation uniforme de l'ensemble de documents dans un processus d'affaire, ni une organisation selon un point de vue sémantique. [10] ont montré que les entreprises ont besoin d'avoir :

- une vue unifiée sur les processus d'affaire,
- une modélisation efficace de leurs processus,
- une représentation de ces processus sous un format lisible par les machines, ce qui permet d'interroger les processus par des requêtes logiques, et
- une vue sémantique de l'ensemble de documents.

Une des solutions proposées est l'utilisation des technologies offertes par le web sémantique. Grâce aux ontologies, nous serons en mesure d'identifier les concepts de base décrivant un domaine de connaissance. L'utilisation d'une ontologie nous permet de définir un vocabulaire commun utilisé pour décrire et organiser des documents. La représentation à l'aide des ontologies offre des avantages tels que le partage d'information entre les parties de différents domaines et les agents logiciels, et la réutilisabilité des connaissances. Durant notre étude, nous utiliserons les ontologies pour la modélisation de docu-

ments dans les processus d'affaire. Dans la section 5.4.2, nous donnons un exemple réel de construction d'une ontologie décrivant le processus d'affaire `Manage Purchase Order` en utilisant OWL 2.

3.4 La différence entre XML et Web Sémantique

Dans l'élaboration du web sémantique, XML fournit la couche de base de la manipulation syntaxique. Bien que XML soit un langage universel pour définir des balises, il ne fournit aucun moyen d'approcher la sémantique des données. Cette sémantique est traitée par les humains et non par les machines. Le web sémantique est venu pour compléter le web et l'analyse de documents, en y intégrant une analyse sémantique. Cet aspect est offert grâce à RDF et OWL. Les ontologies sont devenues vitales pour beaucoup d'applications telles que des systèmes de gestion d'information, et le commerce électronique. Selon Tim Berners-Lee, le web peut atteindre son plein potentiel, s'il devient un lieu où les données peuvent être partagées et traitées par des outils automatisés, ainsi que par les gens. Demain, les programmes devront partager des données et des processus, même si ces programmes ont été conçus de manière indépendante.

Le partage d'information est un problème dans le processus d'affaire surtout au niveau de documents échangés entre les différentes parties du processus. Pour remédier à ce problème, la solution est de permettre à l'ordinateur d'avoir un meilleur accès à la sémantique de l'information [8]. Ainsi, pour un document, nous n'avons pas seulement besoin de stocker ces métadonnées (p.e. son auteur, titre, date de création, etc.), mais nous devons aussi rendre accessibles aux machines les concepts les plus importants abordés dans le document, les relations de ces concepts avec ceux dans d'autres documents, etc. L'utilisation des ontologies est très importante : partager des modèles de documents, dont le but est à la fois partager la sémantique entre les différentes parties de processus d'affaire et rendre l'information interprétable par les machines. Les technologies comme RDF, RDFS permettent la construction des représentations sophistiquées de la connaissance, en plus d'offrir un ensemble de primitives et des éléments pour les différents types de ressources. L'utilisation d'une ontologie permet d'avoir un vocabulaire

commun qui gère la communication entre les entreprises et sert à éliminer les problèmes de chevauchement de sémantique.

3.5 Conclusion

Dans ce chapitre, nous avons présenté le langage XML, le web sémantique ainsi que la différence entre ces derniers. XML est un langage de balisage extensible permettant de définir des langages de balisage. Il n'est lié à aucun langage de programmation, ou des systèmes d'exploitation. Le XML permet d'accéder à une panoplie de technologies pour le traitement, la structuration, la transformation et l'accès de données. Son point fort réside dans sa lisibilité et son interopérabilité. Il permet de structurer un ensemble de documents. Mais son inconvénient majeur est le traitement de la sémantique de données. Cet inconvénient a été remédié par l'apparition du web sémantique et la notion d'ontologie. Cette notion offre la possibilité de créer des concepts, des propriétés et des relations logiques entre eux. Ces avantages donnent une vue sémantique aux données. Nous avons décrit la vision du web sémantique à travers les échanges de documents et la gestion de processus d'affaire.

CHAPITRE 4

ÉTAT DE L'ART SUR LES STANDARDS D'AFFAIRE

4.1 Introduction

L'ingénierie de documents est une nouvelle discipline pour la spécification, la conception et la mise en œuvre des documents échangés électroniquement. Cette discipline est appliquée dans plusieurs domaines afin de faciliter la communication et gérer la grande masse d'informations qui circulent dans/en dehors d'un organisme.

Aujourd'hui, il existe une grande variété de modèles de documents d'affaire qui gèrent l'échange de données entre les partenaires commerciaux. Ainsi, il devient inévitable de traiter les différents documents dans le processus d'affaire afin de concevoir des normes ou des conventions communes pour l'échange électronique de documents.

L'idée d'échange électronique des données entre les entreprises n'est pas nouvelle, elle est mise en œuvre depuis les années 1960. La gestion des échanges électroniques de données dans les affaires inter-entreprises a été dominée pendant longtemps par des standards d'échange traditionnels, comme UN/EDIFACT [12]. L'EDI a offert aux entreprises la perspective, de réduire les coûts, d'éliminer l'échange des documents papier, et d'améliorer leur productivité en utilisant l'échange électronique de l'information. Cependant, seules les grandes compagnies ont eu les moyens de développer des services EDI, et elles imposent à ses partenaires commerciaux de suivre leur modèle commercial. Ceci implique que si un partenaire change son système d'échange, alors tous ses partenaires doivent également changer de leur côté. Plusieurs travaux et standards ont été proposés pour la gestion de documents d'affaire. Ces travaux ont essayé d'utiliser plusieurs techniques existantes (p.e. XML, le web sémantique, etc.) pour donner des meilleurs résultats et remédier aux limites des EDI traditionnels. Il s'avère que les grandes entreprises n'ont pas accepté ce changement vers d'autres techniques. En effet, le coût élevé de l'implantation de l'EDI dans les grandes entreprises montrent qu'ils ne sont pas prêts à réinvestir dans une autre solution.

Avant d'entrer dans les détails de l'utilisation des modèles de processus d'affaires pour soutenir les solutions B2B qui désigne une relation commerciale entre deux entreprises basée sur l'utilisation d'un support numérique pour les échanges d'information. Il est intéressant de regarder l'histoire des initiatives en B2B. Cela permet de classer les différentes initiatives et de voir l'évolution de ces travaux au fur et à mesure de l'apparition de nouvelles technologies.

4.2 Approches basées sur XML

L'échange informatisé de données permet au système informatique d'une entreprise de communiquer et d'échanger des données numériques avec des systèmes informatiques d'autres sociétés. Ce qui permet aux entreprises l'automatisation des échanges électroniques des documents commerciaux tels que les bons de commande et les factures. Environ 90% des grandes entreprises ont investi dans l'EDI [12] et la gestion de documents. Le concept d'échange électronique des documents d'entreprises a conduit à l'élaboration de plusieurs standards, par exemple aux États-Unis le US Transport Data Coordinating Committee (TDCC) en 1968 et GTDI en Europe en 1983. En raison de la mondialisation des échanges, le UN/ECE/WP.4 (un prédécesseur de l'UN/CEFACT) a lancé, au milieu des années 1980, une initiative menant à l'UN/EDIFACT, devenue une norme ISO en 1987. UN/CEFACT est un organisme des Nations unies dont l'appellation en français est *Facilitation des Procédures Commerciales et du Commerce Électronique*, ce qui correspond en anglais à *United Nations Centre for Trade Facilitation and Electronic Business*. Cet organisme encourage la collaboration entre les gouvernements et les entreprises ainsi qu'entre les entreprises, afin d'assurer l'interopérabilité des échanges d'information.

Le langage XML s'est rapidement imposé comme la solution pour la définition des formats d'échange de données sur internet entre les applications du commerce électronique. Il a aidé à remédier les limites des normes traditionnelles, car il utilise un mécanisme d'identification des données par le marquage explicite de l'information. Plusieurs vocabulaires d'affaires basées sur XML ont été développés, nous en présentons ici trois

basées sur XML dans l'ordre de leur apparition : xCBL¹, RosettaNet² et ebXML³. L'effort de standardisation xCBL a commencé en 1997, avant la recommandation W3C du langage XML en 1998. RosettaNet a été lancé en 1998. Le projet ebXML a débuté dans la fin de l'année 1999 pour fournir une infrastructure complète et de répondre aux besoins liés à l'utilisation de XML dans l'e-business. L'e-business traduit en affaires électroniques, présente une vaste dimension du commerce électronique ou e-commerce. Il recouvre les applications faisant appel aux technologies d'information et de communication pour traiter les relations entre une entreprise et ses partenaires. Il utilise des moyens électroniques pour réaliser des affaires, comme le web et l'internet. L'e-business couvre plusieurs processus :

- Les processus de production, qui comprennent l'approvisionnement, la commande et la gestion des stocks, le traitement des paiements, la relation électronique avec les fournisseurs, et les processus de contrôle de la production.
- La gestion des clients, qui comprend les promotions et les efforts de commercialisation, la vente sur Internet, le traitement des bons d'achat, et des paiements.
- La gestion des processus internes, qui comprend les services aux employés, la formation, le partage d'informations internes, et le recrutement.

Il ne faut surtout pas confondre entre l'e-commerce et l'e-business. Le commerce électronique représente un cas particulier et ne couvre qu'une partie de l'e-business. Il utilise les supports électroniques pour tout ou partie des activités commerciales entre une entreprise et les particuliers (la publicité, les commandes en ligne, le paiement électronique, la livraison).

4.2.1 xCBL

Fondé en 1997, xCBL a été initialement appelé Common Business Library (CBL) et développé dans un projet de recherche pour tester les limites de XML pour le commerce

¹<http://xcbl.org/>

²<http://www.rosettanet.org/>

³<http://ebxml.org/>

électronique. En Janvier 1999, CommerceOne acquies la technologie CBL. Cela a conduit à la création de xCBL 2.0, la première version accessible au public.

xCBL est fondamentalement une collection de spécifications XML de documents d'affaires utilisés dans l'e-business. Il fournit une description générale, mais ne définit pas les processus d'affaires ou de messagerie d'une façon détaillée. xCBL est un ensemble de blocs XML et un framework de documents qui permet la création de documents XML solides et réutilisables, pour faciliter le commerce mondial. Il est utilisé comme un langage commun entre les participants dans le domaine du commerce électronique. Cette interopérabilité permet aux entreprises d'échanger facilement des documents d'affaire, en donnant un accès global aux acheteurs, aux fournisseurs et aux intervenants dans un processus d'affaire. xCBL 4.0, la dernière version validée en 2003, supporte tous les documents essentiels et transactions dans le secteur du commerce électronique, y compris l'automatisation de la chaîne d'approvisionnement multi-entreprises, les achats directs et indirects, la planification, les ventes aux enchères, la facturation et le paiement dans un contexte international (paiement en devises). xCBL est disponible dans de nombreux formats : xCBL 4.0 est disponible en tant que W3C XML Schema seulement. Mais les versions précédentes sont fournies comme un ensemble de schémas de SOX (SOX est le langage de schéma créé par Commerce One et utilisées dans ses propres produits). Techniquement, xCBL 4.0 contient 44 documents et utilise plusieurs espaces de noms, où chaque espace de noms représente un domaine fonctionnel, par exemple :

Functional area : core

Namespace : `rrn:org.xcbl:schemas/xcbl/v4_0/core/core.xsd`

Documents : aucun

Description : C'est la bibliothèque de base de xCBL 4.0.

Functional area : ordermanagement

Namespace : `rrn:.../xcbl/v4_0/.../v1_0/ordermanagement.xsd`

Documents : Order, ChangeOrder, OrderResponse, OrderRequest, OrderStatusRequest, OrderStatusResult, OrderConfirmation, OrderConfirmationResponse.

Description : Il contient les documents xCBL associés à la création de bons d'achat et la gestion des commandes.

```

1 <Order xmlns:core="../../../core.xsd" xmlns="../../../ordermanagement.xsd" xmlns:xsi="..."
2   xsi:schemaLocation="rnrn:org.xcbl:schemas/xcbl/v4_0/../../../ordermanagement.xsd">
3   <OrderHeader>
4     <OrderNumber>
5       <BuyerOrderNumber>445882</BuyerOrderNumber>
6     </OrderNumber>
7     <OrderIssueDate>2010-12-10T00:00:01</OrderIssueDate>
8     <OrderReferences> [33 lines]
42    <Purpose> [2 lines]
45    <RequestedResponse> [2 lines]
48    <OrderCurrency>
49      <core:CurrencyCoded>CAD</core:CurrencyCoded>
50    </OrderCurrency>
51    <OrderLanguage>
52      <core:LanguageCoded>en</core:LanguageCoded>
53    </OrderLanguage>
54    <OrderDates> [3 lines]
58    <OrderParty>
59      <BuyerParty> [28 lines]
88      <SellerParty> [50 lines]
139     <ShipToParty>
140       <core:PartyID> [5 lines]
146       <core:NameAddress>
147         <core:Name1>Alain</core:Name1>
148         <core:StreetSupplement1>2222, VIGI AVENUE</core:StreetSupplement1>
149         <core:Building>Building</core:Building>
150         <core:PostalCode>H3J1K7</core:PostalCode>
151         <core:City>Montréal</core:City>
152         <core:Region>
153           <core:RegionCoded>QC</core:RegionCoded>
154         </core:Region>
155         <core:Country>
156           <core:CountryCoded>CANADA</core:CountryCoded>
157         </core:Country>
158       </core:NameAddress>
159       <core:PrimaryContact> [12 lines]
172     </ShipToParty>
173   </OrderParty>
174   <OrderTermsOfDelivery> [4 lines]
179   <OrderHeaderNote>Header Text manually entered</OrderHeaderNote>
180   <ListOfStructuredNote> [9 lines]
190 </OrderHeader>

191 <OrderDetail>
192   <ListOfItemDetail>
193     <ItemDetail>
194       <BaseItemDetail>
195         <LineItemNum>
196           <core:BuyerLineItemNum>GIOGARMANI560SQMB</core:BuyerLineItemNum>
197         </LineItemNum>
198         <ItemIdentifiers> [9 lines]
208         <TotalQuantity>
209           <core:QuantityValue>2</core:QuantityValue>
210         </TotalQuantity>
211         <core:UnitOfMeasurement> [2 lines]
213       </BaseItemDetail>
214       <PricingDetail>
215         <core:ListofPrice>
216           <core:Price>
217             <core:UnitPrice>
218               <core:UnitPriceValue>99.00</core:UnitPriceValue>
219             </core:UnitPrice>
220             <core:Currency>
221               <core:CurrencyCoded>CAD</core:CurrencyCoded>
222             </core:Currency>
223           </core:UnitPrice>
224           <core:CalculatedPriceBasisQuantity>
225             <core:QuantityValue>2</core:QuantityValue>
226             <core:UnitOfMeasurement> [2 lines]
229           </core:CalculatedPriceBasisQuantity>
230         </core:Price>
231       </core:ListofPrice>
232       <core:LineItemTotal>
233         <core:MonetaryAmount>198.00</core:MonetaryAmount>
234       </core:LineItemTotal>
235     </PricingDetail>
236     <DeliveryDetail> [13 lines]
250     <ListOfStructuredNote> [25 lines]
276   </ItemDetail>
277   <ItemDetail> [83 lines]
361 </ListOfItemDetail>
362 </OrderDetail>
363 <OrderSummary>
364   <OrderTotal>
365     <core:MonetaryAmount>198.00</core:MonetaryAmount>
366     <core:Currency> [2 lines]
369   </OrderTotal>
370 </OrderSummary>
371 </Order>

```

Figure 4.1 – Exemple d'un document XML d'une commande conformément à xCBL.

Reprenons l'exemple décrit dans la Figure 2.3 qui présente un bon d'achat d'une commande faite en ligne. Par la suite, nous adoptons à l'architecture xCBL. La Figure 4.1 représente un document XML décrivant notre commande par les schémas offerts dans `ordermanagement`.

4.2.2 RosettaNet

Fondé en 1998, RosettaNet est un organisme indépendant, autofinancé, à but non lucratif regroupant plus de 500 entreprises de haute technologie. Parmi les entreprises qui soutiennent et utilisent les normes RosettaNet, Intel, IBM, Oracle, Netscape, Cisco Systems, Microsoft, et Texas Instruments. Ces membres créent et implémentent des standards pour le traitement des transactions du B2B en particulier au niveau de la chaîne d'approvisionnement. Rosettanet se focalise sur l'industrie informatique et développe des standards verticaux efficaces. Ces standards définissent des schémas XML décrivant une centaine de processus d'échange d'information entre les partenaires en s'appuyant sur Partner Interface Processes (PIP) [2, 14]. Les PIPs sont organisés en sept groupes correspondant à un processus d'affaire particulier (Gestion des profils des partenaires, information des produits, gestion des commandes, gestion des stocks, marketing, service après-vente et support technique, et la fabrication). Ces groupes sont décomposés en segments représentant le point central du réseau commercial. Chaque segment aura un identifiant dans le système de RosettaNet. Nous prenons le cas du groupe 3 (Gestion des commandes) qui est composé de plusieurs segments, par exemple :

- PIP 3A2 : Demande de prix et disponibilité.
- PIP 3A3 : Transfert du panier d'achat.
- PIP 3A4 : Gestion des bons de commande.
- PIP 3A6 : Demande d'état des commandes distribuées.
- PIP 3A7 : Notification d'acceptation du bon de commande.
- PIP 3B4 : Demande d'état de la livraison.

Pour réaliser des affaires électroniques dans le cadre de RosettaNet, les partenaires doivent passer par certaines étapes. Premièrement, les partenaires de la chaîne d'approvisionnement se réunissent et analysent leurs scénarios d'affaires inter-entreprises (par exemple, comment ils interagissent pour faire des affaires les uns avec les autres, quels sont les documents qu'ils échangent et dans quel ordre). Puis ils définissent les processus électroniques mis en œuvre dans le cadre du RosettaNet. La figure suivante présente un scénario d'achat entre deux entreprises dans le cadre de RosettaNet.

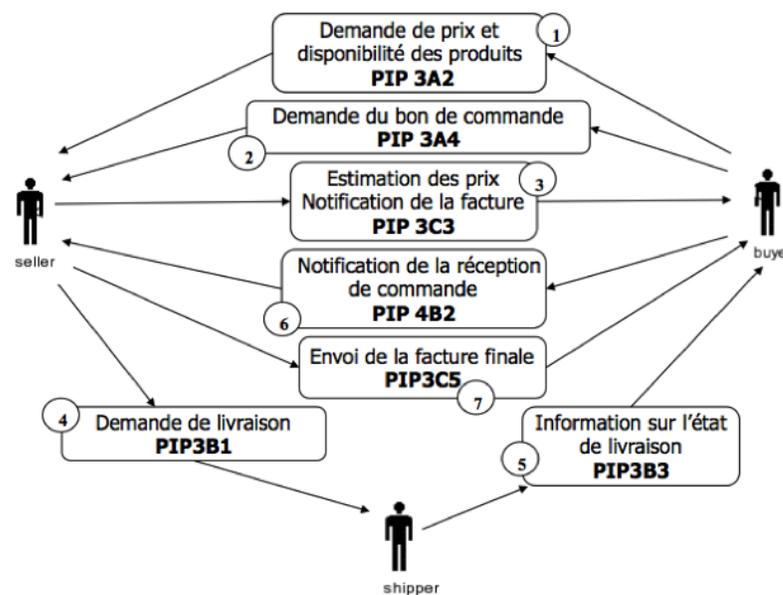


Figure 4.2 – Un scénario d'achat entre deux entreprises dans le cadre de RosettaNet.

Considérons, par exemple, ce scénario où un acheteur demande le prix et la disponibilité de certains produits d'un vendeur (PIP3A2). Après avoir reçu la réponse, l'acheteur initie une demande de commande (PIP3A4) et un document XML sera communiqué, validé par le schéma `3A4_MS_R02_00_PurchaseOrderRequest.dtd`. Le vendeur, d'autre part, traite le bon de commande et envoie une notification d'une facture (PIP3C3) à l'acheteur. Le vendeur envoie une demande de livraison (PIP3B1) au livreur (il y a un tiers dans ce scénario, qui est un livreur). Après la livraison des marchandises, le livreur envoie le statut de la livraison (PIP3B3). Lorsque l'acheteur reçoit la livraison, il envoie une notification de la réception de la commande (PIP4B2) au vendeur. Enfin, le

vendeur prépare une facture finale et en avise l'acheteur (PIP3C5).

```

1 <!-- RosettaNet XML Message Schema
2 3A4_MS_R02_00_PurchaseOrderRequest.dtd (19-Apr-2001 10:
3 This document has been prepared by Edifecs (http://www.
4 based On the Business Collaboration Framework from requi
5 in conformance with the RosettaNet methodology. -->
6 <!ENTITY % common-attributes "id CDATA #IMPLIED" >
7 <!ELEMENT Pip3A4PurchaseOrderRequest (
8   fromRole ,
9   GlobalDocumentFunctionCode ,
10  PurchaseOrder ) >
11
12  thisDocumentGenerationDateTime ,
13  thisDocumentIdentifier ,
14  toRole ) >
15 <!ELEMENT FromRole ( PartnerRoleDescription ) >
16 <!ELEMENT PartnerRoleDescription (
17   ContactInformation? ,
18   GlobalPartnerRoleClassificationCode
19   , PartnerDescription ) >
20 <!ELEMENT ContactInformation (
21   contactName? , EmailAddress? ,
22   facsimileNumber? , telephoneNumber? ,
23   PhysicalAddress? ) >
24 <!ELEMENT contactName ( FreeFormText ) >
25 <!ELEMENT FreeFormText ( #PCDATA ) >
26 <!-->
27 <!-->
28 <!-->
29 <!-->
30 <!-->
31 <!ELEMENT PartnerDescription (
32   BusinessDescription ,
33   GlobalPartnerClassificationCode? ,
34   PhysicalAddress? ,
35   ContactInformation? ) >
36 <!ELEMENT BusinessDescription (
37   GlobalBusinessIdentifier? ,
38   GlobalSupplyChainCode? ,
39   businessName? ,
40   NationalBusinessTaxIdentifier? ) >
41 <!-->
42 <!-->
43 <!-->
44 <!-->
45 <!ELEMENT PurchaseOrder (
46   AccountDescription? , comments? ,
47   ContractInformation* , DocumentReference* ,
48   FinancingTerms* ,
49   generalServicesAdministrationNumber? ,
50   GlobalGovernmentPriorityRatingCode? ,
51   GlobalPurchaseOrderFillPriorityCode? ,
52   GlobalPurchaseOrderTypeCode+ ,
53   governmentContractIdentifier? ,
54   installAt? ,
55   OrderShippingInformation? ,
56   ProductLineItem+ ,
57   proprietaryInformation* ,
58   purchaseOrderIdentifier ,
59   requestedEvent* , requestedShipFrom* ,
60   SecondaryBuyer? , shipTo ,
61   TaxExemptStatus? , totalAmount? ) >
62 <!ELEMENT AccountDescription (
63   accountName , AccountNumber? ,
64   billTo , CreditCard? ,
65   financedBy? ,
66   GlobalAccountClassificationCode? ,
67   prePaymentCheckNumber? ,
68   WireTransferIdentifier? ) >
69 <!-->
70 <!-->
71 <!-->
72 <!ELEMENT PhysicalAddress (
73   addressLine1? , addressLine2? ,
74   addressLine3? , cityName? ,
75   GlobalCountryCode? ,
76   GlobalLocationIdentifier? ,
77   NationalPostalCode? ,
78   postOfficeBoxIdentifier? , regionName? ) >
79 <!-->
80 <!-->
81 <!-->
82 <!ELEMENT CreditCard (
83   cardHolderName , creditCardIdentifier ,
84   expiryDate , GlobalCreditCardClassificationCode ,
85   ProprietaryCIDIdentifier? ) >
86 <!-->
87 <!-->
88 <!-->
89 <!-->

```

Figure 4.3 – Un extrait du 3A4_MS_R02_00_PurchaseOrderRequest.dtd décrivant le RosettaNet PIP3A4 qui représente la demande d'un bon de commande.

PIPs incluent la spécification des partenaires dans un processus d'affaires (l'acheteur, le vendeur, le livreur, etc.); activités commerciales entre ces partenaires, le type, le contenu, et la séquence des documents commerciaux échangés par les interactions entre ces partenaires en exécutant ces activités. Ils précisent également le temps, la sécurité, l'authentification et des contraintes de performance de ces interactions. La structure et le contenu des documents d'affaires échangés sont spécifiés grâce à une DTD [2]. Les partenaires commerciaux qui utilisent les règles de RosettaNet doivent échanger des documents conformes aux DTD décrits dans PIPs. En plus des PIPs, RosettaNet comprend deux dictionnaires, qui fournissent un ensemble de propriétés pour ces PIPs. Ces dictionnaires sont RosettaNet Technical Dictionary et RosettaNet

Business Dictionary. L'échange de documents dans le cadre de RosettaNet est géré par la mise en œuvre de la norme RNIF. Elle englobe le routage, l'emballage et la sécurité. La norme RNIF définit la structure du message, le cryptage du message, et la signature numérique. Elle est basée sur les technologies HTTP, MIME, et les normes XML.

4.2.3 ebXML

ebXML est une suite de spécifications qui permettent aux entreprises de toute taille et dans n'importe quel lieu géographique de mener des affaires sur Internet. Il est basé sur des normes comme XML et UML. En utilisant ebXML, les entreprises ont maintenant une méthode standard pour échanger des données, définir et enregistrer les processus d'affaires. ebXML a commencé en 1999 comme une initiative d'OASIS⁴ et CE-FACT/UN. Fondée en 1993, OASIS est un organisme international sans but lucratif qui s'intéresse au développement et l'adoption de standards dans l'e-business. OASIS compte plus de 5000 participants représentant plus de 600 organisations et membres dans 100 pays.

Le but d'ebXML est de créer un seul marché électronique mondial où les entreprises peuvent communiquer, échanger des données, devenir des partenaires commerciaux et réaliser des affaires. Toutes ces opérations seront effectuées automatiquement en échangeant des documents XML. En conséquence, ebXML est basé sur un ensemble de concepts comme :

- les processus d'affaires (BP) : la définition d'un processus d'affaires consiste à étudier son scénario de réalisation ainsi que la séquence des messages qui aide à gérer ce processus. Ces processus seront modélisés sous forme des diagrammes UML (diagramme de cas d'utilisation et de séquences). Le modèle d'information et de processus d'affaires ebXML permet aux partenaires commerciaux de saisir les détails de leurs scénarios d'affaires. Un processus d'affaires décrit la relation entre des partenaires commerciaux, leurs rôles et les transactions d'affaires effec-

⁴<http://www.oasis-open.org/org>

tuées avec d'autres partenaires.

- Profil de collaboration (CPP) : il consiste à déterminer les processus d'affaire qu'un partenaire commercial peut supporter. Ce document contient les informations d'un partenaire, les processus supportés, les besoins, etc.
- Accord de collaboration (CPA) : Il représente un agrément mutuel entre deux partenaires commerciaux sur le processus d'échanges électroniques de données dans le cadre d'ebXML. Ce document décrit l'interaction entre deux CPP et les besoins des partenaires. Il contient les informations sur la définition des interactions entre les deux parties qui s'engagent sur le type de collaboration. Dans cet exemple, nous montrons le processus de construction d'un CPA. Soit un acheteur A qui interroge le registre ebXML et découvre un partenaire commercial B comme vendeur. Tout d'abord, il télécharge le CPP du vendeur B dans son serveur. Ensuite, il crée un CPA(A,B) qui sera communiqué avec le vendeur B. Enfin, une fois qu'ils sont d'accord, les deux partenaires A et B s'engagent pour l'exécution de la collaboration.
- les composants de données communs (CDC) : ils représentent l'ensemble des informations sémantiques que les partenaires partagent. Ces informations seront communiquées dans les documents échangés au cours d'un processus d'affaires. Un processus d'affaires décrit la relation entre des partenaires commerciaux, leurs rôles et les transactions d'affaires effectuées avec d'autres partenaires. Chaque transaction d'affaires est réalisée par un échange électronique de documents d'affaires contenant un ensemble de composants communs.
- un registre des données : Le registre ebXML est un entrepôt des informations soumises par un organisme. Le contenu de l'information peut être un schéma XML et des documents, des descriptions de processus, des modèles UML, etc. Il fournit un ensemble de services qui permettent le partage d'informations entre les parties intéressées. Ces services sont définis dans le registre de spécification de services (ebRS) [1], en ce qui concerne l'accès au registre et la recherche des informations des partenaires à travers des requêtes soumises au registre. Les informations des

partenaires seront identifiées par des identifiants uniques générés automatiquement par le registre. En plus de ebRS, ebRIM fournit un schéma de haut niveau pour le registre ebXML. Il fournit des informations sur le type de métadonnées qui sont stockées dans le registre ainsi que les relations entre les classes de métadonnées [7].

- une couche de transport et de routage de données : l'envoi, la réception, la notification et l'interrogation des registres sont faits par un service de messagerie ebXML incluant des règles de sécurité.

La meilleure façon de décrire la technologie ebXML est de proposer le scénario d'un cas d'utilisation entre deux partenaires commerciaux. Ce cas d'utilisation commence par la configuration des interfaces ebXML et aboutit à l'exécution d'une transaction d'affaire simple. Avant de passer à la phase d'utilisation d'ebXML, il existe une démarche d'intégration du standard dans les entreprises voulant communiquer. Cette phase est nommée phase de conception passant par la modélisation et le profilage. D'une part, la modélisation consiste à définir les processus, les messages ainsi que les données à communiquer. Ces modèles de processus métiers (BP) seront modélisés, en adoptant la norme UML, sous forme des diagrammes de cas d'utilisation et de séquences. Ces derniers vont être stockés dans les registres ebXML. D'autre part, le profilage consiste à déterminer les processus supportés par l'entreprise comme l'achat et la vente.

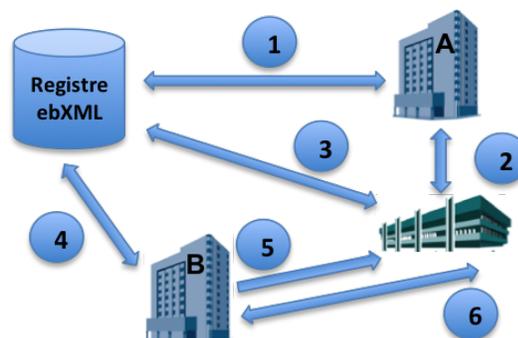


Figure 4.4 – L'approche ebXML- Automatisation des échanges interentreprises.

La Figure 4.4 donne une vision de l'architecture technique d'un système ebXML.

Elle montre un exemple d'échange utilisant la plateforme ebXML et les étapes nécessaires pour configurer puis déployer des applications ebXML.

1. Une compagnie A est informée qu'un registre ebXML est accessible sur Internet.
2. Cette compagnie, après avoir consulté les contenus du registre ebXML, décide de construire et de déployer son propre système ebXML.
3. La compagnie A soumet les informations de son entreprise, appelée Profil d'affaires, au registre ebXML. Le profil soumis à la base de registre ebXML décrit les capacités ebXML de l'entreprise, ainsi que ses scénarios d'affaires supportés. Les scénarios d'affaires supportés par la compagnie A sont déclarés sous forme de versions des documents XML et blocs d'Information qui leur sont associés (p.e les règles de calcul d'une taxe sur les ventes). Après vérification de la conformité des formats et des descriptions des règles d'utilisation du scénario d'affaires, un accusé de réception est envoyé par le registre à la compagnie A.
4. La compagnie B découvre les scénarios d'affaires supportés par la compagnie A en consultant le registre ebXML. La compagnie B envoie à la compagnie A une notification pour établir une relation d'échange électronique ebXML en utilisant un scénario supporté par la compagnie A.
5. Avant de s'engager dans l'exécution du scénario, la compagnie B envoie à l'interface ebXML de la compagnie A une proposition d'accord d'échange. L'accord d'échange proposé montre que les deux compagnies s'engagent mutuellement sur les scénarios d'affaires.
6. La compagnie A accepte l'échange. Les compagnies A et B sont alors prêtes à réaliser des échanges électroniques utilisant ebXML.

freebXML⁵ est une initiative qui vise à favoriser le développement et l'adoption de la technologie ebXML. Elle essaie de créer un site centralisé pour le partage du code ebXML «libre» et les applications de mise en oeuvre de son architecture. Une solution

⁵<http://www.freebxml.org/>

ebXML est l'initiative Oracle WebLogic⁶ qui consiste à échanger des messages d'affaires avec les partenaires commerciaux en utilisant le protocole d'affaires ebXML. Il existe aussi IBM WebSphere Partner Gateway V6.1⁷ qui a ajouté des nouvelles fonctionnalités, des performances et des améliorations d'utilisabilité. Il s'agit notamment de l'intégration de l'architecture ebXML Messaging Service (ebMS) et CPA utilisé au Canada, en Europe et en Asie dans le domaine de la santé et l'industrie des automobiles. IBM WebSphere Partner Gateway est un produit entièrement conçu pour l'échange de données et de documents entre les partenaires d'affaires. Ici au Canada, il y a le projet en cours du gouvernement d'Ontario qui tente d'intégrer l'architecture ebXML dans le e-gouvernement⁸. En plus, il y a la société torontoise Semantion⁹ et la société GetMate¹⁰ qui s'intéressent au développement des outils qui supporte ebXML.

4.3 Approches basées sur les techniques du web sémantique

La fusion entre les techniques du web sémantique et les services web est une des directions de recherche prometteuses pour améliorer l'intégration des applications d'échange, se basant sur la sémantique de données, au sein des entreprises. Plusieurs standards et langages ont été proposés, tels que WSMO [18] utilisé par [14], et OWL-S [15], définissant des frameworks et des langages formels pour les services web basés sur la sémantique.

Dans la section précédente, nous avons étudié différentes approches utilisant le langage XML dans le domaine d'ingénierie de documents et la représentation des processus d'affaire. Au chapitre 3, nous avons étudié le langage XML et les techniques du web sémantique et la différence entre les deux. Le langage XML permet de structurer un ensemble de documents mais non d'exprimer la sémantique des données. Vu ses limites, l'apparition du web sémantique et la notion d'ontologie peuvent amener une solution.

Dans le cadre de ses travaux sur le web sémantique, le W3C a publié en 2004 une re-

⁶<http://docs.oracle.com/cd/E14981-01/wli/docs1031/tpintro/ebxml.html>

⁷<http://www-01.ibm.com/software/integration/wspartnergateway/whatsnew/details.html>

⁸http://www.ebxml.org/case_studies/

⁹<http://www.ebxmlsoft.com/>

¹⁰http://xeni.co.kr/GetMate_info/GetMate_is.html

commandation définissant le langage OWL, afin de modéliser des ontologies utilisables et échangeables sur le web. Une ontologie est la définition formelle de la description d'un domaine de connaissance. OWL décrit la structure d'un domaine en termes de classes et de propriétés. Les classes peuvent être des noms (URI) ou des expressions et supporte un ensemble de propriétés.

Beaucoup de chercheurs se sont concentrés sur le développement de modèles ou d'ontologies donnant une description appropriée de l'information dans les entreprises. Une des premières initiatives dans la modélisation de processus d'affaire a été le projet TOVE [6]. Il vise à développer un ensemble d'ontologies pour la modélisation de processus d'affaires de toutes sortes d'entreprises. Son objectif est l'implémentation d'un ensemble d'axiomes qui aident à déduire des réponses automatiques à propos de l'entreprise [6]. Par ailleurs, le manque d'une version officielle (exprimé en utilisant l'un des langages de construction d'ontologie) et la non-existence des raisonneurs efficaces ont compliqué la mise en œuvre de ce travail.

[4] et [5] ont examiné la façon d'enrichir les registres ebXML par des ontologies OWL pour en décrire la sémantique. D'une part, ils ont étudié les divers axiomes d'OWL qui vont être utilisés pour décrire la hiérarchie d'ebXML, et d'autre part, ils ont proposé une méthode d'interrogation, via des requêtes, de la sémantique stockée dans les registres ebXML. Ils proposent une représentation sous forme d'ontologie OWL du modèle ebRIM [7]. ebXML RIM (Registry Information Model) fournit un schéma de haut niveau pour le registre ebXML. Il fournit des informations sur le type de métadonnées qui sont stockées dans le registre ainsi que les relations entre les classes de métadonnées. La classe de niveau supérieur dans le RIM est le `RegistryObject`. Il s'agit d'une classe de base abstraite utilisée par la plupart des classes dans le modèle. Une instance de la classe `Association` représente une association entre un source `RegistryObject` et une cible `RegistryObject`. Chaque association doit avoir un attribut `AssociationType` qui identifie le type de cette association. Les instances de `ClassificationNode` sont utilisés pour définir des structures d'arbre où chaque nœud de l'arbre est un `ClassificationNode`.

La Figure 4.5 illustre la représentation d'un exemple de `owl:subClassOf` en se

basant sur ebXML RIM. Dans ce cas, une classe OWL a été représentée par une instance de `ClassificationNode` et les propriétés RDF ont été traitées par un objet de type `Association`. Une instance de cette classe relie entre deux objets de type `source RegistryObject` et `target RegistryObject`, décrivant respectivement les classes `ReserveAFlightService` et `AirTransportationService`. Ainsi la propriété `rdfs:domain` peut être mappé par `source RegistryObject` et la propriété `rdfs:range` peut être associée à une `target RegistryObject`.

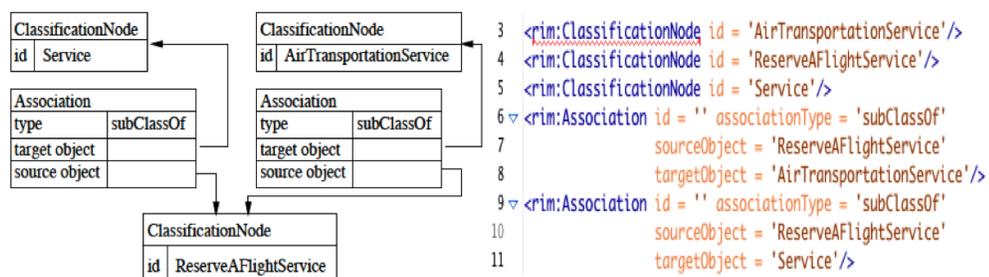


Figure 4.5 – Représentation d’un exemple de la propriété `owl:subClassOf` dans le registre ebXML. À gauche, c’est la représentation d’une instance du mapping avec le modèle ebXML RIM et à droite le langage de représentation de ce modèle en utilisant XML.

[4] ont montré l’effort d’ingénierie consacré à l’intégration d’une ontologie OWL dans un registre ebXML pour donner une vue sur la sémantique de données. Malgré l’importance de cet effort, il reste beaucoup de travail à faire surtout au niveau des raisonneurs. Notamment, le travail de [5] est limité à OWL Lite, tandis que les ontologies OWL-S sont écrites dans le langage OWL DL.

Nous avons vu à la section 3.3 qu’il existe un support d’interrogation ainsi que des moteurs d’inférence qui vérifient la consistance de l’ontologie créée c.à.d. tester la non existence des définitions contradictoires. La plupart de ces moteurs sont conçus pour raisonner sur les logiques de description mais acceptent en entrée que des fichiers OWL. Dans le cadre de ce travail, la représentation des ontologies n’est pas faite en utilisant la syntaxe OWL comme le montre la Figure 4.5. Alors, il est difficile d’appliquer des raisonnements sur l’ontologie ni la vérification de sa consistance.

L’esprit de transition vers les ontologies a été l’objet des études effectuées sur Ro-

settaNet [14]. [14] a étudié les limites de RosettaNet, p.e. le problème d’alignement de processus d’affaire et la définition ambiguë des messages. Ces derniers proposent une solution qui consiste à modéliser les documents d’affaires RosettaNet sous la forme d’une ontologie. Cependant, ils ont opté pour l’utilisation du Framework WSMO (Web Service Modeling Ontology) et les propriétés du langage WSML.

WSMO [3] est un Framework, pour les services web sémantiques, composé d’un modèle conceptuel et d’un langage pour décrire les aspects pertinents des services. Le modèle conceptuel WSMO est exprimé dans le langage de modélisation de services Web (WSML) qui offre une syntaxe pour échanger des données entre les services. WSML distingue les syntaxes conceptuelles et logiques. La syntaxe conceptuelle est utilisée pour modéliser les concepts fondamentaux de WSMO (Web Services, les ontologies, les objectifs et les médiateurs), l’expression logique est utilisée pour décrire les contraintes supplémentaires et des axiomes. Ce langage se compose d’un certain nombre de variantes basées sur différents formalismes logiques et correspondent aux différents niveaux d’expressivité logique, qui sont à la fois syntaxiques et sémantiques. Par l’utilisation du WSML, les auteurs essaient de modéliser les services web et offrir une vue sémantique sur les connaissances et les raisonnements à appliquer suite à l’utilisation des ontologies. Ils ont montré par des exemples réels les détails de leur approche de transformation et les apports des ontologies ajoutés à RosettaNet.

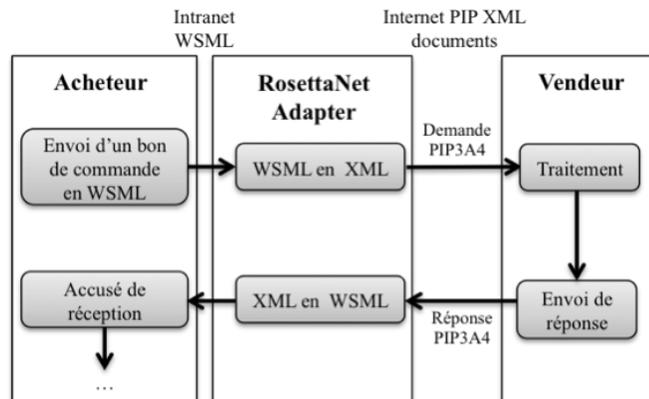


Figure 4.6 – Architecture de RosettaNet Adapter selon [14].

Selon [14], l’étape de transformation était juste un point d’entrée pour la réalisation

d'un échange dynamique. En termes de stratégie de migration pour les infrastructures et processus existants, la solution présentée dans ce travail n'impose pas de changements à l'infrastructure des partenaires d'affaires. Au contraire, les partenaires commerciaux peuvent toujours utiliser leur infrastructure existante, mais le profil de l'acheteur va opter pour une infrastructure basée sur les techniques du web sémantique. L'infrastructure proposée opte pour une couche d'adaptation nommée RosettaNet Adapter. Cette couche transforme les messages envoyés par l'acheteur en format WSMML vers des messages en XML transmis et compréhensible par le nouveau vendeur et vice versa (voir Figure 4.6).

4.4 Synthèse

Nous avons vu les principales normes basées sur XML pour l'organisation des affaires inter-entreprises et la gestion de documents circulant dans les processus d'affaires. [13] a comparé ces deux standards. Ces standards partagent un but principal et commun qui consiste à offrir des normes gérant les affaires entre les entreprises dans le monde entier.

À mon avis, le standard RosettaNet est meilleur qu'ebXML et semble être le plus réussi. Le consortium RosettaNet comprend de nombreux fournisseurs qui offrent des logiciels et des services pour le cadre de RosettaNet. Par contre, RosettaNet cible des segments spécifiques de l'industrie, alors que ebXML vise plus une portée mondiale et tente de faciliter le commerce électronique pour toutes les entreprises. Le cadre ebXML est en avance, à certains égards. Il a recueilli le soutien de consortiums de l'industrie et les fournisseurs par exemple UN/CEFACT et OASIS, mais les implementations semblent rares. ebXML a l'intention de fournir des spécifications pour toutes les industries et domaines. Il fournit des lignes directrices pour la définition des processus inter-entreprises et des composants de base pour les documents commerciaux. Il définit également les registres et l'automatisation de la découverte des profils de partenaires. Mais, le marché n'est probablement pas prêt encore pour les registres ebXML et la communication automatisée entre les partenaires commerciaux, car même une intégration unique entre les partenaires est encore difficile à atteindre. Les descriptions des processus d'affaire

existantes dans le cadre d'ebXML semblent en effet assez généraux et ne permettent pas de réduire l'incertitude et la confusion. Puisqu'ils n'offrent pas une définition spécifique des processus d'affaires, mais fournissent simplement aux partenaires commerciaux, les moyens, et les outils pour réaliser les processus d'affaires. Par contre, RosettaNet définit les processus, les documents commerciaux et de la messagerie d'une façon détaillée. Les spécifications de RosettaNet sont plus concrètes par la description détaillée des processus d'affaire. Ils fournissent des DTD et des dictionnaires pour la validation de données transférées. Par contre, ebXML fournit des méta-modèles décrivant des processus d'affaires, comment capturer la sémantique des données avec des composants de base et le fonctionnement fiable de messagerie. ebXML définit également d'autres spécifications supplémentaires comme la façon de trouver des partenaires commerciaux via les registres et offrir une réalisation dynamique des affaires. Même si RosettaNet est mon choix préféré, il a beaucoup de limites en ce qui a trait au problème d'alignement de processus d'affaire et la définition ambiguë des messages. Premièrement, le processus interne de chaque acheteur doit être ajusté à chaque vendeur, c.à.d. que tous les processus doivent être conformes avec les PIPs [14]. La définition de schémas contient un ensemble d'éléments optionnels qui ne sont pas implémentés par chaque compagnie. Dans ce cas, un travail manuel est nécessaire pour interpréter les nouvelles informations envoyées par un nouveau partenaire. Lorsque le nombre de partenaires augmente, l'échange devient de plus en plus difficile [14].

En plus du but commun, ces standards ont des limites communes. Malgré les succès apportés par ces standards, ils souffrent de certaines limites surtout la sémantique, les relations entre les concepts dans les documents échangés et l'hétérogénéité des échanges. Ces limites sont dues à l'utilisation du langage XML. Vu les limites de ce langage, l'apparition du web sémantique et la notion d'ontologie étaient une solution effective. Ces standards ont été le sujet de plusieurs études d'amélioration au niveau de la sémantique. [4] et [19] ont examiné la façon d'enrichir les registres ebXML par des ontologies OWL pour en décrire la sémantique. Dans le cadre du travail de [4], la représentation des ontologies n'est pas faite en utilisant la syntaxe OWL, ce qui rend difficile la réalisation des raisonnements sur l'ontologie ou la vérification de sa consistance. Il reste beaucoup

à faire d'autant plus que les modèles ne sont pas rédigés en utilisant la syntaxe OWL. Il laisse l'application des raisonnements et la vérification de l'ontologie, une étape difficile qui nécessite une migration vers la syntaxe OWL. La réalisation des raisonnements sur l'ontologie aide à déduire des nouvelles informations cachées et des relations entre les concepts dans les documents. À mon avis, la modification du registre ebXML pour soutenir OWL est nécessaire, afin d'avoir une capacité de raisonnement. Mais, [5] ont déclaré que cette approche nécessite des changements considérables dans l'architecture de registre. Ils ont tenté d'intégrer les ontologies OWL-S dans le registre ebXML, mais la capacité de raisonnement en utilisant les OWL reasoners (p.e. Racer et Pellet) n'est pas offerte jusqu'à date. Notamment, le travail de [5] est limité à OWL Lite, tandis que les ontologies OWL-S sont écrites dans le langage OWL DL. Plusieurs contributions ont été proposées pour la modélisation des services web tels que les langages (OWL-S [15], WSML [3], etc.) qui ont inspiré des principes du langage OWL.

L'esprit de transition vers les ontologies a été l'objet des études effectuées sur RosettaNet [14]. [14] qui a essayé d'intégrer la notion d'ontologie au niveau de processus d'échange avec RosettaNet et en se basant sur le langage WSML [18] qui reste toujours dans le niveau de prototypage. Cette approche utilise une couche d'adaptation qui transforme des documents XML en WSML et vice versa, entre deux partenaires dans le cadre RosettaNet.

4.5 Conclusion

Dans ce chapitre, nous avons présenté les travaux d'ingénierie de documents basés sur XML et les techniques du web sémantique. La structuration, l'interopérabilité, et le traitement de la sémantique de données des processus d'affaire étaient les buts principaux de l'ensemble des travaux étudiés dans ce chapitre. Les entreprises et les organismes ont souffert des problèmes des EDI traditionnels surtout au niveau des processus d'échange. L'apparition de XML était une solution partielle. Son apparition a poussé des grandes entreprises à s'investir pour créer des standards d'échange basé sur XML, comme RosettaNet et ebXML. L'étude faite sur ces travaux a démontré qu'ils souffrent

de certaines limites surtout la sémantique et l'hétérogénéité des échanges. Nous avons vu les solutions apportées par la notion d'ontologie et les propriétés du langage OWL.

CHAPITRE 5

CONTRIBUTIONS

5.1 Introduction

Avant d'entrer dans les détails de notre approche d'utilisation des ontologies dans l'organisation et la gestion de documents d'affaires, il est utile de présenter les problèmes qui se présentent lors d'un échange de documents dans un processus d'affaire et les solutions proposées pour y remédier. Nous avons vu que l'échange de documents peut causer des problèmes liés à la compréhension du sens des mots dans les documents échangés. Le sens d'un mot pourrait changer selon les situations ou contextes. Ce problème est dû au fait que chaque partenaire commercial utilise son propre modèle d'affaire avec une définition personnelle de documents. En plus, le partage d'information est un problème dans le processus d'affaire surtout au niveau des documents échangés entre les différentes parties du processus. La solution à ce problème est de permettre à l'ordinateur d'avoir un meilleur accès à la sémantique de l'information. En se basant sur XML, des standards comme xCBL, ebXML et RosettaNet, ont été développés pour l'organisation des affaires inter-entreprises et la gestion de documents circulant dans les processus d'affaires. Malgré les pistes que le langage XML a offert au niveau de l'organisation et de la structuration, la représentation de la sémantique de données n'est pas encore possible. Cette constatation nous a amené à élaborer une nouvelle technologie pour représenter la sémantique de données, et à offrir un support d'interrogation afin d'extraire de nouvelles connaissances à partir de ces données.

5.2 Description de l'approche

L'utilisation des ontologies est importante et prometteuse. Le partage des modèles de documents permet de communiquer la sémantique entre les différentes parties d'un processus d'affaire et de rendre l'information interprétable par les machines. L'utilisation des ontologies identifie les concepts les plus significatifs et les relations utilisées

dans un domaine de connaissance. Les technologies comme RDF et RDFS permettent la construction de représentations complexes de la connaissance, en plus d’offrir un ensemble de primitives et d’éléments pour les différents types de ressources. Ce qui permet d’avoir un vocabulaire commun qui gère la communication entre les entreprises et d’éliminer les problèmes de chevauchement de sémantique.

Une architecture partageant des ontologies décrit les types de transactions entre un ensemble de partenaires d’affaire pour communiquer avec un langage commun. En plus de l’avantage de l’utilisation des ontologies par rapport au langage XML, nous proposons d’offrir une architecture tenant compte de la description des processus d’affaires à plusieurs niveaux d’abstraction : depuis la description des modèles d’affaires (niveau abstrait) dans le cas d’ebXML, jusqu’à la description des instances de documents (niveau détaillé) dans le cadre de RosettaNet.

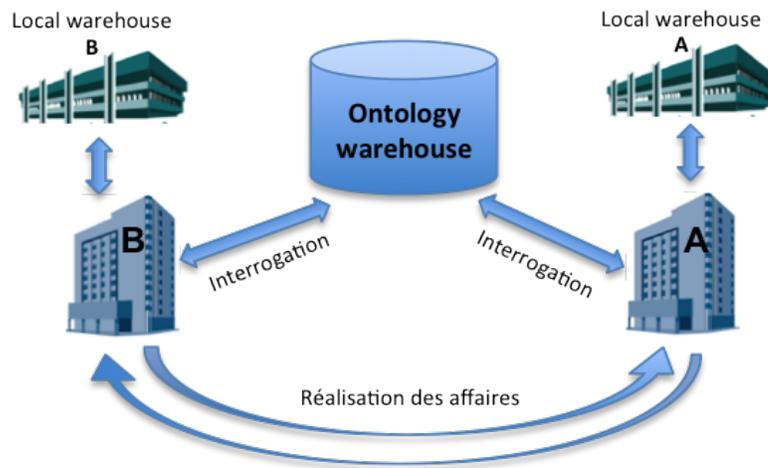


Figure 5.1 – Architecture de l’approche.

La Figure 5.1 résume notre approche. Elle décrit le processus d’affaire, étant donné deux entreprises A et B. Supposons que la compagnie A voulait réaliser des affaires selon un critère défini. Alors, elle interroge l’Ontology warehouse pour extraire les partenaires pertinents à leur critère de recherche. La compagnie B semble pertinente parmi les partenaires supportant notre architecture. Par conséquent, la compagnie A envoie à B une demande de réalisation d’affaire. De son côté, la compagnie B demande le

traitement du profil de la compagnie A. Suite à la réception de l'accord de la compagnie B, les deux compagnies s'engagent mutuellement sur le scénario d'affaire.

Notre *Ontology warehouse* est composé de deux niveaux : (1) niveau conceptuel pour les modèles et la conception de notre ontologie, et (2) niveau de données pour les informations échangées. En plus, il se base sur un ensemble de concepts à un niveau de détails différents.

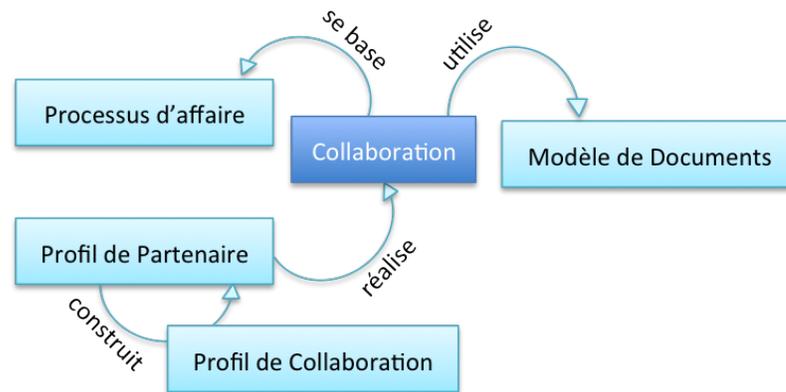


Figure 5.2 – Modèle des concepts. Un partenaire réalise une collaboration avec un autre partenaire en utilisant un modèle de documents, et en se basant sur un modèle d'affaire décrivant le processus d'échange. Une fois, les deux partenaires engagés, un profil de collaboration sera construit.

Ces concepts se basent sur un modèle décrit à la Figure 5.2. Ils sont représentés comme suit :

- **Processus d'affaire (Business Process)** : il comprend la définition d'un processus d'affaires : son scénario de réalisation et la séquence des messages échangés. Nous visons différents types de processus d'affaires (internes ou externes), par exemple :
 - Les processus de production comprenant l'approvisionnement, la commande et la gestion des stocks, les processus de contrôle de la production, etc.
 - La gestion des clients et des produits, qui comprend les promotions, les descriptions de produits, le traitement des bons d'achat, et des paiements, etc.

- La gestion des processus internes, qui comprend les services aux employés, le partage d’informations internes, le service de recrutement, etc.
- Profil des partenaires (*Partner Profile*) : il comprend les informations de chaque partenaire d’affaire, ses besoins, les processus qu’ils supportent et les partenaires avec lesquels il réalise des affaires.
- Profil de collaboration : il représente l’ensemble des interactions effectuées entre deux partenaires d’affaires A et B. Le profil de collaboration est surtout utile lorsqu’un nouveau partenaire C cherche à collaborer avec les partenaires A et B selon les caractéristiques d’échange décrites par notre ontologie. C interroge notre ontologie via des requêtes logiques pour extraire les types de transactions réalisées par A et B. Par la suite, il se base sur ces exemples pour pouvoir collaborer sans avoir un malentendu durant le processus.
- Modèles des documents (*Document Model*) : il représente la description des documents lors de la réalisation d’une transaction d’affaire, ainsi que les composantes sémantiques que les partenaires partagent et la relation entre eux.
- Support d’interrogation et de raisonnement : ce support assure la communication entre les partenaires et facilite l’accès et l’extraction de connaissances. La communication sera effectuée à l’aide des requêtes logiques, bien sûr en tenant compte de l’aspect sécurité de données. Les requêtes peuvent être :
 - Qu’elles sont les compagnies qui offrent des rabais sur certains produits ?
 - Qu’elles sont les compagnies qui existent dans la région de Montréal ?
 - Qu’elles sont les compagnies qui supportent la vente en ligne ?
 - etc.

Les requêtes effectuées par un partenaire seront enregistrées dans le *Partner Profile* pour une utilisation future. Le but est de connaître les préférences d’un partenaire qui sera informé des changements apportés sur le prix d’un produit ou bien s’il y a un nouveau partenaire qui joint notre architecture et qui semble

pertinent. Nous essayons par ce support de rendre la réalisation des affaires plus dynamique et fluide.

5.3 Les phases de l'approche

Nous allons nous inspirer des étapes d'ingénierie de documents décrites dans le chapitre 2 :

1. Analyse de contextes d'utilisation et du traitement de l'ensemble de processus d'affaires existants, c.-à-d. pour chaque processus (p.e. processus de production), il faut comprendre le déroulement du processus, les membres et les séquences des interactions effectuées.
2. L'analyse de documents d'affaires (p.e. bons de commande) consiste à traiter l'ensemble des composantes sémantiques (patterns) que les partenaires partagent et essayer de construire des modèles de documents en se basant sur ces composantes.
3. L'étape de construction de profils de partenaires d'affaires consiste à extraire des informations de description (p.e. adresse(s) physique(s), adresse(s) électronique(s)) ainsi que l'ensemble des processus qu'ils supportent.
4. La phase d'implémentation consiste à construire notre ontologie à l'aide du langage OWL 2.

5.4 Prototype

Nous essayons de transformer le PIP3A4 du XML vu à la section 3.2 vers une ontologie OWL 2 dont les principes sont décrites à la section 3.3. Nous examinons manuellement le schéma XML du PIP3A4 offert par RosettaNet. Nous dégagons les différents concepts décrits dans ce document et nous étudions la sémantique qui les relie.

5.4.1 Description du PIP3A4

Le PIP3A4, nommé Request Purchase Order, appartient au 3ème cluster Order Management et le segment Quote and Order Entry. Ce PIP permet à

un acheteur d'émettre un bon de commande et d'obtenir une confirmation d'un vendeur. Les acteurs principaux dans ce PIP sont le vendeur et l'acheteur (partenaires principaux), dont chacun possède un rôle déterminé par la propriété `Partner Role Description`. Il existe plusieurs interactions dans la réalisation de ce processus (voir Figure 5.3), qui commence par l'envoi du bon de commande par l'acheteur jusqu'à la confirmation du vendeur si le bon a été accepté, rejeté ou retardé. RosettaNet fournit un schéma XML décrivant différents concepts utilisés, tels que : `PurchaseOrderRequest` (le nœud racine) contient l'élément `DocumentHeader` définissant les informations du document (date de création, version, etc) et l'élément `PurchaseOrder` qui présente la spécification de la commande (description des articles, l'identification des partenaires, etc.)

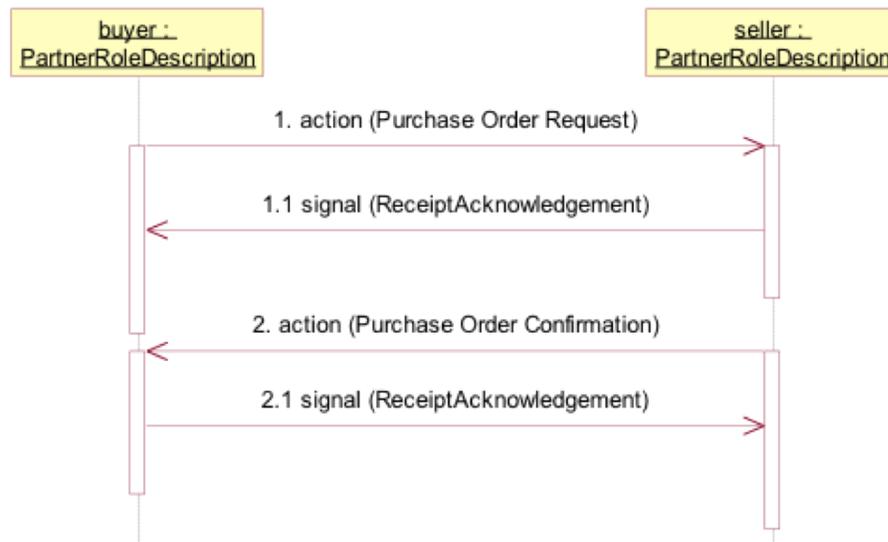


Figure 5.3 – Diagramme de séquence pour la réalisation de commande dans PIP3A4.

Malgré la définition détaillée des processus d'affaires offerte par RosettaNet dans ces différents PIPs, nous constatons un problème de définition ambiguë des messages dû à l'utilisation différente des PIPs. La définition de schémas contient un ensemble d'éléments optionnels qui ne sont pas implémentés par chaque compagnie. Dans ce cas, un travail manuel est nécessaire pour interpréter les nouvelles informations envoyées par un nouveau partenaire. Alors, lorsque le nombre des partenaires augmente, l'échange

devient de plus en plus difficile [14]. Durant notre analyse du schéma XML du PIP3A4, nous avons remarqué que la plupart des éléments définis dans le schéma sont facultatifs, ce qui conforte notre constatation, p.e. l'élément `SpecifiedFullPartner` possède plusieurs éléments optionnels comme `PhysicalAdress` ou bien `ContactInformation` (voir Figure 5.4).

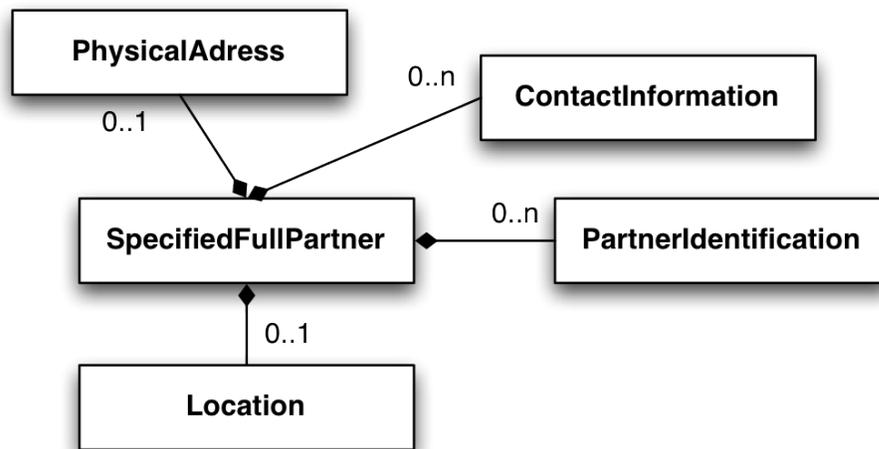


Figure 5.4 – La définition de la structure de l'élément `SpecifiedFullPartner` prise du document `PIP3A4_R11.03.00D_RequestPurchaseOrder` offert par RosettaNet pour définir PIP3A4.

En plus du problème de l'ambiguïté de définition des messages, nous avons remarqué l'existence de plusieurs éléments dont la sémantique est ambiguë : p.e. l'élément `PartnerDescription`, dont on ne connaît pas la nature (acheteur ou vendeur) ; de plus, son utilisation dupliquée dans le document empêche la compréhension de sa sémantique.

L'utilisation des ontologies est importante et prometteuse surtout dans la définition des concepts et leurs relations, p.e. un partenaire peut avoir plusieurs comptes à plusieurs types (compte chèque, épargne, carte de crédit). Le partage des modèles de documents permet de communiquer la sémantique entre les différentes parties d'un processus d'affaire et de rendre l'information interprétable par les machines. L'utilisation des ontologies identifie les concepts les plus significatifs et les relations utilisées dans un domaine de connaissance. Les technologies comme RDF et RDFS permettent la construction de

représentations complexes de la connaissance, en plus d'offrir un ensemble de primitives et d'éléments pour les différents types de ressources. Ce qui permet d'avoir un vocabulaire commun qui gère la communication entre les entreprises et d'éliminer les problèmes de chevauchement de sémantique.

5.4.2 Conception et Implémentation

Nous avons analysé le schéma XML définissant le document PIP3A4, décrit à la section 5.4.1, pour extraire les concepts décrivant ce PIP, les propriétés, et les relations entre ces concepts. Notre travail est décomposé en cinq étapes :

5.4.2.1 Étape 1

La détermination de l'ensemble des termes de l'ontologie (les noms de classes et les noms de propriétés) et puis la construction de la taxonomie (hiérarchie des concepts) représentée dans la Figure 5.5 (1) et décrite dans le Tableau 5.I. Les individus en OWL sont liés par des propriétés. Il y a deux types de propriétés en OWL :

- `Data Property (owl:DatatypeProperty)` relie les individus de classes à des valeurs littérales (`String`, `boolean`, etc.), p.e. `AccountName` (voir Figure 5.5 (2)).
- `Object Property (owl:ObjectProperty)` relie les individus de deux classes, p.e. `describedBy` (voir Figure 5.5 (3)).

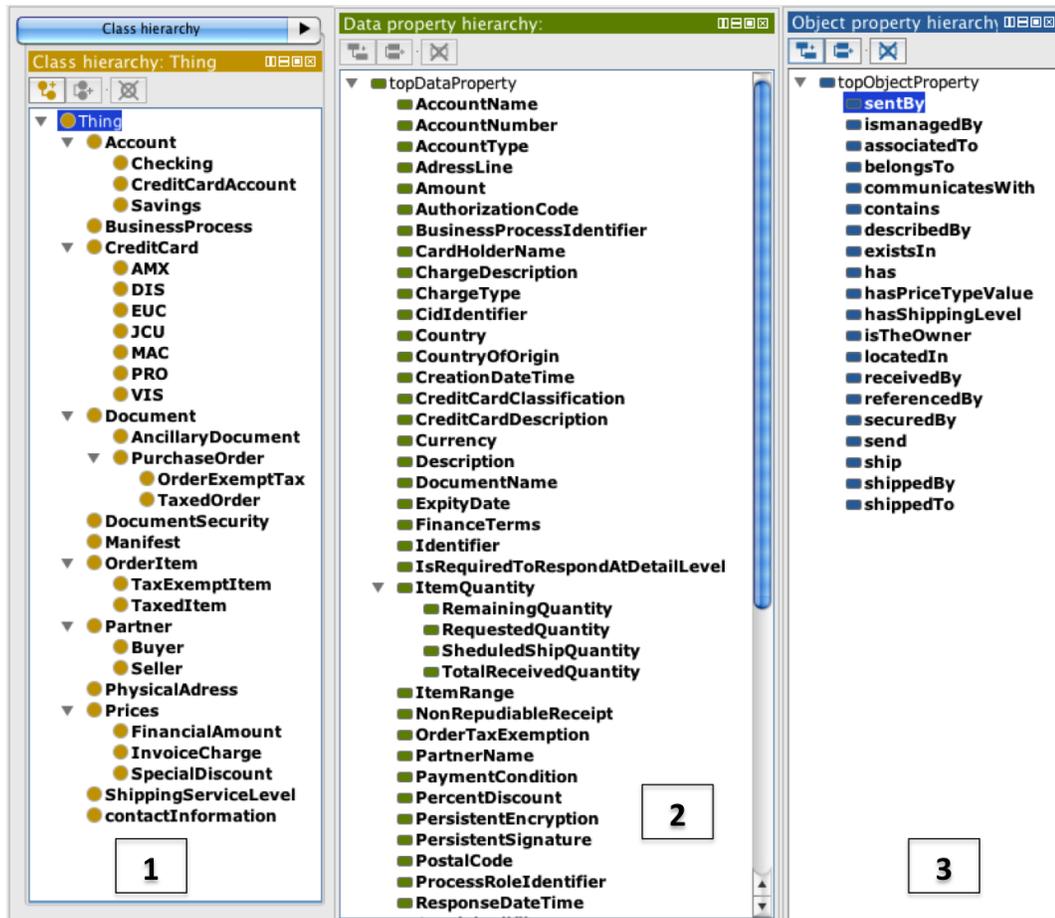


Figure 5.5 – Description de (1) la taxonomie de notre ontologie (liste des concepts), (2) Les Data Property et (3) les Object Property.

Nous avons analysé manuellement le schéma XML afin de déduire une liste des concepts et relations logiques permettant de représenter les connaissances de ce PIP. Nous avons adopté des règles de transformation du XML vers OWL. Nous utilisons l'exemple décrit dans la Figure 5.6, qui présente un extrait du schéma XML associé à l'élément `AccountDescription`, pour montrer les démarches suivies au cours de la transformation.

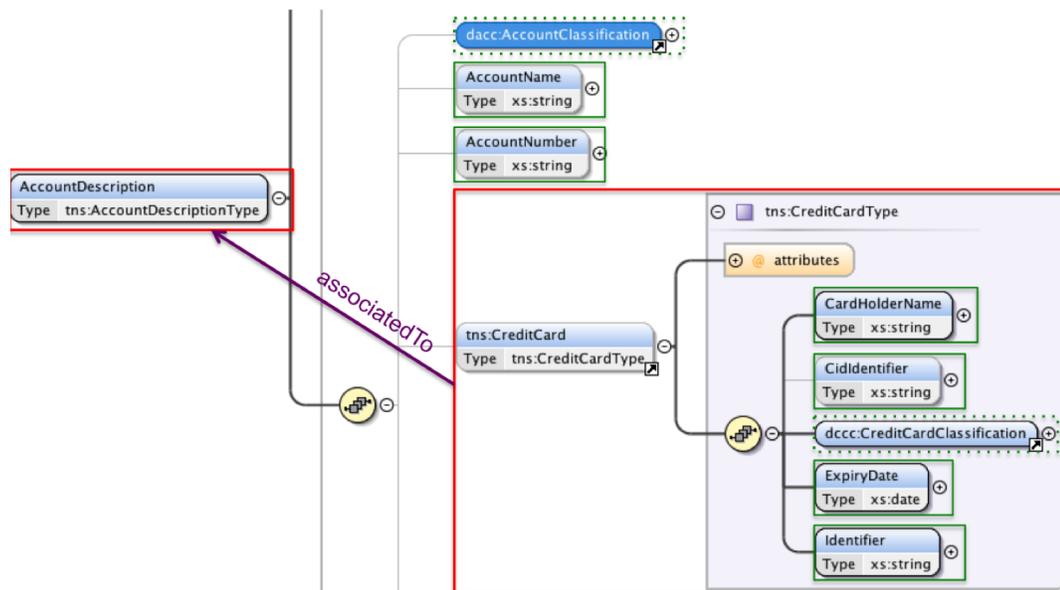


Figure 5.6 – Un extrait du schéma XML décrivant l'élément `AccountDescription`.

Les classes (`owl:Class`) sont décrites par des `xsd:complexType` et `xsd:element` dans le schéma XML. Lorsqu'un élément de l'arborescence est une feuille contenant un littéral et sans attributs, il sera considéré comme un `owl:DatatypeProperty` dont le nom du domaine est celui de la classe représentant son parent, p.e. l'élément `AccountName` est un élément simple ayant un littéral et contenu dans l'élément `AccountDescription`, il sera représenté par `owl:DatatypeProperty` dont le nom du domaine est la classe `Account` décrivant l'élément `AccountDescription`.

Quand un élément de l'arborescence contient un élément complexe ayant non seulement un littéral, une relation de type `owl:ObjectProperty` établit une relation entre ces deux éléments qui vont représenter deux classes. Les éléments `AccountDescription`

et `CreditCard` auront une relation sémantique de type `owl:ObjectProperty` nommée `associatedTo`.

La description des classes OWL peut inclure des restrictions sur les propriétés avec des contraintes de valeurs ou de cardinalité. L'élément `AccountClassification` possède des restrictions de valeurs (CRC, SAV et CHK) que nous avons utilisées pour définir respectivement des sous classes de la classe `Account` (`CreditCardAccount`, `Savings` et `Checking`). L'héritage des classes OWL est représenté par `subClassOf`. Par ailleurs, l'élément `CreditCardClassification` permet de définir des sous classes de `CreditCard` p.e. AMX, VISA, etc.

Le schéma XML peut également contenir des contraintes d'arité comme `xsd:minOccurs` ou `xsd:maxOccurs` représentées en OWL par des contraintes de cardinalité équivalentes `owl:minCardinality` et `owl:maxCardinality`, p.e `CreditCard associatedTo exactly 1 CreditCardAccount`.

Il existe un ensemble de classes disjointes, c-à-d qu'une instance d'une classe ne peut pas être définie par une autre classe, p.e. un compte chèque `Checking` ne peut pas être à la fois un compte d'épargne `Saving`, d'autre part un produit ne peut pas être à la fois un produit exempté des taxes `TaxExemptItem` et taxé au même temps `TaxingItem`. Pour ce faire, nous avons associé à la classe `OrderItem` une propriété booléenne `taxExemption` pour déterminer le type du produit à l'aide de la commande logique suivante `OrderItem and (taxExemption value true)`. OWL 2 introduit `owl:AllDisjointClasses` comme un raccourci qui nous permet de déclarer plusieurs classes disjointes.

Tableau 5.I – Description des concepts utilisés.

concepts	Description
Account	Description d'un compte client ou fournisseur (Dans notre cas Buyer ou Seller), subclassOf -> il peut être Checking, CreditCardAccount ou Savings.
BusinessProcess	Description du processus d'affaire
CreditCard	Description de la carte de crédit peut avoir des sous-classes (AMX, DIS, EUC, JCU, MAC, PRO ou VIS).
Document	Description des documents qui gèrent le processus d'affaire, il peut être un AncillaryDocument, qui est un document auxiliaire définissant un produit, ou PurchaseOrder.
DocumentSecurity	Il offre une sécurité supplémentaire à un document.
Manifest	Information sur les pièces jointes et le contenu du processus.
OrderItem	Description d'un produit dans un bon de commande qui peut être exempté ou non des taxes.
Partner	C'est l'identité d'un partenaire qui peut être un Buyer ou un Seller.
PhysicalAdress	Il représente l'adresse physique du partenaire
Prices	Il représente les types des montants associés à un produit (le prix ordinaire du produit, des charges supplémentaires "InvoiceCharge" ou des rabais "SpecialDiscount")
ShippingServiceLevel	C'est le type de service utilisé pour la livraison, (p.e. ATD : Airport to Door).
ContactInformation	Les informations sur un partenaire (p.e. hasName ou phone)

5.4.2.2 Étape 2

La détermination de l'ensemble des propriétés pour décrire notre ontologie et la détection de l'ensemble des relations complexes (restrictions des valeurs, cardinalité, etc). Concernant les propriétés et les relations entre les classes, OWL 2 permet de déclarer des relations de différentes natures comme les propriétés inverses (`send` et `sentBy` ou bien `ship` et `shippedBy`).

- `Buyer send min 1 PurchaseOrder`
- `PurchaseOrder sentBy Buyer`

5.4.2.3 Étape 3

Nous créons un ensemble d'instances ou d'individus pour chaque classe, ensuite associations des relations entre ces individus (voir Figure 5.7).

Nous avons créé 3 partenaires dont 2 sont des acheteurs (`Lapalme_P3` et `Jridi_P2`) et un vendeur (`Rali_P1`). `Lapalme_P3` possède deux types de comptes (épargne et crédit) ainsi qu'il est propriétaire de deux cartes de crédit chacune associée à un compte. `Jridi_P2` possède 3 comptes dont deux sont des comptes crédit et un compte épargne.

L'instance, décrite dans la Figure 5.7, est nommée `OI6` qui ne nécessite pas une expédition rapide (`isDropShip false`). Ce produit provient d'USA (`CountryOfOrigin "USA"^^string`) avec une quantité de 781 (`ItemQuantity 781.0f`). N.B. que ce produit est exempté des taxes (`taxExemption false`) avec un prix de 10000 CAD (`hasPriceTypeValue Pr5`) avec `Pr5` est de type `FinancialAmount`.

5.4.2.4 Étape 4

La vérification de la consistance de l'ontologie c-à-d raisonner sur ces classes et individus dans la mesure où le permet la sémantique formelle du langage OWL, à l'aide des raisonneurs, p.e. Pellet et Racer. Cette étape consiste à vérifier la non existence des définitions contradictoires dans notre ontologie.

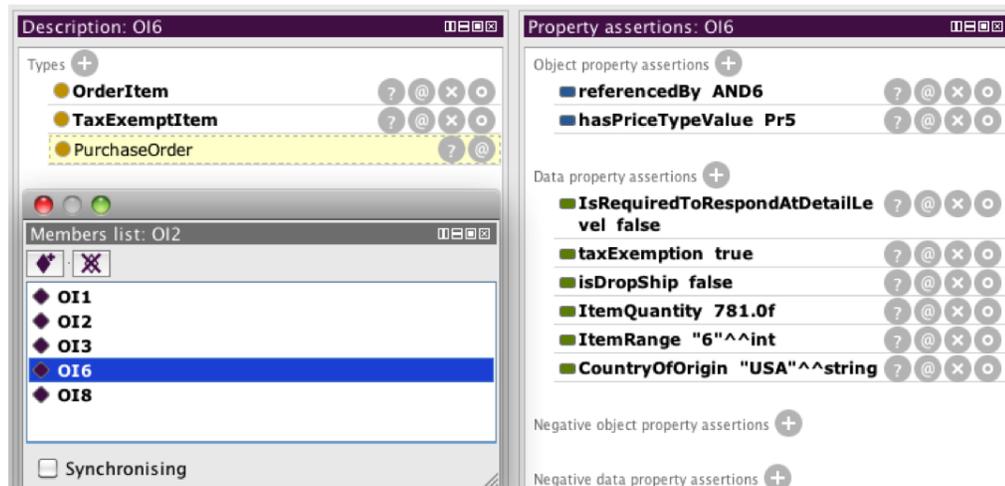


Figure 5.7 – Une instance de la classe OrderItem.

5.4.2.5 Étape 5 : Expérimentation et Validation de l'ontologie

La validité de l'ontologie par des requêtes en logique de description (vérification de l'ensemble de nos propriétés, classes et instances), p.e. nous voulons extraire tous les acheteurs ayant au moins deux comptes de type crédit. La requête sera comme suit : `Buyer and has min 2 CreditCardAccount`. (has) est une propriété de type Object Property qui relie les classes Buyer et CreditCardAccount. Le résultat sera Lapalme_P3 et Jridi_P2 puisqu'ils possèdent deux compte crédit, comme mentionné à l'étape 3.

Ou bien, la requête suivante (`OrderItem CountryOfOrigin value "USA"`) ressort tous les produits provenant des États Unis. L'un des produits sera OI6 comme le montre la Figure 5.7.

5.5 Conclusion

L'ambiguïté sémantique existe encore dans de nombreuses normes d'affaires basées sur XML. Pour résoudre ces limites, nous avons proposé une architecture à base d'ontologie pour le B2B. Nous avons décrit ses composants de base et son principe de fonctionnement. Nous avons présenté une preuve de concepts définissant le PIP3A4

de RosettaNet sous forme d'une ontologie OWL 2. En outre, nous avons validé notre ontologie par différents types de requêtes en logique de description (vérification des propriétés, des classes et des individus).

CHAPITRE 6

CONCLUSION

Dans ce travail, nous nous intéressons à la gestion des documents d'affaires dans le cadre du web sémantique. Dans le chapitre 1, nous avons identifié les problématiques liées à l'échange des documents dans le domaine d'affaires. La structuration, l'interopérabilité, et le traitement de la sémantique de données dans les processus d'affaires étaient les défis majeurs dans les processus d'échange. La solution était l'utilisation de l'ingénierie de documents qui est une nouvelle discipline pour la spécification, la conception et la mise en œuvre des documents échangés électroniquement. Dans le chapitre 2, nous avons défini cette discipline ainsi que les étapes de réalisation.

L'apparition de nouvelles technologies de modélisation de processus d'affaires ainsi que des langages comme XML et le web sémantique, ont aidé à réduire les limites des EDI traditionnels. L'apparition de XML était une solution partielle. Son apparition a poussé des grandes entreprises à créer des standards d'échange basés sur XML. Dans le chapitre 4, nous avons vu les principaux standards basés sur XML comme xCBL, ebXML et RosettaNet. Nous avons décrit le principe de fonctionnement et les composants de chaque standard. Malgré leur impact sur la réalisation des affaires, ces standards souffrent de certaines limites surtout au niveau le traitement de la sémantique des échanges, la possibilité d'effectuer des raisonnements sur les données et l'hétérogénéité des échanges. L'utilisation des ontologies était une piste exploratrice par plusieurs travaux. Elle représente une solution à ce problème. Dans le chapitre 3, Nous avons vu les solutions apportées par la notion d'ontologie et les propriétés du langage OWL. Dans le chapitre 5, nous avons défini notre architecture à base d'ontologie pour la gestion des affaires entre les entreprises. Nous avons conçu une ontologie pour la gestion des bons de commande en se basant sur la définition du PIP3A4 de RosettaNet.

Notre objectif est de rendre possible à la machine d'interpréter les données ainsi que d'effectuer des raisonnements sur ces données. Notre contribution est de créer une ontologie partagée entre les partenaires d'affaire (les applications d'affaire) qui couvre

la plupart des secteurs d'affaire et qui les aident à communiquer à travers un langage commun.

ÉCHÉANCIER

HIVER 2011	Début de la thèse avec le prédoc écrit.
ÉTÉ 2011	Définition de la problématique et l'objectif de notre travail. J'ai étudié le domaine d'ingénierie de documents ainsi que les standards d'affaire existants.
AUTOMNE 2011	Cours Fondements de l'apprentissage machine.
HIVER 2012	Cours Traitement de connaissance et réussite du prédoc spécialité. J'ai eu la possibilité dans le cours du traitement de connaissance de bien aborder le sujet d'ingénierie de documents. En plus, j'ai décrit l'architecture de mon approche ainsi que la construction d'une première ontologie décrivant le processus d'affaire lié à la gestion de bon de commande.
ÉTÉ 2012	Étude d'une transformation automatique des PIPs de rosettaNet du schéma XML vers une ontologie OWL. Ensuite, nous voulons évaluer notre ontologie en utilisant différentes métriques présentées dans plusieurs travaux, comme OntoQA [20].
AUTOMNE 2012	Manipulation de notre ontologie à l'aide du plugin Jena et la réalisation de notre prototype.
HIVER 2013	Conception d'un système d'annotation automatique qui permet de transformer des documents électroniques, sous format pdf ou word, vers notre ontologie.
ÉTÉ 2013	Validation de notre système à l'aide d'une étude empirique.
AUTOMNE 2013	Rédaction de la thèse.
HIVER 2014	Rédaction de la thèse.

BIBLIOGRAPHIE

- [1] OASIS/ebXML Registry Technical Committee et al. OASIS/ebXML Registry Services Specification v2.0 (2002). URL <http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrs.pdf>. consulté le 20 Février 2012.
- [2] S. Damodaran. B2B integration over the internet with XML : RosettaNet successes and challenges. Dans *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 188–195. ACM, 2004.
- [3] J. De Bruijn, H. Lausen, A. Polleres et D. Fensel. The Web Service Modeling Language WSML : An overview. *The Semantic Web : Research and Applications*, pages 590–604, 2006.
- [4] A. Dogac, Y. Kabak et G.B. Laleci. Enriching ebXML registries with OWL ontologies for efficient service discovery. Dans *Proceedings. 14th International Workshop on Research Issues on Data Engineering : Web Services for e-Commerce and e-Government Applications*,, pages 69–76. IEEE, 2004.
- [5] A. Dogac, Y. Kabak, G.B. Laleci, C. Mattocks, F. Najmi et J. Pollock. Enhancing ebXML registries to make them OWL aware. *Distributed and Parallel Databases*, 18(1):9–36, 2005.
- [6] M. Fox. The tove project towards a common-sense model of the enterprise. *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 25–34, 1992.
- [7] S. Fuger, F. Najmi et N. Stojanovic. ebXML Registry Information Model version 3.0 (2005). URL <http://docs.oasis-open.org/regrep/v3.0/specs/regrep-rim-3.0-os.pdf>. consulté le 20 Février 2012.
- [8] R.J. Glushko et T. McGrath. *Document engineering*. MIT Press, 2005.

- [9] W3C OWL Working Group et al. OWL 2 web ontology language : Document overview. W3C recommendation, 27 october 2009. *The World Wide Web Consortium*. <http://www.w3.org/TR/owl2-overview>, 2009.
- [10] M. Hepp, D. Roman et al. An ontology framework for semantic business process management. *Proceedings of Wirtschaftsinformatik*, 2007, 2007.
- [11] P. Hitzler, M. Krötzsch et S. Rudolph. *Foundations of semantic web technologies*. Chapman & Hall/CRC, 2009.
- [12] C. Huemer. ebXML : An emerging B2B framework. *University of Vienna, Institute of Computer Science and Business Informatics*. Vienna : University of Vienna, 2002.
- [13] P. Kantola et J.J. Korhonen. RosettaNet vs. ebXML–Security Solutions and Exception Handling. *Helsinki University of Technology*, 2002.
- [14] P. Kotinurmi, A. Haller et E. Oren. Ontologically Enhanced RosettaNet B2B Integration. *Semantic Web for Business : Cases and Applications*, 2009.
- [15] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne et al. OWL-S : Semantic markup for web services (2004), W3C Member Submission. URL <http://www.w3.org/Submission/OWL-S/>. consulté le 12 Février 2012.
- [16] B. Motik, B.C. Grau, I. Horrocks, Z. Wu, A. Fokoue et C. Lutz. OWL 2 web ontology language : Profiles. *W3C candidate recommendation, W3C (June 2009)* <http://www.w3.org/TR/2009/CR-owl2-profiles-20090611>, 2009.
- [17] RosettaNet Program Office. *Overview clusters, segments, and PIPS*, version 02.13.00 édition, December 2011.
- [18] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler et D. Fensel. Web Service Modeling Ontology. *Applied Ontology*, 1(1):77–106, 2005.

- [19] S. Schulte, M. Siebenhaar et R. Steinmetz. Integrating Semantic Web Services and Matchmaking into ebXML Registry. Dans *Proceedings of the Fourth International Workshop SMR2 2010 on Service Matchmaking and Resource Retrieval in the Semantic Web*, numéro 667, pages 69–83, November 2010.
- [20] S. Tartir et I. Budak Arpinar. Ontology evaluation and ranking using OntoQA. Dans *Semantic Computing, 2007. ICSC 2007. International Conference on*, pages 185–192. Ieee, 2007.