Modélisation de services et de workflows sémantiques à base d'ontologies de services et d'indexations. Application à la modélisation géologique.

Nabil Belaid

▶ **To cite this version:**

Nabil Belaid. Modélisation de services et de workflows sémantiques à base d'ontologies de services et d'indexations. Application à la modélisation géologique.. Sciences de l'ingénieur [physics]. ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechique - Poitiers, 2011. Français. NNT : . tel-00605153

HAL Id: tel-00605153

https://theses.hal.science/tel-00605153

Submitted on 30 Jun 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ecole Nationale Supérieure de Mécanique et d'Aérotechnique
Laboratoire d'Informatique Scientifique et Industrielle

THESE

pour l'obtention du Grade de

DOCTEUR DE L'ÉCOLE NATIONALE SUPÉRIEURE DE MÉCANIQUE ET D'AEROTECHNIQUE

(Faculté des Sciences Fondamentales et Appliquées)

(Diplôme National — Arrêté du 7 août 2006)

Ecole Doctorale : Sciences et Ingénierie pour l'Information
Secteur de Recherche : INFORMATIQUE ET APPLICATION

Présentée par :

Nabil BELAID

****

Modélisation de services et de workflows sémantiques à base d'ontologies de services et d'indexations. Application à la modélisation géologique.

****

Directeurs de Thèse : **Yamine AIT-AMEUR** et **Jean-François RAINAUD**

****

Soutenue le 23 mai 2011
devant la Comission d'Examen

****

JURY

Présidente :	**Nicole LEVY**	Professeur, CNAM, Paris
Rapporteurs :	**Marianne HUCHARD**	Professeur, Université Montpellier II, Montpellier
	Flavio OQUENDO	Professeur, Université de Bretagne Sud, Vanne
Examinateurs :	**Yamine AIT-AMEUR**	Professeur, ENSMA, Futuroscope
	Ladjel BELLATRECHE	Professeur, ENSMA, Futuroscope
	Stéphane JEAN	Maître de Conférences, Université de Poitiers, Poitiers
	Daniel RAHON	Chef de projet, IFP, Solaize
	Jean-François RAINAUD	Chef de projet, IFP, Rueil-Malmaison

Remerciements

Merci à

Yamine AIT-AMEUR, mon directeur de thèse, pour m'avoir guidé tout au long de ma thèse. Je le remercie pour sa disponibilité et pour la partie importante des contributions présentées dans ce mémoire.

Jean-François RAINAUD, mon promoteur de thèse, pour ses conseils avisés et pour sa disponibilité. Je le remercie pour m'avoir permis de participer à un projet ANR et de contribuer à l'encadrement des projets de master.

Marianne HUCHARD et **Flavio OQUENDO** pour m'avoir fait l'honneur d'être rapporteurs de ma thèse.

Nicole LEVY et **Daniel RAHON** pour avoir accepté d'être membres du jury en tant qu'examinateurs.

Stéphane JEAN pour sa disponibilité, sa relecture et ses conseils.

Ladjel BELLATRECHE pour avoir été mon promoteur de master.

Sébastien SCHNEIDER pour avoir accepté l'invitation à ma soutenance de thèse.

François DORIN pour sa relecture.

Guy PIERRA pour m'avoir accueilli au sein du LISI lorsqu'il en était le directeur.

Pierre FERY-FORGUES et **Van BUI-TRAN** pour m'avoir accueilli au sein de l'IFP.

Tous les autres membres du LISI et de l'IFP et particulièrement **Ahmed**, **Beiting**, **Chadlia**, **Chimène**, **Christian**, **Claudine**, **Dago**, **David**, **Frédéric**, **Éric**, **Idir**, **Laura**, **Laurent**, **Loé**, **Manu**, **Michaël**, **Philippe**, **Sébastien**, **Sybille**, **Valéry**, **Yacine**, **Youness** etc. pour tous les bons moments passés en leur compagnie.

Enfin et surtout **Diane**, **mon père**, **ma mère**, **Khelil** et **Zaki** mes deux frères, **Kamel**, **Yasmine** et **Rachid** ainsi que mes amis **Abdenour**, **Amine**, **Chabane** et **Tarik** pour leur soutien. Je leur dédie cette thèse.

Table des matières

Introduction générale **1**

Partie I État de l'art

Chapitre 1 Ontologies et bases de données à base ontologique **7**

1 Introduction 9

2 Les ontologies de domaine 9

2.1 Définition formelle d'une ontologie de domaine 10

2.2 Caractéristiques d'une ontologie de domaine 10

2.3 Comparaison entre ontologies de domaine et modèles conceptuels . 11

2.4 Une taxonomie des ontologies de domaine 12

2.4.1 Les Ontologies Conceptuelles Canoniques (OCC) 12

2.4.2 Les Ontologies Conceptuelles Non Canoniques (OCNC) 13

2.4.3 Ontologies Linguistiques (OL) 14

2.5 Relations entre les différentes catégories d'ontologies : le modèle en oignon 14
2.6 Les modèles d'ontologies 15
2.6.1 PLIB 16
2.6.2 OWL 16
2.6.3 Modèle d'ontologie noyau 16
3 Annotation sémantique 17
4 Les Bases de Données à Base Ontologique (BDBO) 20
4.1 BDBO de type 1 20
4.2 BDBO de type 2 21
4.3 BDBO de type 3 22
5 OntoDB 24
6 OntoQL 25
6.1 Exploitation des ontologies 26
6.2 Exploitation des instances 27
6.3 Exploitation du Méta-modèle (modèle d'ontologies) 28
6.4 Lien entre les niveaux 28
6.5 Interface JAVA pour l'accès à OntoDB : JOBDBC et OntoAPI . . . 29
7 Conclusion 31

Chapitre 2 Les workflows en ingénierie **33**

1 Introduction 35
2 Services Web 35
2.1 Web Service Description Language (WSDL) 36
2.2 Simple Object Access Protocol (SOAP) 37
2.3 Universal Description, Discovery and Integration (UDDI) 39
3 Workflows et compositions de services informatiques 40
3.1 Chorégraphie 41

3.2 Orchestration 41
3.3 Les langages de description syntaxique de workflows 42
3.3.1 Business Process Execution Language (BPEL) 44
3.3.2 XML Process Definition Language (XPDL) 45
3.4 Exécution de workflows 47
3.5 Insuffisances des descriptions syntaxiques de workflows 48
4 Enrichissement sémantique des services et workflows informatiques 49
4.1 Langages de description sémantique de workflows 50
4.1.1 Web Ontology Language for Web Services (OWL-S) . . 50
4.1.2 Web Service Modeling Ontology (WSMO) 52
4.2 Annotation sémantique de langages de workflows existants 53
4.2.1 Semantic Annotation for WSDL (SAWSDL) 53
4.2.2 SEmantic Service MArkup (SESMA) 54
4.3 Workflows scientifiques et autres approches de gestion sémantique de workflows 55
4.3.1 WDO-It! 56
4.3.2 Pegasus 56
4.3.3 Kepler 56
4.3.4 Taverna 57
4.3.5 Triana 57
4.3.6 Vistrail 57
4.3.7 Askalon 58
4.4 Synthèse sur les descriptions sémantiques de workflows 58
5 Conclusion 59

Chapitre 3 Modélisation géologique **61**

1 Introduction 63
2 Géologie et objets géologiques simples 63

2.1 Interfaces sédimentaires 64
2.1.1 Horizon 64
2.1.2 Chenal 65
2.2 Contacts tectoniques 65
3 Modélisation géologique 66
3.1 Étapes de la modélisation géologique : plusieurs modèles géologiques 67
3.2 Interprétation sismique 68
3.2.1 Acquisition d'images sismiques 68
3.2.2 Marqueurs aux puits 70
3.2.3 Extraction d'objets d'intérêt 70
3.2.4 Construction de modèle structural 71
4 Gestion des processus en modélisation géologique 72
4.1 Processus d'interprétation sismique 72
4.2 Implémentation de processus en workflows informatiques 72
5 Conclusion 74

Synthèse générale de l'état de l'art **77**

Partie II Notre proposition

Chapitre 4 Un modèle conceptuel pour une gestion sémantique de services et de workflows **83**

1 Introduction 85
2 Enrichissement sémantique de modèles de données 87

2.1 Modèles de données 87
2.2 Ontologies de domaine 88
2.3 Indexation des modèles de données 90
3 Enrichissement sémantique des services et des workflows informatiques . . 93
3.1 Services informatiques 93
3.2 Ontologies de services 94
3.3 Indexation de services et de workflows informatiques 97
3.4 Modèle conceptuel global 98
3.5 Contrôle sur les ontologies de services 98
4 Construction de nouveaux workflows 100
4.1 Description de workflows sémantiques 101
4.2 Composition fonctionnelle dans les workflows sémantiques 102
4.3 Évolution de l'ontologie des services 103
4.4 Construction et génération de workflows informatiques 104
5 Exploitation de l'approche 104
6 Conclusion 105

Chapitre 5 Modélisation logique de notre approche sur OntoDB 107

1 Introduction 109
2 Modélisation logique de données sur OntoDB 110
2.1 Création de concepts d'ontologie de domaine 110
2.2 Indexation de modèles de données 111
3 Modélisation logique de services sur OntoDB 112
3.1 Modèle d'ontologie de services sur OntoDB 113
3.2 Instanciation de services sémantiques 113
3.3 Indexation de services et de workflows informatiques 114
4 Extension globale d'OntoDB pour la gestion de services sémantiques . . . 114
5 Exploitation du modèle logique sur OntoDB 116

5.1 Recherche de services sémantiques à travers leurs attributs sémantiques . . . 118
5.2 Recherche de services sémantiques à travers des services et workflows informatiques indexés . . . 119
5.3 Recherche de services ou de workflows informatiques . . . 119
5.4 Recherche de services ou de workflows informatiques avec restriction sur les propriétés et attributs sémantiques . . . 120
5.5 Adaptativité de services . . . 121
5.5.1 Adaptativité parfaite . . . 121
5.5.2 Adaptativité spécialisée . . . 122
6 Extension d'OntoQL . . . 123
7 Implantation des contraintes . . . 124
7.1 Implantation de contraintes de covariance . . . 124
7.2 Implantation de contraintes sur la relation entre données sémantiques et modèles de données . . . 126
8 Conclusion . . . 128

Partie III Application à la modélisation géologique

Chapitre 6 Application de notre proposition à la modélisation géologique 133
1 Introduction . . . 135
2 Gestion des données en modélisation géologique . . . 135
2.1 Modèles de données en modélisation géologique . . . 135
2.2 Ontologies de domaine en modélisation géologique . . . 138

2.3 Indexation des modèles données de modélisation géologique 138
3 Gestion des services en modélisation géologique 139
3.1 Services et workflows en modélisation géologique 141
3.2 Ontologies de services en modélisation géologique 141
3.3 Indexation des services et workflows informatiques de modélisation géologique 144
4 Construction de workflows en modélisation géologique 146
4.1 Processus de construction d'un workflow sémantique pour l'interprétation sismique 146
4.2 Composition sémantique en interprétation sismique 147
4.3 Mise à jour de l'ontologies des services de l'interprétation sismique par addition de services sémantiques composés 148
4.4 Construction d'un workflow informatique pour l'interprétation sismique 149
5 Conclusion 150

Chapitre 7 Implémentation de notre approche sur l'interprétation sismique 153

1 Introduction 155
2 Persistance des données de modélisation géologique sur OntoDB 155
2.1 Création des concepts liés à la modélisation géologique 155
2.2 Indexation des modèles de données liés à la modélisation géologique 156
3 Implémentation des services nécessaires à la modélisation géologique sur OntoDB 157
3.1 Création d'ontologies de services de modélisation géologique . . . 158
3.2 Indexation de services et de workflows informatiques en modélisation géologique 158
4 Exploitation de l'implémentation en modélisation géologique sur OntoDB . 160
4.1 Recherche de services à travers leurs attributs sémantiques 161

4.2 Recherche de services sémantiques à travers des services et workflows informatiques indexés 162
4.3 Recherche générale de services ou de workflows informatiques en modélisation géologique 162
4.4 Recherche de services ou de workflows informatiques de modélisation géologique avec restriction sur propriétés 163
4.5 Autres requêtes 165
5 Conclusion 165

Chapitre 8 Geological Workflow Editor (GWE) 167

1 Introduction 169
2 Présentation générale du GWE 169
2.1 GWE sur OntoDB 169
2.2 Profil des utilisateurs 169
3 Réutilisation de services et de workflows existants 171
3.1 Gestion d'ontologies de modélisation géologique 173
3.1.1 Gestion d'ontologies de domaine 173
3.1.2 Gestion d'ontologies de services 174
3.2 Indexation de services et de workflows informatiques 177
3.3 Exploitation des services et des workflows représentés dans le GWE 178
3.4 Exécution de workflows informatiques 178
4 Création de nouveaux workflows 179
4.1 Construction de workflows sémantiques 181
4.2 Validation de workflows sémantiques 182
4.3 Enrichissement de l'ontologie des services 184
4.4 Association des services et workflows informatiques aux services sémantiques 184
4.5 Construction du workflow informatique 185
4.6 Indexation du workflow informatique obtenu 186

5 Conclusion 187

Conclusion générale et perspectives **189**

Bibliographie **197**

Table des figures **207**

Liste des tableaux **211**

Glossaire **213**

Introduction générale

Contexte et problématique

Dans les domaines de l'ingénierie, les services offerts par les logiciels permettent d'assister les scientifiques dans la réalisation de leurs tâches de modélisation, de traitement ou de stockage d'informations. Certains programmes ou services informatiques permettent d'effectuer des travaux répétitifs nécessitant une grande précision, des calculs arithmétiques ou des simulations de manière entièrement automatique. En outre, la composition de ces programmes informatiques permet de créer d'autres programmes plus complexes apportant une plus grande valeur ajoutée. Elle permet la systématisation des tâches à effectuer par les acteurs des domaines de l'ingénierie. Dans les approches traditionnelles, l'activité de composer ces programmes ou logiciels reste à la charge des ingénieurs. Ils doivent s'assurer que les conditions d'une composition correcte sont réunies.

L'un des objectifs visés par des travaux récents concerne l'automatisation du processus de composition de services. En effet, ce processus faciliterait la construction de programmes complexes et offrirait aux ingénieurs un moyen puissant pour construire des analyses et mener des études complexes. Un autre objectif concerne l'enrichissement sémantique de ces programmes informatiques et de leur composition. Ces programmes ou services informatiques et leurs compositions ne sont effectivement pas explicitement liés aux fonctions et aux processus qu'ils implémentent. Aucune ressource excepté leur description syntaxique ne peut fournir d'information sur ce qu'ils réalisent. Cette absence de sémantique explicite rend les logiciels, les services informatiques et leurs compositions difficilement compréhensibles et réutilisables, particulièrement pour les acteurs qui n'ont pas contribué à leur conception ou ayant des connaissances techniques limitées tels que les experts des domaines visés. La découverte et la composition de services informatiques existants en vue d'en construire d'autres réalisant des tâches et des objectifs plus complexes leur sont encore moins envisageables. Ces acteurs restent ainsi très dépendants des développeurs de logiciels que ce soit pour l'utilisation de services et de compositions de services existants ou pour la conception de nouvelles compositions.

En ingénierie pétrolière, domaine que nous avons considéré pour nos travaux, cette situation est avérée. Dans ce domaine, le contrôle des émissions de CO_2 est devenu une préoccupation majeure. Une des solutions proposées consiste à réinjecter le CO_2 produit dans des réservoirs pétroliers en fin de production. Cependant, pour un stockage efficace et sans risque, il est nécessaire d'effectuer une modélisation préalable des réservoirs cibles destinés à contenir le CO_2. Cette activité de modélisation est réalisée par différents experts en modélisation géologique. Afin de modéliser les réservoirs cibles, des données obtenues grâce à des campagnes de mesure doivent être analysées. Par exemple, des images de réservoirs, acquises par des explorations sismiques, sont analysées par des géophysiciens et des géologues afin d'identifier et d'interpréter les structures géologiques qui composent les réservoirs pétroliers. Pour permettre cette interprétation, l'analyse et le stockage de ces données, des experts en géologie communiquent avec des développeurs informatiques qui en établissent des modèles informatiques dont les plus évolués sont utilisés pour des simulations et l'établissement de prévisions d'écoulement de CO_2. Cette tâche de modélisation de réservoirs s'étale sur plusieurs mois. C'est encore au travers de communications entre experts en géologie et développeurs informatiques que des logiciels, services informatiques et compositions sont développés pour implémenter les fonctions et processus décrivant la modélisation géologique. Ils sont cependant conçus sans méthodologie formelle et leurs descriptions syntaxiques sont dépourvues de sémantique portant sur les informations manipulées ou sur les fonctions réalisées. De plus, les compositions définies sont souvent manuelles même si certains systèmes visent leur automatisation.

Pour apporter une solution aux problèmes évoqués, les systèmes classiques de gestion de services informatiques et de leur composition doivent être sémantiquement enrichis pour faciliter leur gestion pour les ingénieurs. Ces systèmes permettraient ainsi de rechercher un service par sa fonction et non par sa description informatique, de remplacer un service par un autre de fonction équivalente ou de construire une composition de services à travers leurs descriptions sémantiques et non à travers leurs descriptions informatiques. Dans cette perspective, de récentes approches ont été proposées. Cependant, leurs mécanismes d'enrichissement sémantique procèdent soit par annotation sémantique soit par extension des langages de description syntaxique ne séparant pas la partie syntaxique utile à l'exécution de la partie sémantique utile à la découverte et à la gestion sémantique en général. Ainsi, les services et leur composition sont surchargés et ne peuvent plus être utilisés sans que de nouveaux outils soient développés. De plus, aucune classification hiérarchique ou relationnelle selon la sémantique des services informatiques n'est fournie rendant leur gestion très restreinte. Nous pensons que pour remédier à ces limitations, il faut permettre d'une part, une séparation claire entre les services informatiques et leur description sémantique, et d'autre part, une hiérarchisation sémantique de ces services informatiques.

Notre proposition

Afin d'atteindre ces objectifs, nous proposons une approche basée sur la définition d'ontologies[1], et plus précisément la définition d'ontologie de services dont les concepts sont des *services sémantiques*. Ces ontologies de services, dont nous avons contribué à construire des prototypes[2], permettent de caractériser les services, leur composition et les modèles de données qu'ils manipulent. Ensuite, nous proposons d'enrichir les services, leur composition et les données d'une couche sémantique en les reliant à leurs caractérisations ontologiques.

Notre objectif final est ainsi de proposer un modèle outillé qui permet aux ingénieurs une meilleure gestion des services et de leur composition, particulièrement dans le domaine de la modélisation géologique. Cette contribution permettrait plus précisément de réaliser deux objectifs :

1. **réutiliser les services et leur composition existants.** Les ingénieurs pourraient ainsi stocker, rechercher et exécuter les services et les compositions définis de manière plus naturelle ;
2. **construire de nouvelles compositions.** Les ingénieurs pourraient ainsi composer les services informatiques existants en vue de construire des compositions réalisant des fonctions plus complexes.

Structure du mémoire

Ce mémoire est organisé en trois parties.

La première partie définit les concepts auxquels nos travaux font référence ainsi que les travaux connexes aux nôtres.

Le chapitre 1 présente les concepts permettant la mise en œuvre de notre approche, à savoir les ontologies, les indexations sémantiques, la structure de base de données à base ontologique (BDBO) OntoDB et son langage de requête OntoQL.

Le chapitre 2 propose de définir les descriptions syntaxiques et sémantiques des workflows[3] et des services Web[4] dont une meilleure gestion est l'objet de nos travaux.

Le chapitre 3 présente le domaine que nous avons considéré pour nos travaux, à savoir le

[1]Une ontologie est une représentation formelle et partagée de la connaissance d'un domaine donné à travers la définition d'un ensemble de concepts et les relations entre ces concepts.

[2]Par définition, les ontologies doivent être consensuelles, c'est-à-dire définies et partagées par les experts d'un domaine donné. Un nombre réduit d'experts en géologie ayant participé à l'élaboration de certaines de ces ontologies, nous les considérons comme des prototypes qui nous permettent de valider notre approche sur l'outil que nous avons développé.

[3]Compositions de services informatiques.

[4]Services informatiques accessibles et exécutables via le Web.

domaine de la modélisation géologique en ingénierie pétrolière. Nous y donnons le vocabulaire nécessaire à la compréhension des insuffisances liées à la gestion des services informatiques et de leur composition dans ce domaine.

La deuxième partie décrit de manière détaillée l'approche que nous proposons.

Le chapitre 4 présente la conceptualisation de notre approche pour la description sémantique de services et de workflows. Nous y montrons comment nous proposons d'indexer les services, les workflows et les modèles de données qu'ils manipulent. Nous y mettons ensuite en évidence comment utiliser cette approche pour construire de nouveaux workflows.

Le chapitre 5 propose d'adapter l'approche proposée dans un modèle logique persistant au dessus de la structure de BDBO OntoDB que nous utilisons comme entrepôt sémantique. Nous y montrons comment nous stockons les services, les workflows, les données qu'ils manipulent et les descriptions ontologiques qui en définissent le sens dans OntoDB à travers son langage de requêtes OntoQL, ainsi que la mise en œuvre des concepts et des contraintes auxquels notre approche fait appel.

La troisième partie concerne l'application de notre approche sur la modélisation géologique et son implémentation.

Le chapitre 6 décrit la conceptualisation de notre approche appliquée à la modélisation géologique. Nous y expliquons comment indexer les services, les workflows de modélisation géologique. Nous présentons les ontologies créées qui permettent ces indexations.

Le chapitre 7 montre l'application du modèle logique sur OntoDB à la modélisation géologique. Nous y montrons le stockage dans OntoDB des services, des workflows de modélisation géologique et des ontologies conçues. Nous expliquons ensuite l'exploitation sémantique qui est ainsi permise.

Le chapitre 8 présente le *Geological Workflow Editor (GWE)* qui est le prototype logiciel que nous avons implémenté pour la gestion sémantique des workflows en modélisation géologique.

Enfin, nous finissons notre mémoire en rappelant nos contributions relatives à la gestion des services et des workflows et en introduisant les perspectives ouvertes par nos travaux.

Première partie

État de l’art

Chapitre 1

Ontologies et bases de données à base ontologique

Sommaire

1 Introduction 9

2 Les ontologies de domaine 9

2.1 Définition formelle d'une ontologie de domaine 10

2.2 Caractéristiques d'une ontologie de domaine 10

2.3 Comparaison entre ontologies de domaine et modèles conceptuels 11

2.4 Une taxonomie des ontologies de domaine 12

2.4.1 Les Ontologies Conceptuelles Canoniques (OCC) . . . 12

2.4.2 Les Ontologies Conceptuelles Non Canoniques (OCNC) 13

2.4.3 Ontologies Linguistiques (OL) 14

2.5 Relations entre les différentes catégories d'ontologies : le modèle en oignon 14

2.6 Les modèles d'ontologies 15

2.6.1 PLIB 16

2.6.2 OWL 16

2.6.3 Modèle d'ontologie noyau 16

3 Annotation sémantique 17

4 Les Bases de Données à Base Ontologique (BDBO) 20

4.1 BDBO de type 1 20

4.2 BDBO de type 2 21

4.3 BDBO de type 3 22

5 OntoDB 24

6 OntoQL 25

6.1 Exploitation des ontologies 26

6.2 Exploitation des instances 27

6.3 Exploitation du Méta-modèle (modèle d'ontologies) 28

6.4 Lien entre les niveaux 28
6.5 Interface JAVA pour l'accès à OntoDB : JOBDBC et OntoAPI . . 29
7 Conclusion 31

Résumé. Dans nos travaux, nous nous intéressons à l'enrichissement sémantique de la description de programmes ou services informatiques ainsi qu'à leur persistance en bases de données. Dans ce chapitre, nous présentons les notions d'ontologies de domaine, d'indexation sémantique et de base de données à base ontologique que nous utilisons dans notre proposition.

1 Introduction

Pour répondre au besoin de gestion des programmes et processus informatiques, notre proposition se base sur l'enrichissement de leur description ainsi que leur stockage en base de données pour assurer le passage à l'échelle. Les ontologies sont des modèles permettant d'expliciter la sémantique d'un domaine d'étude. Leur gestion en base de données est possible en utilisant des structures de bases de données particulières nommées bases de données à base ontologique (BDBO). Ainsi, ontologies et bases de données à base ontologique apparaissent comme des notions pertinentes pour l'enrichissement sémantique et la persistance des programmes et processus informatiques. Aussi, ce chapitre présente ces deux notions en montrant leur intérêt pour notre approche.

Ce chapitre est organisé comme suit. Dans la section 2, nous présentons les ontologies de domaine représentant des caractérisations sémantiques et permettant d'expliciter la sémantique. Nous évoquons particulièrement ce qui caractérise les ontologies de domaine par rapport aux modèles conceptuels. Nous présentons ensuite ce que sont les annotations sémantiques dans la section 3. Dans la section 4, nous présentons le concept de BDBO représentant des entrepôts sémantiques de données et permettant la persistance d'ontologies et des données dont elles définissent le sens. Nous détaillons le cas d'une BDBO appelée OntoDB. Nous présentons également OntoQL qui représente un langage de requête pour BDBO. Enfin, nous donnons une conclusion sur ces concepts.

2 Les ontologies de domaine

La notion d'ontologie trouve son origine dans une branche de la philosophie traitant de la science de l'être. Initiée par Aristote, cette discipline essaie de définir les propriétés générales de l'être à travers ce qui le caractérise de façon essentielle. Le terme "ontologie" a été introduit en informatique dans les années 70 par McCarthy dans le domaine de l'Intelligence Artificielle [Psyché et al., 2003]. Par analogie, "ontologie" est repris pour désigner l'ensemble structuré des termes et concepts représentant le sens d'un domaine d'informations. Le concept d'ontologie est utilisé dans de nombreux domaines incluant le traitement du langage naturel, la fouille d'informations, le commerce électronique, le Web Sémantique, l'ingénierie et l'intégration d'informations, etc. Cette diversification de domaines d'utilisation a vu naître de nombreuses interprétations de ce terme.

L'une des définitions pour la notion d'ontologie qui est la plus répandue est celle donnée par initialement par [Gruber, 1993] et complétée par [Borst, 1997] : "une ontologie est une spécification formelle d'une conceptualisation partagée". Ainsi, une ontologie constitue en soi un modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts. Elle est également employée pour raisonner à propos des objets du domaine concerné.

Vu la diversité des interprétations de cette notion, nous commençons par donner, dans cette section, une définition formelle d'une ontologie de domaine. Ensuite, nous mettons en évidence les principales caractéristiques des ontologies de domaine en précisant ce qui les distingue des modèles conceptuels. Nous présentons par la suite, une taxonomie des ontologies de domaine et indiquons les relations existantes entre les différentes catégories identifiées dans cette taxonomie. Enfin, nous exposons les modèles d'ontologies qui permettent la définition de ces ontologies de domaine.

2.1 Définition formelle d'une ontologie de domaine

Formellement, une ontologie peut être définie comme un quadruplet [Pierra, 2003a] `O = <C, P, Sub, Applic>`, avec :

- `C` : ensemble des classes utilisées pour décrire les concepts d'un domaine donné (comme les services de voyages, les pannes des équipements, les composants électroniques, etc.). Chaque classe est associée à un identifiant universel unique ;
- `P` : ensemble des propriétés utilisées pour décrire les instances de l'ensemble des classes de C. Nous supposons que P définit toutes les propriétés susceptibles d'être présentes dans une base de données. Chaque propriété est associée à un identifiant universel unique ;
- `Sub : (C` $\rightarrow 2^C$`)`[5] est la relation de subsomption qui, à chaque classe C_i de l'ontologie, associe ses classes subsumées directes[6] ;
- `Applic : (C` $\rightarrow 2^P$`)`, associe à chaque classe de l'ontologie l'ensemble des propriétés applicables pour chaque instance de cette classe.

Cette définition présente une ontologie comme un modèle objet. Cependant, une ontologie possède des caractéristiques originales par rapport aux modèles usuels.

2.2 Caractéristiques d'une ontologie de domaine

Une ontologie est une conceptualisation ayant les caractéristiques pricipales suivantes [Jean et al., 2006b] :

- *formelle*. Une ontologie est une conceptualisation basée sur une sémantique formelle qui permet d'en vérifier la consistance et/ou de réaliser des raisonnements et des déductions à partir de ses concepts et de ses instances ;
- *consensuelle*. Une ontologie est une conceptualisation acceptée par une communauté et n'est ainsi pas conçue spécifiquement pour un besoin particulier. Au contraire, elle décrit les concepts d'un domaine d'étude de manière la plus objective possible pour satisfaire les besoins techniques et métiers partagés par l'ensemble des membres de la communauté. Ainsi, deux systèmes développés au sein de la même communauté et portant sur le même

[5] désigne l'ensemble des parties de C.

[6] C1 subsume C2 si et seulement si $\forall x \in C2, x \in C1$.

domaine d'application peuvent être basés sur la même ontologie. L'ontologie fournit alors une interface d'accès commune à ces deux systèmes, indépendante de leurs implantations particulières, qui permet de réaliser certains traitements automatiques comme l'échange d'informations entre deux ou plusieurs systèmes d'informations ou l'intégration de ces systèmes. Précisons que la communauté dans laquelle une ontologie est consensuelle doit être plus large que celle impliquée dans le développement d'un système particulier ;
– *référençable.* Chaque concept d'une ontologie est associé à un identifiant permettant de le référencer à partir de n'importe quel environnement, indépendamment de l'ontologie dans laquelle il a été défini. Dans le Web sémantique, le mécanisme d'URI (Uniform Resource Identifier) est utilisé pour réaliser cette identification universelle.

2.3 Comparaison entre ontologies de domaine et modèles conceptuels

Les ontologies et les modèles conceptuels présentent à la fois des similitudes et des différences [Spyns et al., 2002][Fankam et al., 2008b]. Comme les modèles conceptuels (MC), les ontologies conceptualisent l'univers du discours au moyen de classes associées à des propriétés et hiérarchisées par la relation de subsomption. Les principes de bases de la modélisation formelle sont donc similaires. Par contre, les différences suivantes ont été identifiées entre ces deux types de modèles :

– *objectif de modélisation.* Les MC prescrivent l'information qu'ils représentent dans un système informatique particulier. Au contraire, les ontologies décrivent les concepts d'un domaine indépendamment de toutes applications et systèmes informatiques particuliers dans lesquels l'ontologie pourrait être utilisée ;
– *consensualité.* Le caractère consensuel des ontologies permet de représenter de la même façon les mêmes concepts dans tous les systèmes d'une "communauté". Au contraire, un MC est spécifique à une application donnée ;
– *identification des concepts.* Les classes et les propriétés définies dans les ontologies sont associées à des identifiants universels, ce qui leur permet d'être référencées à partir de n'importe quel format ou modèle, indépendamment de leur structure. Au contraire, la conceptualisation effectuée dans un MC ne possède pas de mécanisme permettant de le référencer de l'extérieur ;
– *souplesse de description.* Toutes les instances des classes d'une ontologie, peuvent ne pas initialiser les mêmes propriétés. Elles n'ont pas forcement la même structure. Cette souplesse dans la description des instances est permise par le fait que les concepts des ontologies sont consensuels et ainsi contiennent toutes les propriétés possibles. Ainsi, seul un ensemble de propriétés définies dans l'ontologie peut être pertinent dans l'application considérée. Cette caractéristique a pour conséquence de rendre les ontologies utiles pour des échanges ou intégrations de systèmes informatiques permettant à chacune d'utiliser les propriétés pertinentes dans son contexte tout en conservant un référentiel commun.

Même si toutes les ontologies présentent les caractéristiques évoquées précédemment, leurs objectifs de modélisation peuvent varier et ainsi amener à différentes catégories.

2.4 Une taxonomie des ontologies de domaine

Différents types d'ontologies existent. Une première distinction peut être faite suivant que l'ontologie vise à conceptualiser l'ensemble des termes décrivant le domaine dans un certain langage, ou bien l'ensemble des objets du domaine. Ces deux manières de conceptualiser conduisent à distinguer les *ontologies linguistiques* (OL) des *ontologies conceptuelles* (OC) [Cullot et al., 2003][Pierra, 2003b]. Suivant [Pierra, 2003b], les OL, sont les ontologies dont l'objectif est la représentation de la signification des termes utilisés dans un domaine particulier et dans une langue naturelle donnée. Les OC sont les ontologies dont le but est la représentation des catégories d'objets et de leurs propriétés dans un domaine particulier.

Une seconde distinction parmi les OC peut être faite. En effet, au sein même de la catégorie des OC, les ontologies présentent des caractéristiques différentes selon le domaine d'application où elles sont utilisées, selon le modèle d'ontologie avec lequel elles sont décrites et surtout selon que celles-ci définissent un langage canonique ou qu'elles introduisent une redondance. Deux catégories d'ontologies peuvent alors être distinguées : les *ontologies conceptuelles non-canoniques* (OCNC) et les *ontologies conceptuelles canoniques* (OCC). Dans ce qui suit, nous présentons les trois catégories d'ontologies de domaine recensées : OCC, OCNC et OL.

2.4.1 Les Ontologies Conceptuelles Canoniques (OCC)

Les *Ontologies Conceptuelles Canoniques* (OCC) sont les ontologies dont les définitions ne contiennent aucune redondance. Dans les OCC, les définitions se focalisent sur l'identification et la caractérisation des concepts canoniques qui représentent un domaine cible. Ces descriptions sont fournies par des constructeurs orientés OCC permettant, par exemple, de représenter le contexte dans lequel chaque élément d'une ontologie est défini [Pierra, 2003b]. Par contre, elles n'incluent pas d'opérateurs permettant de définir des équivalences conceptuelles pour définir d'éventuels mappings entre des ontologies ou dériver de nouveaux concepts. La définition et l'exploitation d'OCC sont principalement réalisées dans les domaines de l'échange de données entre systèmes d'informations. Un exemple d'une OCC standardisée, définie pour le commerce électronique peut être consulté dans [IEC61360-4, 1999].

Comme exemple d'application des OCC, nous pouvons citer le projet STEP, dans lequel des modèles d'échanges sont définis dans le langage EXPRESS [Schenk et Wilson, 1994] pour différentes catégories d'objets techniques. Ils peuvent alors être utilisés dans l'industrie pour échanger de façon unique et non ambiguë des descriptions de différents produits du domaine entre différentes organisations.

Pour illustrer la notion d'OCC, la figure 1.1 présente deux OCC sur le domaine des vins.

Ces OCC définissent le concept de vin, sachant qu'un vin est décrit par une couleur. L'OCC (A) est *orientée propriétés* ce qui signifie qu'une classe n'est créée dans cette ontologie que lorsqu'elle est nécessaire pour définir le domaine d'une propriété. Ainsi, cette ontologie ne définit qu'une seule classe nommée (`A_WINE`) et la propriété `color`. A l'opposé, l'OCC (B) est *orientée classes*, ce qui signifie qu'elle utilise des classes pour caractériser ses instances plutôt que des propriétés. Ainsi, dans cette ontologie, différentes classes sont créées pour les différentes couleurs de vins possibles. Dans cette exemple, la classe des vins rouges nommée `B_REDWINE` ainsi que celle des vins blancs nommée `B_WHITEWINE` sont présentées. L'appartenance d'une instance à l'une de ces deux classes permet de connaître sa couleur puisque cette information est portée par cette classe.

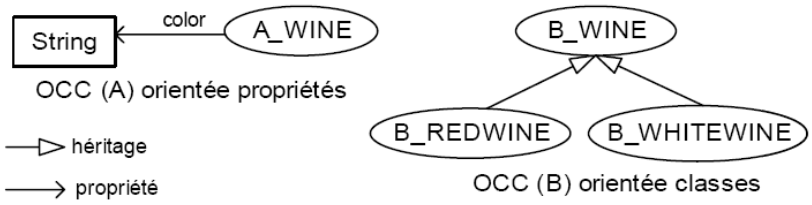


FIG. 1.1 – OCC locales sur le domaine des vins.

2.4.2 Les Ontologies Conceptuelles Non Canoniques (OCNC)

Les *Ontologies Conceptuelles Non Canoniques* (OCNC) sont les ontologies qui contiennent non seulement des concepts canoniques, mais aussi des concepts non-canoniques.

Les OCNC sont particulièrement utiles lorsqu'il est nécessaire d'établir des équivalences conceptuelles. C'est le cas lorsque plusieurs bases de données sont construites suivant différents schémas. Les constructeurs OCNC peuvent être utilisés pour faciliter leur intégration. Ils sont également très utiles pour définir des mapping entre différentes ontologies du même domaine. Ces mappings sont réalisés à travers des opérateurs issus de la logique de description tels que l'intersection ou l'union. Cela permet d'implanter des mécanismes de raisonnement dans des outils nommés *raisonneurs* tels que RACER [Haarslev et Möller, 2001].

Pour illustrer nos propos, reprenons l'exemple des deux OCC de la figure 1.1. A l'aide d'opérateurs permettant de créer des OCNC, avec une syntaxe issue des logiques de description, nous pouvons écrire l'équivalence conceptuelle suivante.

$\texttt{B_REDWINE} \equiv \texttt{A_WINE} \sqcap \texttt{color} : \texttt{red}$

Cette définition indique que la classe `B_REDWINE` définie dans l'OCC (B) est équivalente à la classe `A_WINE` définie dans l'OCC (A) restreinte par la valeur de la propriété `color`. En conséquence, les deux OCC locales peuvent être automatiquement fusionnées dans l'OCNC de la figure 1.2. Cette OCNC fournit un accès global aux données des deux OCC, soit en laissant les données dans chacune des bases de données, soit en les matérialisant dans une seule base

de données. Cet exemple montre que les constructeurs d'OCNC sont utiles pour les tâches d'intégration.

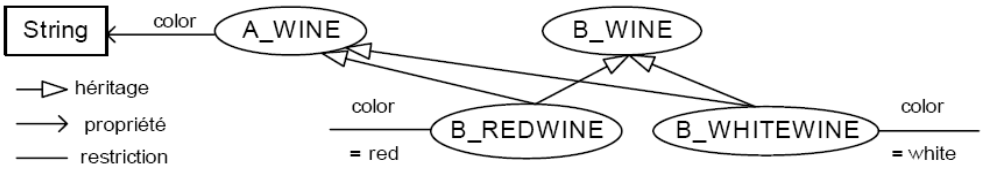


FIG. 1.2 – OCNC intégrée sur le domaine des vins.

2.4.3 Ontologies Linguistiques (OL)

Les *Ontologies Linguistiques* (OL) sont les ontologies qui définissent l'ensemble des termes qui apparaissent dans un domaine particulier. Dans ces ontologies, des relations supplémentaires entre les termes (par exemple, la synonymie ou l'homonymie) sont définies. Les relations entre les termes étant souvent très contextuelles, les inférences automatiques basées sur ces ontologies nécessitent, en général, la supervision d'un expert.

La définition et l'exploitation d'OL sont traitées par les communautés de recherche d'informations et de traitement du langage naturel. WordNet[7] représente un exemple de telles ontologies. Chaque terme est associé à plusieurs *synsets* (ensemble de synonymes) qui représentent la palette de ses significations. La signification des termes dépend du contexte et les relations entre les mots (par exemple, la similarité) n'ont pas de définition formelle. Ainsi, de telles ontologies sont considérées comme imprécises.

Les OL sont utilisées dans de nombreux domaines tels que l'intégration semi-automatique de sources de données. Par exemple, dans le projet MOMIS [Beneventano et al., 2000], une OL est utilisée pour construire semi-automatiquement le schéma global du système d'informations intégré à partir des différentes sources de données. Ce processus doit néanmoins être entièrement validé par un expert humain.

Ayant décrit les différentes catégories d'ontologies de domaine, nous mettons en évidence, dans ce qui suit, la relation entre ces catégories.

2.5 Relations entre les différentes catégories d'ontologies : le modèle en oignon

Les travaux de [Jean et al., 2006b] ont permis d'identifier des relations entre les différentes catégories d'ontologies décrites précédemment. Un modèle en couches est proposé, appelé modèle en oignon, pour la conception d'ontologies permettant d'intégrer ces différentes catégories d'ontologies. Ce modèle, présenté sur la figure 1.3, contient les couches suivantes :

[7] http://www.cogsci.princeton.edu/wn

– en son centre, une OCC fournit une base formelle pour modéliser et échanger la connaissance d'un domaine ;
– une OCNC peut être construite au dessus de cette OCC pour fournir les mécanismes pour lier, à l'aide des opérateurs de non-canonicité, les différentes conceptualisations faites sur ce domaine ;
– enfin, une OL fournit une représentation en langage naturel des concepts de ce domaine, éventuellement dans les différents langages où ces concepts sont significatifs.

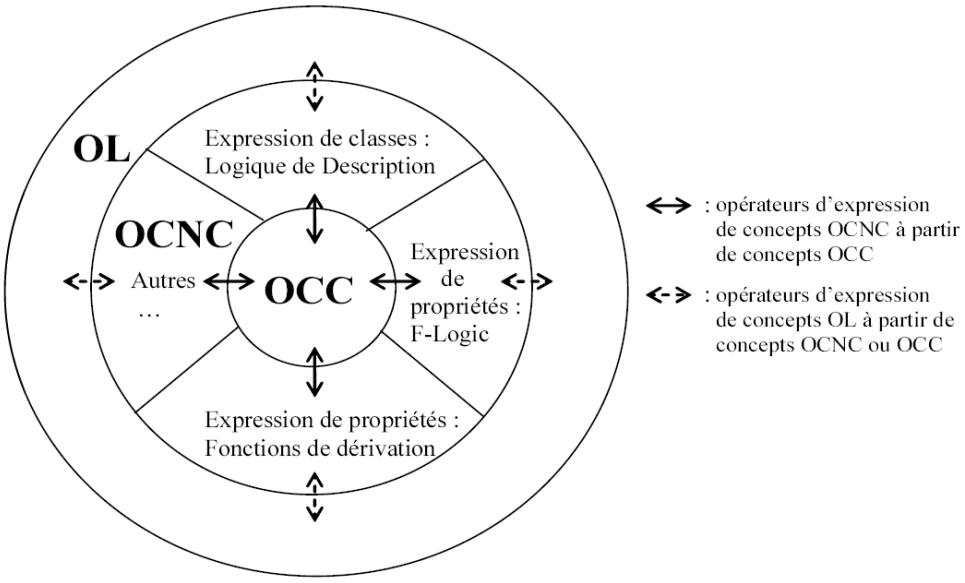


FIG. 1.3 – Le modèle en oignon.

Aussi, ce modèle permet ainsi de combiner les capacités des différents types d'ontologies. Les ontologies que nous utilisons pour l'enrichissement sémantique de programmes et processus informatiques seront basées sur ce modèle. Nous présentons à présent les formalismes permettant de construire des ontologies afin d'en définir un pertinent pour notre approche.

2.6 Les modèles d'ontologies

Un modèle d'ontologies permet de représenter des ontologies à travers les entités et les relations qu'elles contiennent. De nombreux modèles issus de différents champs disciplinaires ont été proposés pour représenter des OCC et des OCNC. Les modèles d'ontologies pour la représentation d'OL restent encore peu formalisés. Des éléments linguistiques sont néanmoins représentables à travers différents modèles d'ontologies permettant de représenter des OCC ou des OCNC.

Le modèle PLIB [ISO13584-42, 1998][ISO13584-25, 2004] issu du domaine des bases de données et le modèle OWL [Dean et Schreiber, 2004] issu des logiques de description et du Web sémantique sont deux exemples de modèles d'ontologies standardisés permettant de représenter

respectivement des OCC et des OCNC. Nous les présentons dans ce qui suit. Nous mettons également en évidence les similitudes et différences entre ces modèles d'ontologies.

2.6.1 PLIB

PLIB (Parts LIBrary) [ISO13584-42, 1998][ISO13584-25, 2004] est un modèle d'ontologies qui permet principalement de représenter des OCC. A l'origine, son objectif était de permettre une modélisation informatique des catalogues de composants industriels afin de les rendre échangeables entre fournisseurs et utilisateurs. PLIB permet de définir avec précision des concepts canoniques. Sa spécificité est de permettre la représentation du contexte de définition d'une ontologie [Pierra, 2007][Pierra, 2003b]. Il est principalement utilisé pour caractériser de manière précise les concepts d'un domaine particulier. En revanche, PLIB ne contient pas d'opérateurs qui permettent de définir des équivalences entre concepts.

2.6.2 OWL

OWL (Web Ontology Language) [Dean et Schreiber, 2004] est un modèle d'ontologies qui permet de représenter des OCNC. Un objectif important d'OWL est de permettre de déterminer, dans un temps fini, si oui ou non les définitions fournies par une ontologie OWL peuvent être déduites à partir de celles d'une autre ontologie [Horrocks et al., 2003]. OWL permet de représenter des concepts canoniques et fournit des opérateurs de non canonicité permettant de définir des équivalences entre concepts tels que *unionOf, intersectionOf, allValuesFrom* etc. Les concepteurs d'OWL ont spécifié trois versions qui permettent différents niveaux d'expressivité et de décidabilité : OWL Lite, OWL DL et OWL Full.

2.6.3 Modèle d'ontologie noyau

Comme nous venons de le voir, les modèles d'ontologies sont différents. Certains proposent un ensemble de constructeurs permettant de définir plus précisément et de façon contextuelle les concepts primitifs d'une ontologie (PLIB). D'autres introduisent des constructeurs d'équivalences conceptuelles permettant ainsi de construire une OCNC (OWL). Toutefois, bien que les modèles d'ontologies soient différents et spécifiques à la représentation d'OCC et d'OCNC, ils ont des constructeurs communs qui constituent un modèle noyau d'ontologies [Jean, 2007]. La figure 1.4 présente les principaux éléments de ce modèle noyau d'ontologies dans un diagramme UML. Les classes et propriétés de ce modèle sont appelées entités et attributs.

Dans la figure 1.4, les concepts suivants sont représentés :

– une ontologie (*Ontology*), aussi appelée espace de noms (*namespace*), qui regroupe la définition d'un ensemble de concepts qui sont des classes et des propriétés ;
– une classe (*Class*) est la description abstraite d'un ou plusieurs objets semblables. Ces

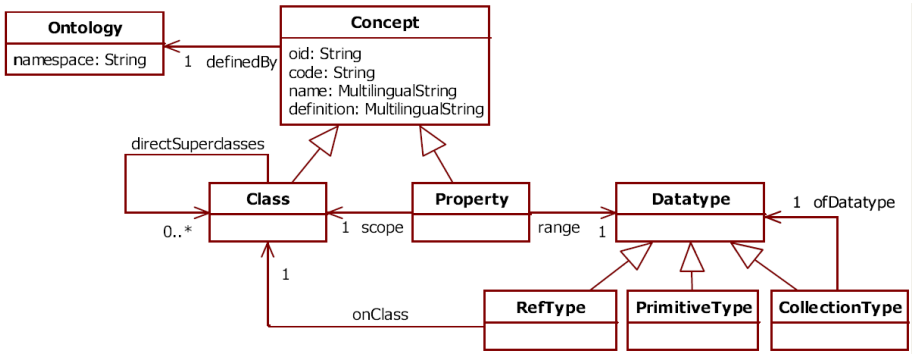


FIG. 1.4 – Modèle d'ontologie noyau.

classes sont organisées dans une hiérarchie acyclique où elles sont liées par une relation d'héritage multiple (*directSuperClasses*) ;

– les propriétés (*Property*) sont des éléments qui permettent de décrire les instances d'une classe. Chacune des propriétés doit être définie sur la classe ou sur l'une des super-classes des instances qu'elle décrit (*scope*). Chaque propriété a un co-domaine (*range*) qui permet de contraindre son domaine de valeurs ;
– le type de données (*Datatype*) d'une propriété peut être un type simple (*PrimitiveType*) tel que le type entier ou le type *chaînes* de caractères. Une propriété peut également prendre ces valeurs dans une classe en faisant référence à ses instances (*RefType*). Enfin, cette valeur peut être une collection dont les éléments sont soit d'un type simple soit des références à des instances d'une classe (*collectionType*).

Ce modèle noyau comportant les principaux constructeurs de modèles d'ontologies, parmi lesquels PLIB et OWL utilisés dans les domaines d'application de nos travaux, nous définissons nos ontologies en nous basant sur celui-ci. Pour prendre en compte les particularités de notre approche, nous étendons ce modèle noyau en utilisant la relation de spécialisation afin de conserver la sémantique des constructeurs initiaux.

Les ontologies permettent ainsi une description sémantique d'un domaine. Elles peuvent être utilisées pour l'enrichissement sémantique à travers un mécanisme d'annotation sémantique.

Dans la section suivante, nous présentons succinctement les mécanismes d'annotations sémantiques qui représentent une autre forme d'enrichissement sémantique.

3 Annotation sémantique

Le terme "annotation" est défini dans le Petit Robert comme une "note critique ou explicative qui accompagne un texte". Les annotations seules n'ont donc pas de sens, elles sont associées aux objets qui ont été annotés. Les annotations sont considérées comme des méta-

données particulières en ce sens qu'elles représentent de nouvelles données attachées à des ressources documentaires [Handschuh, 2005]. L'annotation n'est pas une fin en soi, mais elle a pour but d'être réutilisée ultérieurement. Plusieurs auteurs ont établi les objectifs précis liés aux annotations [Marshall, 1998][Mille, 2005] tels que classifier, hiérarchiser, reformuler, etc. Plus particulièrement, les annotations sémantiques ont pour objectif d'exprimer la "sémantique" à l'intérieur d'une ressource afin d'en améliorer sa compréhension, sa recherche et donc sa réutilisation par d'autres utilisateurs [Corcho, 2006].

Les annotations sont le plus souvent créées manuellement ou semi-automatiquement. Les traitements manuels consistent simplement à mettre en place une interface utilisateur dans laquelle l'utilisateur humain peut sélectionner la ressource à annoter, et, tout en respectant les contraintes imposées par le modèle formel, créer les annotations voulues sur la ressource sélectionnée [Kahan et al., 2001] [Handschuh et al., 2001].

Le traitement semi-automatisé pour la création des annotations sémantiques peut apporter une aide non négligeable à l'annotateur humain. Ces traitements semi-automatiques s'appuient généralement sur un moteur d'extraction d'informations selon deux méthodes : (1) création manuelle de patrons d'extraction [Vergas-Vera et al., 2002][Popov et al., 2003] ou (2) leur apprentissage supervisé [Handschuh et al., 2002][Dingli et al., 2003].

Enfin, il existe aussi des traitements entièrement automatisés mais ceux-ci utilisent des algorithmes basés sur des modèles statistiques ou sur l'exploitation de la redondance dans un corpus de ressources [CIMIANO et al., 2004][Dill et al., 2003]. Mais leur efficacité est limitée et leurs annotations sont souvent non exploitables dans des application concrètes.

Dans la figure 1.5, nous présentons un exemple d'annotation de documents réalisée dans le cadre du projet *e-Wok Hub* [e Wok Hub Consortium, 2008][8] qui a pour objectif l'annotation de documents issus de la géologie pétrolière. Dans cette exemple, nous montrons l'une des première étapes d'annotation qui a pour but la reconnaissance de la langue d'un document.

Le mécanisme d'annotation sémantique permet ainsi de lier une donnée à sa description sémantique représentée par un concept issu d'une ontologie. Nous utilisons ce mécanisme pour l'enrichissement sémantique des programmes et processus informatiques.

Un autre objectif de nos travaux concerne la persistance des programmes et processus informatiques afin d'assurer le passage à l'échelle. La section suivante présente la notion de base de données à base ontologique définie pour permettre la persistance en base de données des ontologies et des données qu'elles décrivent.

[8] http://www-sop.inria.fr/edelweiss/projects/ewok/

Table 1. Work-packages of the PICOREF project Storage capacity assessment Selection of favourable areas for CO2
injectivity Source to sink matching Guaranty of reservoir integrity and its cap rock Development of methods and
Improvement tools for modelling the short- and long- term behaviour of the system. Development of riskâ€" assessme
following methodology is applied (table 2): Table 2. Methodology (PICOREF project) head2right Description of
of phenomena involved head2right Risk assessment head2right Tools and methodology for control head2right Rem
checkbld Existing regulations checkbld Lack of regulations checkbld Documentation required by French adminis
PRELIMINARY RESULTS The characterisation of the reservoir and cap-rock geometry was based on the reproc
acquired during the last three decades. Specific processing was carried out so as to increase the quality of re
corrections were applied to correct the effect of the superficial Tertiary formations and the lateral
Chalk (13-layer model). Six regional transects were laid out and have allowed the construction of a geometric m
important that internal reservoir structures were identified (prograding wedges). Two targets are identif
trap associated with one of the major faults in the Paris basin (the Saint Martin de Bossenay fault) and
Annotation de la langue du document
et génération d'un fichier XML
contenant le document initial
</annotation>
<mediaUnit xmlns:ns4="http://weblab-project.org/core/model/text" xsi:type="ns4:text" uri="weblab://myrepository/d
<annotation uri="weblab://myrepository/doc_090320_163646#0-a0">
<data xmlns:ns5="http://weblab-project.org/core/model/" xmlns:ns6="http://weblab-project.org/core/model/t
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/element
<rdf:Description rdf:about="weblab://myrepository/doc_090320_163646#0">
<dc:language>en</dc:language>
</rdf:Description>
Annotation automatique
de la langue
</rdf:RDF>
</data>
</annotation>

<start>1882</start>
<end>1895</end>

<content>
Table 1. Work-packages of the PICOREF project Storage capacity assessment Selection of favourable areas for C
injectivity Source to sink matching Guaranty of reservoir integrity and its cap rock Development of methods a
injectivity Improvement tools for modelling the short- and long- term behaviour of the system. Development of
above, the following methodology is applied (table 2): Table 2. Methodology (PICOREF project) head2right
Characterisation of phenomena involved head2right Risk assessment head2right Tools and methodology for cont
of an accident checkbld Existing regulations checkbld Lack of regulations checkbld Documentation required
Responsibilities of each actor PRELIMINARY RESULTS The characterisation of the reservoir and cap-rock g
onshore seismic profiles (880 km) acquired during the last three decades. Specific processing was carrie
particular, the necessary static corrections were applied to correct the effect of the superficial T
of sonic speed in the Mesozoic Chalk (13-layer model). Six regional transects were laid out and have all
increase in quality was so important that internal reservoir structures were identified (prograding wedg

FIG. 1.5 – Annotation linguistique d'un document dans le projet *e-Wok Hub*.

4 Les Bases de Données à Base Ontologique (BDBO)

Une Base de Données à Base Ontologique (BDBO) [Alexaki et al., 2001b][Broekstra et al., 2002b][Park et al., 2007] est une base de données qui possède les trois caratéristiques suivantes :

- ontologie et données sont représentées dans la même base de données et peuvent faire l'objet des mêmes traitements (insertion, mise à jour, requêtes, etc.) ;
- toute donnée est associée à un élément ontologique qui en définit le sens et inversement, tout élément ontologique permet d'accéder aux données qui lui correspondent ;
- l'ontologie locale à la base de données possède éventuellement des références vers des ontologies externes.

La figure 1.6 représente synthétiquement l'architecture générale d'une BDBO. On retrouve les deux parties *catalogue système* et *données stockées*, présentes dans chaque base de données. On retrouve également, ce qui caractérise les BDBO, des ontologies, stockées au sein de la même base de données et définissant la sémantique des données.

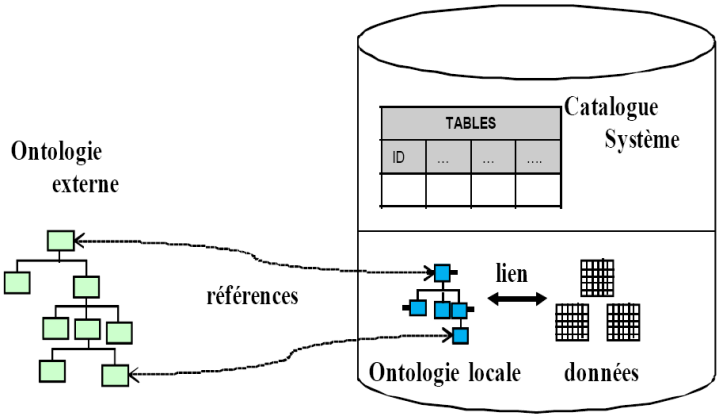


FIG. 1.6 – Architecture générale d'une BDBO.

De nombreuses BDBO ont été proposées dans la littérature telles que Sesame [Broekstra et al., 2002a], RDFSuite [Alexaki et al., 2001b], Jena [McBride, 2001][Wilkinson et al., 2003][Carroll et al., 2004], OntoDB [Dehainsala et al., 2007][Dehainsala, 2007], OntoMS [Park et al., 2007], DLDB [Pan et Heflin, 2003], RStar [Ma et al., 2004], KAON [Bozsak et al., 2002], 3Store [Harris et Gibbins, 2003a] et PARKA [Stoffel et al., 1997].

Une différence importante entre les BDBO est le schéma utilisé pour représenter ontologies et données. Dans ce qui suit, nous présentons une classification des architectures de BDBO en fonction du nombre de schémas utilisés [Fankam et al., 2008a].

4.1 BDBO de type 1

Dans les BDBO de type 1, un seul schéma composé d'une unique table de triplets (subject, predicate, object) est utilisé [Petrini et Sward, 2007][Chong et al., 2005][Harris et Gibbins,

2003b][Wilkinson et al., 2003]. Cette table, peut être utilisée à la fois pour les données de niveau ontologie et de niveau instance. Elle est appelée *table verticale*.

- pour les données de niveau ontologie, les trois colonnes de cette table représentent respectivement l'identifiant d'un élément d'ontologie, un prédicat et la valeur du prédicat qui peut être soit un identifiant d'un élément d'ontologie, soit une valeur littérale ;
- pour les données de niveau instance, les trois colonnes de cette table représentent respectivement l'identifiant d'une instance, une caractéristique d'une instance (c'est-à-dire une propriété ou l'appartenance à une classe) et la valeur de cette caractéristique.

La figure 1.7 illustre un exemple d'ontologie et ses instances sous la forme d'un graphe.

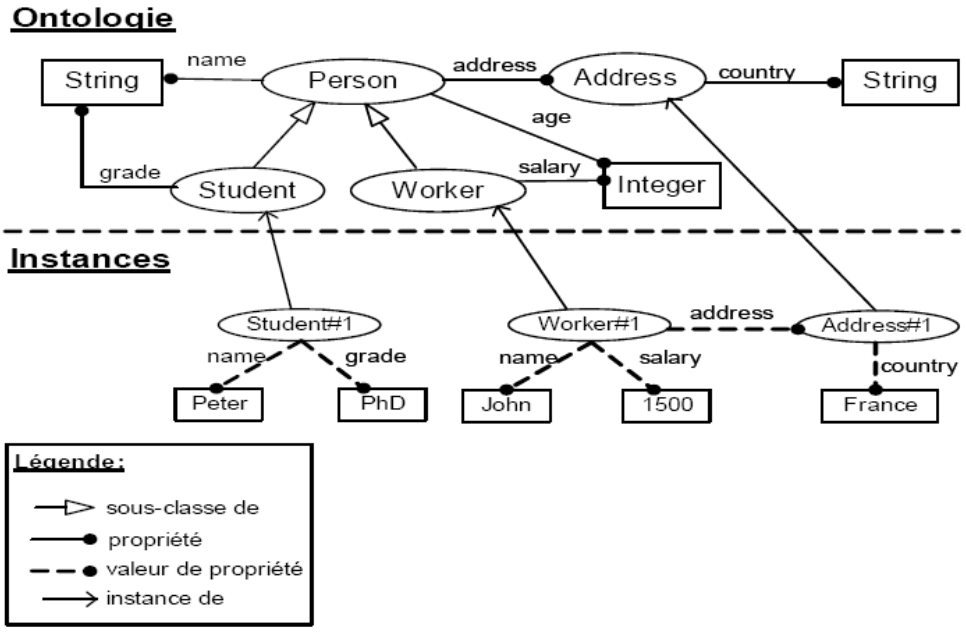


FIG. 1.7 – Un exemple d'ontologie et des instances de ses concepts.

La figure 1.8 montre le schéma de BDBO de type 1 correspondant à la figure 1.7. Une table verticale permet de stocker les classes et leurs relations. Dans cette figure, le triplet (Student, subClassOf, Person) représente une donnée de niveau ontologie : relation de subsomption entre les classes *Student* et *Person*. Le triplet (Peter, grade, PhD) représente une donnée de niveau instance : *Peter* a le grade de *Docteur*.

Ainsi, la table est utilisée pour stocker de la même façon l'instanciation des classes et la description des instances. Il n'y a donc pas de distinction entre le niveau ontologie et le niveau instance.

4.2 BDBO de type 2

Les BDBO de type 2 stockent les données de niveau ontologie et de niveau instance dans deux schémas différents [Alexaki et al., 2001a][Broekstra et al., 2002a][Ma et al., 2004]. Le

TRIPLES			
Subject	**Predicate**	**Object**	
Person	rdf:type	rdfs:Class	Ontologie
Student	rdf:type	rdfs:Class	
Student	rdfs:subClassOf	Person	
name	rdf:type	rdf:Property	
name	rdfs:range	xsd:String	
…	…	…	
Student#1	rdf:type	Student	Instances
Student#1	name	Peter	
Student#1	grade	PhD	
Worker#1	address	Address#1	
…	…	…	

FIG. 1.8 – Schéma de BDBO de type 1.

schéma pour les données de niveau ontologie dépend du modèle d'ontologie utilisé pour représenter les ontologies (par exemple : OWL ou PLIB). Il est composé de tables utilisées pour stocker chaque constructeur d'ontologies tel que les classes, les propriétés ou les relations de subsomption. Pour les données de niveau instance, une table verticale peut être utilisée pour stocker les données de niveau instance sous forme de triplets [Broekstra et al., 2002a][Ma et al., 2004]. Il est également possible d'utiliser une représentation binaire où chaque classe est représentée par une table unaire et chaque propriété par une table binaire [Pan et Heflin, 2003][Abadi et al., 2007]. Enfin, une troisième possibilité consiste à associer à chaque classe une table d'instances ayant une colonne pour chaque propriété associée à une valeur pour au moins une instance de la classe (représentation horizontale) [Dehainsala, 2007][Park et al., 2007].

La figure 1.9 montre le schéma de BDBO de type 2 utilisé pour le stockage des données correspondant à la figure 1.7. Dans cet exemple, les données de niveau ontologie sont stockées en utilisant un schéma pour des ontologies RDFS [Manola et Miller, 2004]. Dans la partie inférieure de cette figure, les données de niveau instance sont représentées en utilisant une représentation binaire. Toutefois, il est à noter que dans ce type de BDBO, la représentation du modèle d'ontologie est figée et ne peut donc pas évoluer.

4.3 BDBO de type 3

Les architectures de BDBO de type 3 étendent celles de type 2 en définissant un schéma supplémentaire appelé méta-shéma. Ce dernier est en particulier utilisé pour permettre l'évolution du schéma de représentation des ontologies, c'est-à-dire du modèle d'ontologie.

Le schéma d'OntoDB [Dehainsala, 2007] par exemple, est enrichi d'un méta-schéma qui conserve le modèle d'ontologies dans un méta-modèle refléxif. Il joue un rôle analogue à celui de la métabase (ou catalogue système) dans les bases de données traditionnelles, mais cette fois-ci pour les ontologies. En effet, le méta-schéma peut permettre : (1) un accès générique aux ontologies, (2) l'évolution du modèle d'ontologies utilisé, et (3) le stockage de différents

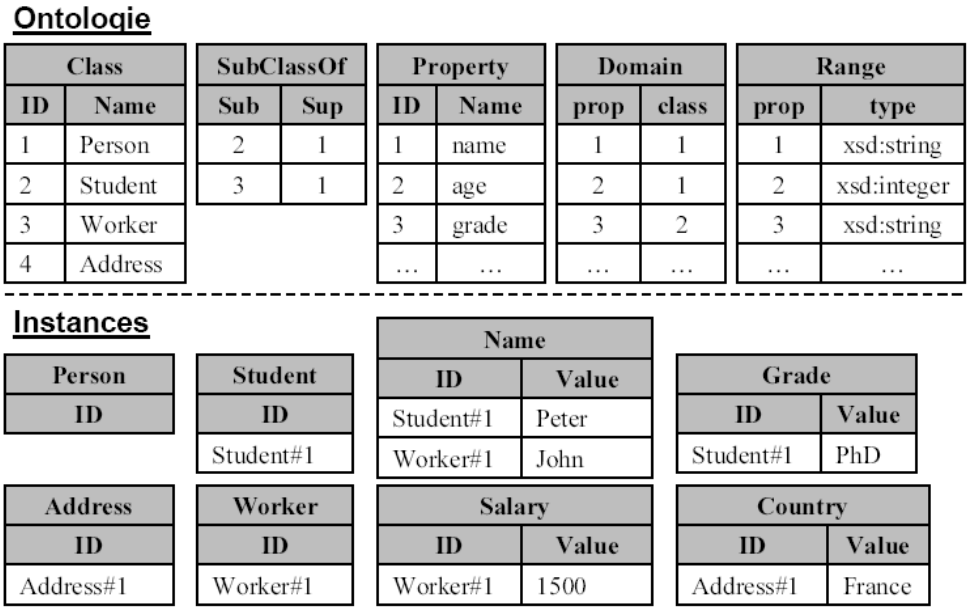


FIG. 1.9 – Schéma de BDBO de type 2.

modèles d'ontologies (OWL, PLIB, etc.). La figure 1.10 présente le méta-schéma correspondant à la figure 1.7. Il montre comment les constructeurs du modèle d'ontologies (constructeur de *Class*, de *Property* etc.) sont stockés au sein du méta-schéma.

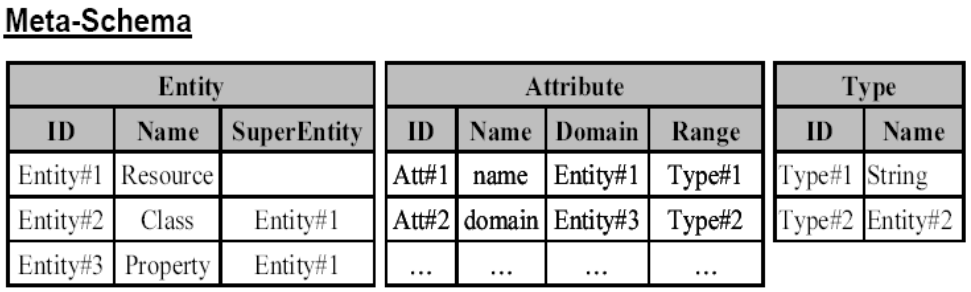


FIG. 1.10 – Schéma de BDBO de type 3.

Cette catégorie d'architecture de BDBO est donc celle qui permet d'avoir accès au modèle d'ontologie et de l'étendre. Ceci étant nécessaire pour représenter des programmes et processus informatiques et les annoter sémantiquement, nous avons choisi une BDBO de type 3 pour mettre en œuvre nos travaux. Plus précisément, nous avons choisi OntoDB qui a été développée et qui est disponible au sein de notre laboratoire.

5 OntoDB

OntoDB [Dehainsala et al., 2007] permet de représenter les données et les ontologies dans des schémas séparés tout en proposant une structure permettant d'accéder, de stocker et de faire évoluer le modèle d'ontologie utilisé. La représentation de cette structure, nommée méta-schéma, permet d'adapter OntoDB aux évolutions du modèle d'ontologie PLIB, mais aussi, lorsque le besoin s'en fait sentir, d'étendre le modèle d'ontologie utilisé par des constructeurs provenant d'autres modèles. En outre, OntoDB offre la possibilité d'interrogation au niveau ontologique grâce au langage de requête OntoQL [Jean et al., 2006a][Jean et al., 2007][Jean, 2007] que nous utilisons dans nos travaux et qui sera donc présenté par la suite. Ce langage est également utilisé pour créer et/ou manipuler les concepts du méta-modèle (modèle d'ontologie) en utilisant des instructions OntoQL.

La figure 1.11 présente les quatre parties qui composent l'architecture d'OntoDB :

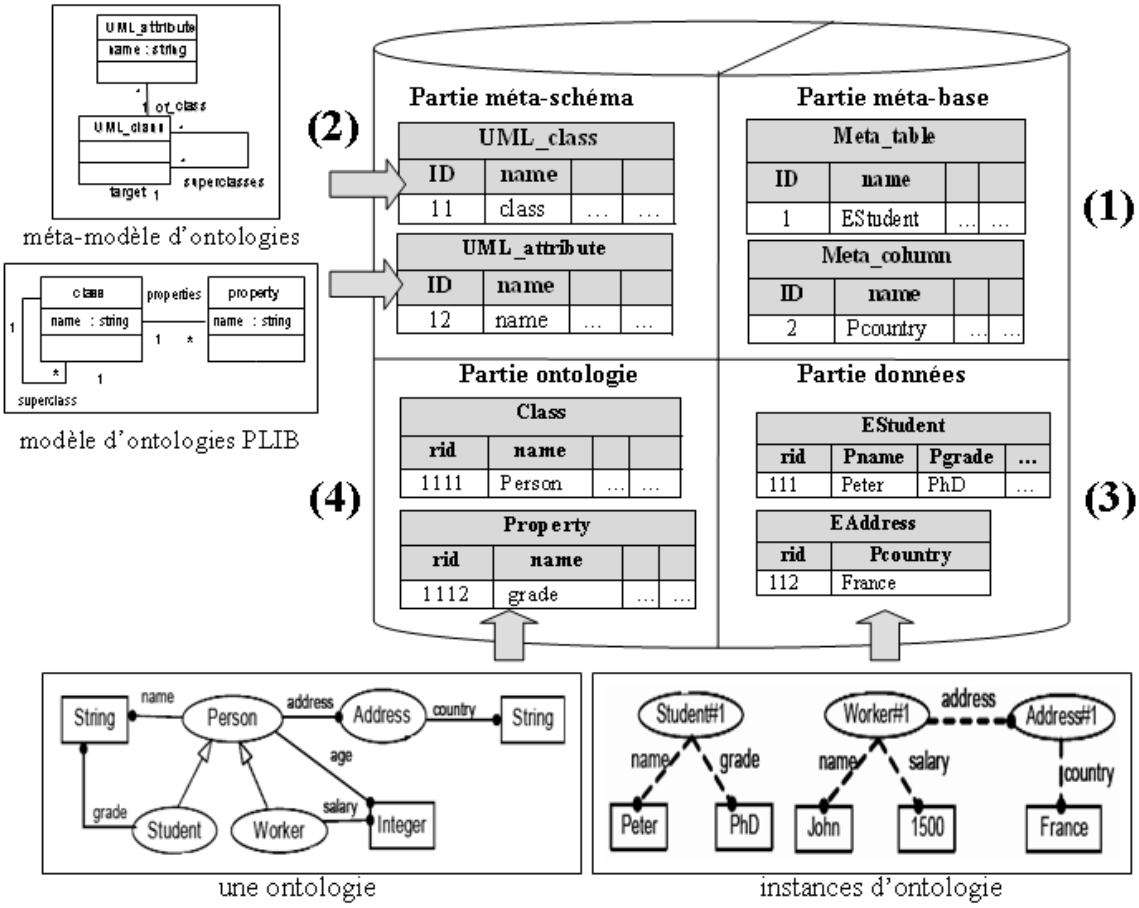


FIG. 1.11 – Architecture d'OntoDB.

1. **la partie *métabase* (1).** La *métabase*, souvent appelée *system catalog*, est une partie traditionnelle des bases de données classiques. Elle est constituée de l'ensemble des tables

systèmes. Ces tables sont celles dont le SGBD se sert pour gérer et assurer le fonctionnement de l'ensemble des données contenues dans la base de données. Dans OntoDB, elles contiennent en particulier la description de toutes les tables et colonnes définies dans les trois autres parties de cette architecture ;

2. **la partie *données* (3).** Elle représente les objets du domaine décrits en termes d'une classe d'appartenance et d'un ensemble de valeurs de propriétés applicables à cette classe. Ces objets sont représentés en associant une table à chaque classe (représentation horizontale) ;
3. **la partie *ontologie* (4).** Elle contient les ontologies définissant la sémantique des différents domaines couverts par la base de données. Ce prototype supporte le modèle d'ontologies PLIB ;
4. **la partie *méta-schéma* (2).** Elle représente, au sein d'un modèle réflexif, à la fois le modèle d'ontologies utilisé et le *méta-schéma* lui-même. Le *méta-schéma* est pour la partie *ontologie*, ce qu'est la *métabase* pour la partie données.

Ayant présenté les BDBO et particulièrement OntoDB qui est une BDBO de type 3, nous exposons, dans ce qui suit, OntoQL qui est un langage de requête pour BDBO.

6 OntoQL

Le langage OntoQL [Jean, 2007] est un langage de requête pour l'exploitation de BDBO de type 3 et implémenté sur OntoDB. Il permet ainsi de (1) définir, modifier et interroger des ontologies, (2) insérer, modifier et interroger des données dont le sens est défini par des ontologies et (3) étendre le modèle d'ontologies utilisé pour définir les ontologies.

Le langage OntoQL est basé sur le modèle noyau présenté précédemment (voir la figure 1.4). Ce modèle d'ontologie noyau n'est pas statique, il peut être étendu. En conséquence, les éléments de ce niveau de modélisation ne sont pas codés comme des mots-clés du langage OntoQL. Une convention est définie permettant de reconnaître dans la grammaire un élément du méta-modèle (modèle d'ontologie) indépendamment de son nom. Il s'agit de préfixer chaque élément de ce modèle par le caractère #. Ce préfixe permet de savoir que la définition de cet élément doit être insérée ou recherchée dans le niveau méta du système de modélisation à trois niveaux utilisé (modèle d'ontologies, ontologie et instance). De plus, le langage OntoQL permet de créer, modifier et supprimer les entités et attributs du méta-modèle en utilisant une syntaxe similaire à la manipulation de types utilisateur en SQL (CREATE, ALTER, DROP).

Dans ce qui suit, nous présentons quelques requêtes en OntoQL permettant l'exploitation d'OntoDB à différents niveaux[9]. Les mêmes types de requêtes seront utilisées pour mettre en œuvre notre approche.

[9]La syntaxe complète du langage OntoQL est disponible dans [Jean, 2007].

6.1 Exploitation des ontologies

OntoQL permet de créer, modifier, supprimer et rechercher dans les classes et les propriétés d'ontologies. Nous ne montrons dans cette section et dans les suivantes que les instructions qui nous seront utiles dans la suite de ce mémoire.

Requête. Créer la classe `Person` (voir figure 1.7) en indiquant que sa version est '001'.

```
CREATE #Class Person (
 DESCRIPTOR (
   #name[FR] = 'Personne',
   #version = '001')
 PROPERTIES (
   name STRING,
   age INT,
   address REF (Address)))
```

Explication. Cette instruction permet d'ajouter la classe `Person` en tant qu'instance de *#Class*. Elle possède deux valeurs d'attribut : un nom en français (attribut *#name[FR]*) et le numéro de version (attribut *#version*) du constructeur. Elle est créée avec trois propriétés. La propriété `name` représente le nom de `Person` et prend des valeurs de chaînes de caractères. La propriété `age` représente l'age de `Person` et prend des valeurs entières. La propriété `address` représente l'adresse de `Person` et prend des valeurs dans les instances de la classe `Address`.

Requête. Créer la classe `Student` (voir figure 1.7) en indiquant que sa version est '001'.

```
CREATE #Class Student UNDER Person (
 DESCRIPTOR (
   #name[FR] = 'Etudiant',
   #version = '001')
 PROPERTIES (
   grade STRING))
```

Explication. Cette instruction permet d'ajouter la classe `Student` en tant qu'instance de *#Class* et qui subsume la classe `Person`. Elle est créée avec la propriété `grade` qui représente le grade de `Student` et prend des valeurs de chaînes de caractères. Notons que cette classe hérite des propriétés de la classe `Person`, c'est-à-dire `name`, `age` et `address`.

Requête. Rechercher les classes dont la version est '003'.

```
SELECT #name
FROM #class
WHERE #version = '003'
```

Explication. Cette requête permet de rechercher toutes les classes qui ont été créées avec comme attribut `#version` la valeur '003'. Elle montre la similitude avec SQL. Ici, la seule différence est que cette requête recherche dans les ontologies (préfixe # utilisé) et non dans les tables classiques.

6.2 Exploitation des instances

OntoQL permet de créer, modifier, supprimer et rechercher dans les instances des classes d'ontologies créées.

Requête. Ajouter une instance à la classe `Student` : 'Peter' qui a un grade de 'PhD' (voir figure 1.7).

```
INSERT INTO Student (name, grade)
   VALUES ('Peter', 'Phd')
```

Explication. Cette instruction permet de créer une instance de la classe `Student` en valuant les propriétés `name` et `grade` à respectivement 'Peter' et 'PhD'. Contrairement à la section précédente, nous n'utilisons pas le préfixe # ce qui indique que nous travaillons au niveau instance. Notons que dans cet exemple, nous utilisons les noms anglais des classes et propriétés. Nous pourrions aussi utiliser les nom dans une autre langue (utilisation de la couche OL de l'ontologie).

Requête. Ajouter une instance à la classe `Worker` : 'John' qui a un salaire de '1500' unités et qui vit en 'France' (voir figure 1.7).

```
INSERT INTO Address (country)
   VALUES ('France')
INSERT INTO Worker (name, salary, address)
   VALUES
   ('John', 1500, (SELECT oid FROM Address WHERE country = 'France'))
```

Explication. La première instruction permet d'ajouter une adresse en ajoutant une instance à la classe `Address` qui est 'France'. La deuxième instruction permet d'ajouter une instance à la classe `Worker` en valuant les propriétés `name`, `salary` et `address` à respectivement 'John', 1500 et 'France'[10].

Requête. Rechercher les travailleurs qui ont un salaire inférieur à 1000 unités.

```
SELECT name
FROM Worker
WHERE salary < 1000
```

[10] Comme la propriété `address` est de type complexe, il faut rechercher la valeur de l'oid (identifiant unique) de l'instance de sa classe cible par une requête.

Explication. Cette requête permet de rechercher dans les instances de la classe `Worker` celles qui ont comme valeur de la propriété `salary` un entier inférieur à 1000.

6.3 Exploitation du Méta-modèle (modèle d'ontologies)

OntoQL est basé sur un modèle d'ontologie noyau qu'il est possible de modifier via le langage lui-même.

Requête. Ajouter le constructeur de restriction `AllValuesFrom` de OWL dans le modèle d'ontologie utilisé.

```
CREATE ENTITY #OWLAllValuesFrom UNDER #Class (
   #onProperty REF(#Property),
   #allValuesFrom REF(#Class) )
```

Explication. Cette instruction permet d'ajouter l'entité `OWLAllValuesFrom` au méta-modèle noyau comme sous-entité de l'entité `Class`. Cette entité est créée avec les deux attributs `onProperty` et `allValuesFrom` qui prennent respectivement comme valeur des identifiants de propriétés et des identifiants de classes.

6.4 Lien entre les niveaux

Nous avons montré des requêtes de niveau ontologie et des requêtes de niveau instance. Les deux types de requêtes peuvent être combinés. Ceci est utile, par exemple, pour retrouver les descriptions ontologiques de l'ensemble des classes auxquelles appartient une instance. Le lien entre ces deux niveaux se fait dans les deux sens.

Requête. Supposons que toutes les classes instances de `#OWLAllValuesFrom` aient la propriété `URI`, retrouver toutes les `URI` des instances de classes de `#OWLAllValuesFrom` ayant comme version '007'.

```
SELECT C.URI
FROM #OWLAllValuesFrom OVF, OVF AS C
WHERE OVF.#version = '007'
```

Explication. Cette exemple montre une requête faisant le lien à partir de l'ontologie vers les données. En effet, le constructeur **AS** permet de calculer les instances de classes déterminées dynamiquement. Ainsi, la requête précédente permet de retrouver toutes les classes instances de `#OWLAllValuesFrom` ayant comme version '007'. Ensuite, toutes les classes obtenues sont parcourus (`OVF AS C`) et leurs instances sont retrouvées.

Requête. Retrouver quelle classe instance de `#OWLAllValuesFrom` contient l'instance dont l'`URI` est *uri25*.

```
SELECT OVF.#name
FROM #OWLAllValuesFrom OVF, OWLAllValuesFromRootClass RC
WHERE TYPEOF(RC).#oid = OVF.#oid
 AND RC.URI = 'uri25'
```

Explication. Cet exemple montre une requête faisant le lien à partir des données vers l'ontologie. `OWLAllValuesFromRootClass` est la classe racine instance de `#OWLAllValuesFrom`, qui subsume toutes les autres. `TYPEOF(RC)` indique qu'on recherche la ou les classes de laquelle `RC` est instance sachant qu'elle a comme `URI` la valeur 'uri25'.

Nous présentons à présent les interfaces qui permettent d'accéder à OntoQL à travers des programmes JAVA que nous utilisons pour le développement de notre prototype logiciel.

6.5 Interface JAVA pour l'accès à OntoDB : JOBDBC et OntoAPI

L'API *JOBDBC* [Jean, 2007] permet d'exécuter des instructions OntoQL. Les principales interfaces de l'API JOBDBC sont `OntoQLSession`, `OntoQLStatement` et `OntoQLResultSet`. Elles sont présentées sur le diagramme UML de la figure 1.12. L'interface `OntoQLSession` encapsule une connexion à la base de données. La méthode `createOntoQLStatement` permet de créer une instance de l'interface `OntoQLStatement`. Cette interface permet d'exécuter une instruction OntoQL en proposant les méthodes `executeQuery` et `executeUpdate`. Le résultat d'une requête OntoQL est retourné sous la forme d'un objet de l'interface `OntoQLResultSet`. Cet objet peut être parcouru avec les méthodes `next`, `getInt`, `getString`, etc.

FIG. 1.12 – Extrait de l'interface JOBDBC.

Requête. Afficher le titre des messages.

```
OntoQLSession session = new OntoQLSessionImpl(connection);
String queryOntoQL = "SELECT titre FROM Message";
OntoQLStatement statement = session.createOntoQLStatement();
OntoQLResultSet resultSet = statement.executeQuery(queryOntoQL);
while (resultSet.next()) {
    System.out.println("Titre du message : " + resultSet.getString(1));
}
```

Explication. La première ligne de ce code source JAVA permet de créer un objet de type `OntoQLSession`. Pour exécuter la requête un objet `OntoQLStatement` est créé. La méthode `executeQuery` de cet objet permet d'exécuter la requête et retourne le résultat sous

la forme d'un objet `OntoQLResultSet`. Les lignes obtenues sont parcourues en réalisant une boucle appelant la méthode `next`. Pour chacune de ces lignes le titre du message courant est affiché en utilisant la méthode `getString`.

JOBDBC fournit les interfaces de base permettant d'exécuter une instruction OntoQL dans un programme JAVA. Cependant l'utilisation de cette API pour accéder aux ontologies requiert de connaitre le langage OntoQL. L'API *OntoAPI* a été conçu dans le but de permettre un accès similaire aux ontologies et aux données d'une BDBO. Elle est composée d'une interface JAVA pour chaque entité du modèle d'ontologies noyau d'OntoQL. Cette interface contient une méthode pour chaque attribut de cette entité. Un extrait de cette API est présenté sur la figure 1.13. OntoAPI contient les interfaces `OntoClass` et `OntoProperty`. Ces interfaces possèdent entre autres les méthodes `getOid` et `getName` pour obtenir la valeur des attributs correspondants. Lorsque de nouveaux éléments sont ajoutés au modèle noyau, cette API peut être re-générée pour les prendre en compte.

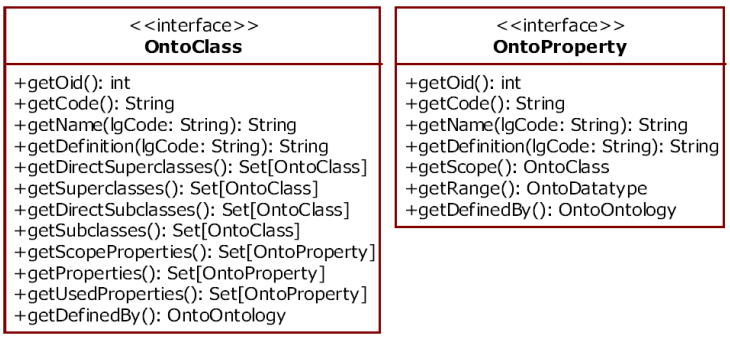


FIG. 1.13 – Extrait de l'interface JOBDBC.

Requête. Rechercher l'oid de la classe`Item`.

```
OntoQLSession session = new OntoQLSession(connection);
String queryOntoQL = "SELECT c.#oid FROM #Class c
                      WHERE c.#name[EN]='Item'";
OntoQLStatement statement = session.createOntoQLStatement();
OntoQLResultSet resultSet = statement.executeQuery(queryOntoQL);
System.out.println("L'oid de la classe Item est" + resultSet.getInt(1));
```

Explication. Les premières lignes de ce code source JAVA permettent de se connecter à la base de données et d'exécuter la requête. L'exécution de cette requête retourne la classe dont le nom en anglais est `Item`. L'objet `OntoQLResultSet` résultant contient une seule colonne. Enfin, l'oid est affiché en utilisant la méthode `getInt`.

7 Conclusion

Les ontologies étant des modèles permettant d'expliciter la sémantique d'un domaine d'étude, elles se posent comme une solution pertinente pour enrichir sémantiquement la description de programmes et de processus informatiques. Dans ce chapitre, nous avons proposé une étude de cette notion afin de définir et de déterminer précisément les ontologies et structures de persistance pertinentes pour notre approche. Après avoir donné une définition formelle d'une ontologie, nous avons vu que différents types d'ontologies existent qui peuvent s'articuler dans le modèle en oignon afin de bénéficier des avantages de chaque catégorie. Les ontologies utilisées dans notre approche suivent ce modèle. Par ailleurs, concernant le stockage en base de données des ontologies et de leurs instances, nous avons vu que seules les BDBO de type 3 permettent la modification du modèle d'ontologies utilisé. Ceci étant nécessaire dans nos travaux afin de représenter la sémantique des programmes et processus informatiques, nous exploitons la BDBO de type 3 OntoDB ainsi que son langage OntoQL que nous avons présentés dans ce chapitre.

Dans le prochain chapitre, nous présentons les aspects liés aux services, aux processus, à leur définition syntaxique et à leur enrichissement sémantique.

Chapitre 2

Les workflows en ingénierie

Sommaire

1 Introduction 35
2 Services Web 35
2.1 Web Service Description Language (WSDL) 36
2.2 Simple Object Access Protocol (SOAP) 37
2.3 Universal Description, Discovery and Integration (UDDI) 39
3 Workflows et compositions de services informatiques 40
3.1 Chorégraphie 41
3.2 Orchestration 41
3.3 Les langages de description syntaxique de workflows 42
3.3.1 Business Process Execution Language (BPEL) 44
3.3.2 XML Process Definition Language (XPDL) 45
3.4 Exécution de workflows 47
3.5 Insuffisances des descriptions syntaxiques de workflows 48
4 Enrichissement sémantique des services et workflows informatiques . 49
4.1 Langages de description sémantique de workflows 50
4.1.1 Web Ontology Language for Web Services (OWL-S) . 50
4.1.2 Web Service Modeling Ontology (WSMO) 52
4.2 Annotation sémantique de langages de workflows existants . . . 53
4.2.1 Semantic Annotation for WSDL (SAWSDL) 53
4.2.2 SEmantic Service MArkup (SESMA) 54
4.3 Workflows scientifiques et autres approches de gestion sémantique de workflows 55
4.3.1 WDO-It! 56
4.3.2 Pegasus 56
4.3.3 Kepler 56

4.3.4 Taverna 57
4.3.5 Triana 57
4.3.6 Vistrail 57
4.3.7 Askalon 58
4.4 Synthèse sur les descriptions sémantiques de workflows 58
5 Conclusion 59

Résumé. L'objectif de nos travaux est une meilleure gestion des programmes et processus informatiques. Dans ce chapitre, nous définissons précisément ces concepts et présentons certains travaux visant à leur meilleure gestion.

1 Introduction

Notre proposition a pour objectif une meilleure gestion des programmes et processus informatiques en se basant sur les notions d'ontologie et de base de données à base ontologique vues précédemment. Un programme informatique (que nous appelons également *service informatique*) correspond à une unité réalisant une fonction atomique. Des programmes informatiques sont par exemple des programmes exécutables, des fonctions de logiciels ou des services Web auxquels nous portons un intérêt particulier. Un processus informatique (ou *workflow informatique*) est défini quant à lui, comme un modèle informatique réalisant une fonction complexe décrite par un processus *abstrait*. Ainsi, un workflow informatique correspond à une composition de programmes informatiques et particulièrement à une composition de services Web. L'objectif de ce chapitre est de définir précisément ces notions afin de clarifier l'objet de nos travaux.

Le modèle informatique sur lequel repose un workflow informatique correspond à une description syntaxique utile pour l'exécution de celui-ci. En revanche, cette description ne comprend que peu d'éléments qui en définissent réellement le sens. Aussi, un intérêt de plus en plus important a été consacré à la description sémantique des workflows informatiques. Ce chapitre a pour second objectif de présenter certains travaux qui s'intéressent, comme nous le faisons, à la gestion sémantique des workflows informatiques.

Ce chapitre est organisé comme suit. Nous présentons les services Web dans la section 2, puis les workflows informatiques dans la section 3. Nous y exposons les concepts de chorégraphie et d'orchestration. Nous y décrivons également quelques langages de description syntaxique de workflows et mettrons en évidence leurs insuffisances. Dans la section 4, nous montrons les travaux majeurs effectués pour enrichir sémantiquement des descriptions de workflows. Nous finissons par une conclusion sur les langages de description syntaxique et sémantique de workflows en montrant le besoin d'une nouvelle approche.

2 Services Web

Les services Web sont des programmes informatiques permettant la communication entre applications distantes à travers le réseau Internet indépendamment de tout langage de programmation et de toute plate-forme d'exécution, sans intervention humaine, et en temps réel. Les acteurs participant à une communication de services Web sont :

- *le fournisseur du service* qui se charge de la description des messages manipulés et des profils des fonctionnalités offertes par le service. Il doit publier le service Web créé dans un annuaire afin qu'il puisse être trouvé par des clients ;
- *l'annuaire de services* fournit des informations sur la description et la localisation des services Web ;

– *le client* est une application qui cherche, localise et invoque le service.

La figure 2.1 décrit le scénario d'invocation d'un service Web.

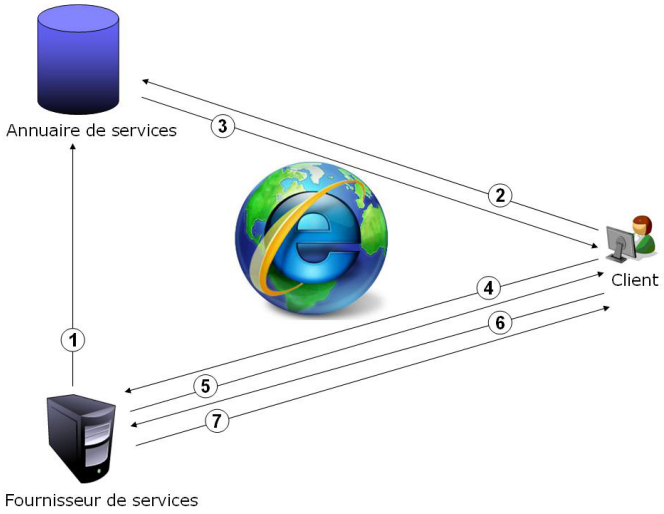


FIG. 2.1 – Scénario d'exécution d'un service Web.

Ainsi, le scénario se déroule comme suit. Un service est implémenté, décrit et publié par un fournisseur de services dans un annuaire de services (1). Un client interroge l'annuaire en utilisant les outils de recherche et de découverte fournis par l'annuaire afin de trouver un service qui implémente les fonctionnalités recherchées (2,3). Une fois un service trouvé, le client se connecte au fournisseur, obtient la description du service (4,5), et invoque les opérations du service (6). Enfin, le service répond aux requêtes envoyées par le client (7).

Les services Web s'appuient sur un ensemble de protocoles standardisant les modes d'invocation entre les acteurs participant à la communication. Il s'agit de SOAP jouant le rôle de protocole de communication, de WSDL permettant la description de services et d'UDDI jouant le rôle d'annuaire de services. Nous les décrivons succinctement dans ce qui suit.

2.1 Web Service Description Language (WSDL)

Le langage WSDL [W3C, 2004] est un standard W3C[11] basé sur une syntaxe XML. Il décrit l'interface publique du service Web en définissant l'adresse du service, son identité, les opérations qui peuvent être invoquées par les clients, les paramètres des opérations ainsi que leurs types. WSDL décrit les fonctionnalités offertes par un service Web par des opérations définissant un échange de messages. Un ensemble d'opérations est enveloppé dans un port défini comme un point final lié à un protocole de transport tel que SOAP, HTTP ou SMTP véhiculant les messages échangés entre le service Web et le client.

[11] WorldWideWeb Consortium (http://www.w3.org) : une communauté internationale qui élabore des protocoles et des standards pour assurer la croissance du Web à long terme.

Une description d'un service Web dans un document WSDL commence par l'élément racine *definitions*. Cet élément contient un attribut *name* définissant le nom du service Web décrit, et des *namespaces* pour définir les liens avec d'autres types de documents référencés. Six autres éléments sont définis à l'intérieur de l'élément *definitions* :

- *types* : définit les types de données (en XML-Schema [W3C, 2001]) dans la définition des messages ;
- *message* : définit les données échangées d'une manière abstraite indépendamment de l'implémentation et du protocole de transport ;
- *operation* : définit un échange de messages entre le service Web et le client, résultat de l'invocation d'une fonctionnalité offerte par le service Web ;
- *portType* : regroupe un ensemble d'opérations liées à un point final en vue de les associer à un protocole de transport ;
- *binding* : décrit les liens entre les messages d'une opération, décrite dans un *portType* donné, avec le protocole de transport utilisé ;
- *service* : définit des ports d'accès qui lient les *bindings* à l'adresse du service (une URL).

Le document WSDL de la figure 2.2, décrit un service Web nommé *PurchaseOrder*. Ce service définit un type de port nommé *purchaseOrderPT* possédant une opération nommée *sendPurchaseOrder*. Cette opération est représentée par un échange de deux messages *Input* et *Output* nommés respectivement *POMessage* et *InvMessage*. *POMessage* est composé de deux parties représentées par les types XSD nommés *customerInfoType* et *purchaseOrderType*. *InvMessage*, quand à lui, contient une seule partie de type XSD *InvoiceType*. Les types XSD sont définis dans un fichier importé dans la balise *types*. L'élément *binding* indique que le protocole de transport utilisé par le portType *purchaseOrderPT* est basé sur le standard SOAP. L'élément *service* définit l'URL où le serviceWeb *purchaseOrderService* peut être invoqué.

2.2 Simple Object Access Protocol (SOAP)

SOAP [W3C, 2000] est le standard adopté par le W3C pour la description et la transmission des messages entre deux points distants. Le transfert se fait le plus souvent à l'aide du protocole HTTP. SOAP utilise une syntaxe XML pour définir la structure des messages échangés par les services web. Les messages sont enveloppés dans des requêtes HTTP classiques (GET, POST). Cette requête comporte un en-tête et un corps contenant le message SOAP envoyé. Un message SOAP se compose des trois éléments suivants :

- l'enveloppe SOAP représente l'élément racine d'un message SOAP et est décrite par l'élément *enveloppe*. Elle contient un ensemble de *namespaces* pour une meilleure identification des éléments contenus dans le message SOAP. L'enveloppe contient l'en-tête et le corps du message SOAP ;
- l'entête SOAP considérée comme optionnelle, est décrite par l'élément *header*. L'entête SOAP permet de fournir des informations complémentaires par rapport aux informations contenues dans le message envoyé. Par exemple, il peut contenir des informations d'au-

```
<definitions name="PurchaseOrder" ...>
    <!- Déclaration des types XSD ------- ->
    <wsdl:types>
        <xsd:schema>
            <xsd:import namespace="http://manufacturing.org/xsd/purchase"
                        schemaLocation="http://manufacturing.org/xsd/purchase.xsd" />
        </xsd:schema>
    </wsdl:types>
    <!- Déclaration des messages ------- ->
    <wsdl:message name="POMessage">
        <wsdl:part name="customerInfo" type="sns:customerInfoType" />
        <wsdl:part name="purchaseOrder" type="sns:purchaseOrderType" />
    </wsdl:message>

    <wsdl:message name="InvMessage">
        <wsdl:part name="IVC" type="sns:InvoiceType" />
    </wsdl:message>

    ...
    <!- Déclaration des portTypes ------- ->
    <wsdl:portType name="purchaseOrderPT">
        <wsdl:operation name="sendPurchaseOrder">
            <wsdl:input message="pos:POMessage" />
            <wsdl:output message="pos:InvMessage" />
        </wsdl:operation>
    </wsdl:portType>

    ...
    <!- Déclaration des liens avec le protocole de transport ----- ->
    <binding name="purchaseOrderPTSoapBinding" type="tns:purchaseOrderPT">
        <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    </binding>
    <!- Déclaration de l'adresse du service ------- ->
    <service name="purchaseOrderService">
        <port name="purchaseOrderPort" binding="tns:purchaseOrderPTSoapBinding">
            <soap:address location="http://manufacturing.org/service/purchase"/>
        </port>
    </service>

</definitions>
```

FIG. 2.2 – Exemple de document WSDL pour le service *Purchase Order*.

thentification de l'expéditeur, d'identification de la session, d'identification de la transaction, etc ;

– le corps SOAP est décrit par l'élément body et représente les données contenues dans les messages échangés entre services.

La figure 2.3 est un exemple de message SOAP de demande de service pour le traitement d'un bon de commande par le service *Purchase Order*. Ce message provoque l'appel de l'opération *sendPurchaseOrder* décrite dans le document de la figure 2.2.

```
POST /StockQuote HTTP/1.1
Host: http://manufacturing.org/service/purchase
...
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Body>
      <m:sendPurchaseOrder xmlns:m="http://manufacturing.org/wsdl/purchase">
         <m:POMessage>...</m:POMessage>
      </m:sendPurchaseOrder>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

FIG. 2.3 – Requête SOAP.

La figure 2.4 est un exemple de message SOAP renvoyé par le service *PurchaseOrder* comme réponse à l'appel de l'opération *sendPurchaseOrder* décrite dans le document de la figure 2.2. La réponse représente une instance du message *InvMessage*.

```
<SOAP-ENV:Envelope
      xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/... >
   <SOAP-ENV:Body>
      <ns0:sendPurchaseOrderResponse
                xmlns:ns0="http://manufacturing.org/wsdl/purchase"
                SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <ns0:InvMessage>...</ns0:InvMessage>
      </ns0:sendPurchaseOrderResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

FIG. 2.4 – Réponse SOAP.

2.3 Universal Description, Discovery and Integration (UDDI)

A l'instar de toutes les ressources disponibles sur le Web, un service Web serait pratiquement impossible à localiser sans passer par des outils de recherche. Les annuaires de services Web représentent des bases de services où les fournisseurs peuvent publier et enregistrer des informations sur leurs services et leurs profils. Les annuaires des services Web peuvent être implémentés par des services Web accessibles par des applications informatiques. Ces annuaires fournissent des informations sur les services Web disponibles dans l'annuaire correspondant à la requête de recherche du client. UDDI [OASIS, 2004] est un standard OASIS[12] permettant de fournir une méthode de publication et de découverte d'informations sur les services Web. UDDI permet de publier et de rendre visible un service Web développé et fourni par une organisation en décrivant des informations sur la localisation de sa description et ses méthodes d'invocation.

[12]Organization for the Advancement of Structured Information Standards : http://www.oasis-open.org/.

La figure 2.5 montre un exemple d'une entrée UDDI décrivant le profil du service Web *PurchaseOrder*. Les éléments *accessPointURL* et *overviewURL* contiennent respectivement le lien vers l'implémentation et le lien vers la description WSDL du service Web *PurchaseOrder*.

```
<businessService serviceKey="" businessKey="00EB06FC-3C45-42B8-B394-1F7F35602B2E">
   <name>PurchaseOrderService</name>
   <descriptionxml:lang="en">...</description>

   <bindingTemplates>
      <bindingTemplate bindingKey="" serviceKey="">
         <description xml:lang="en">...</description>
         <accessPointURL Type="http">http://manufacturing.org:8080/services/PurchaseOrder</accessPoint>
         <tModelInstanceDetails>
            <tModelInstanceInfo tModelKey="UUID:0B6E1227-329A-4657-89B5-35DA36CA2554">
               <description xml:lang="en">...</description>
                  <instanceDetails>
                     <overviewDoc>
                        <overviewURL>http://manufacturing.org/wsdl/purchase.wsdl</overviewURL>
                     </overviewDoc>
                  </instanceDetails>
            </tModelInstanceInfo>
         </tModelInstanceDetails>
      </bindingTemplate>
   </bindingTemplates>
</businessService>
```

FIG. 2.5 – Entrée de l'annuaire UDDI.

Ayant présenté les services informatiques et particulièrement les services Web, nous nous intéressons, dans la prochaine section, à la description des workflows et composition de services informatiques qui sont également l'objet de nos travaux.

3 Workflows et compositions de services informatiques

Un workflow est la modélisation informatique d'un flux d'informations ou de processus au sein d'une organisation réalisée en utilisant des ressources afin d'accomplir des tâches spécifiées, comme par exemple la transmission automatique de documents entre des personnes [Osterweil, 1987][IEEE, 1990][Georgakopoulos et al., 1995][Wesk, 2007]. De façon plus pratique, le workflow décrit le circuit de validation, les tâches à accomplir entre les différents acteurs d'un processus, les délais, les modes de validation et fournit à chacun des acteurs les informations nécessaires pour la réalisation de sa tâche. La composition en workflows vise a créer des services de plus grande valeur ajoutée car réalisant des fonctions plus complexes. Pour un processus de publication en ligne par exemple, la modélisation des tâches de l'ensemble de la chaine éditoriale dans le but d'accélérer sa réalisation serait un workflow.

Les workflows sont particulièrement utilisés pour décrire comment des services ou activités "informatiques" sont connectés pour construire des procédés métiers complexes. Dans nos travaux, nous nous intéressons à ces workflows de composition de services exclusivement informatiques tels que les services Web. Nous décrivons ce type de workflow dans cette section.

Nous commençons par exposer les deux notions par lesquelles les compositions de services sont traditionnellement décrites, c'est-à-dire l'orchestration et la chorégraphie [Peltz, 2003]. Ensuite, nous décrivons les langages de description syntaxique et d'exécution de composition de services informatiques orientés orchestration en mettant en évidence leurs insuffisances.

3.1 Chorégraphie

La chorégraphie de services décrit, d'un point de vue global, une collaboration de l'ensemble des services participants, qui interagissent afin d'atteindre un but commun. Elle modélise la séquence des échanges de messages entre services composants et définit les conditions dans lesquelles ces messages sont échangés entre des clients, des fournisseurs et des partenaires. De plus, chaque partenaire impliqué dans le procédé décrit le rôle qu'il joue dans l'interaction. La chorégraphie offre ainsi une vision globale en s'intéressant à la description des interactions (collaborations) entre plusieurs partenaires.

Le figure 2.6 montre un schéma de chorégraphie de quatre services Web où chaque service communique avec deux autres. Par exemple, le service *SW1* envoie deux messages en parallèle aux services *SW2* et *SW4* (1). Lorsque les services *SW2* et *SW4* ont communiqué avec les autres services, ils répondent au service *SW1* en séquence suivant l'ordre *SW4* (6) suivi de *SW2* (7).

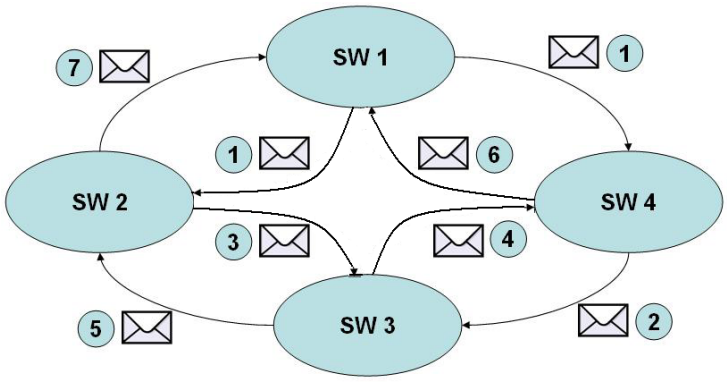


FIG. 2.6 – Une chorégraphie de services Web.

Parmi les langages qui permettent de représenter une chorégraphie de composition de services, WS-CDL (Web Services Choreography Description Language) [Kavantzas et al., 2005] est en cours de recommandation au W3C.

3.2 Orchestration

L'orchestration de services permet de définir l'enchaînement des exécutions des services participants. Elle décrit la manière dans laquelle les services peuvent interagir ensemble au

niveau des messages, incluant la logique métier et l'ordre d'exécution des interactions (ordonnanceur ou séquenceur). L'orchestration fait souvent référence à un processus exécutable qui interagit avec d'autres services dans le but d'accomplir les objectifs de l'orchestrateur.

La figure 2.7 montre un schéma d'orchestration de services Web correspondant au protocole de communication du service *SW1* avec les services *SW2* et *SW4* de la figure 2.6. L'exemple de l'orchestration montre que le service *SW1* envoie des messages en parallèle aux services *SW2* et *SW4*. Le service *SW1* se met en attente des réponses des services *SW4* et *SW2* en séquence.

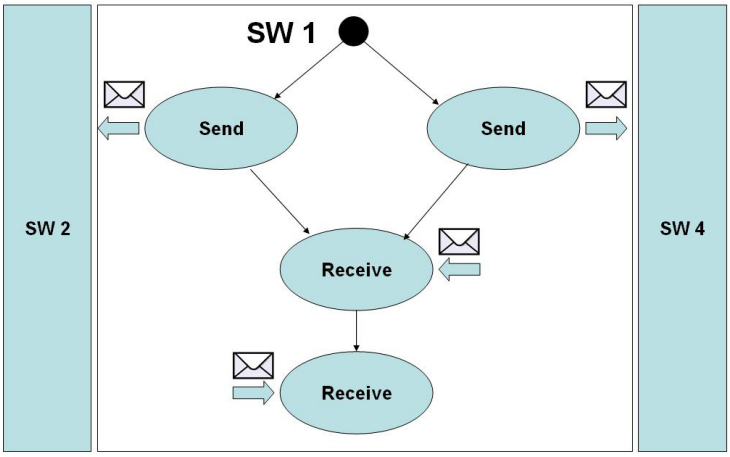


FIG. 2.7 – Une orchestration de services Web.

Une différence importante entre l'orchestration et la chorégraphie est que la chorégraphie offre une vision globale et plus collaborative de la coordination en décrivant le rôle que joue chaque participant impliqué dans l'application. En revanche, l'orchestration offre une vision locale et centralisée, c'est-à-dire que le procédé est toujours sous le contrôle d'un des partenaires métier représentant l'orchestrateur [Rojas, 2006]. Dans nos travaux, nous nous concentrons principalement sur la description de services qu'elle soit syntaxique ou sémantique dans une logique d'orchestration car cette logique est celle mise en place dans notre contexte applicatif. Nous présentons dans ce qui suit, quelques langages standards de description d'orchestration.

3.3 Les langages de description syntaxique de workflows

Depuis le début des années 2000, de nombreuses normes de description syntaxique pour la modélisation et l'exécution de workflows ont vu le jour et sont constamment enrichies de nouvelles propositions de différents fournisseurs. Un système de gestion de workflows (WfMS) est le dispositif logiciel permettant la création de modèles syntaxiques de workflows, leur stockage, leur exécution, leur administration et leur monitoring [Workflow Management Coalition, 1994][Wesk, 2007]. A l'heure actuelle plus de 250 Systèmes de Gestion de Workflow sont utilisés ou en développement :

– produits commerciaux : COSA, VisualWorkflow, Forté Conductor, Lotus Domino Workflow, Mobile, MQSeries/Workflow, Staffware, Verve Workflow, I-Flow, InConcert, Changengine, SAP R/3Workflow, Eastman, FLOWer, etc ;
– projets *open source* : JBoss JBPM, Bull Bonita, Active BPEL, Enhydra Shark-JaWE, etc ;
– prototypes du monde académique : Meteor, ADEPTFlex, OpenFlow, YAWL, APEL, JOpera, etc.

La plupart de ces systèmes utilisent leurs propres formalismes ou langages pour exprimer les modèles de workflow. Il existe néanmoins des organisations essayant de définir des standards au niveau des langages et des architectures de systèmes de workflow. La relation et l'évolution historique entre les différentes normes de modélisation de workflows sont illustrées sur la Figure 2.8.

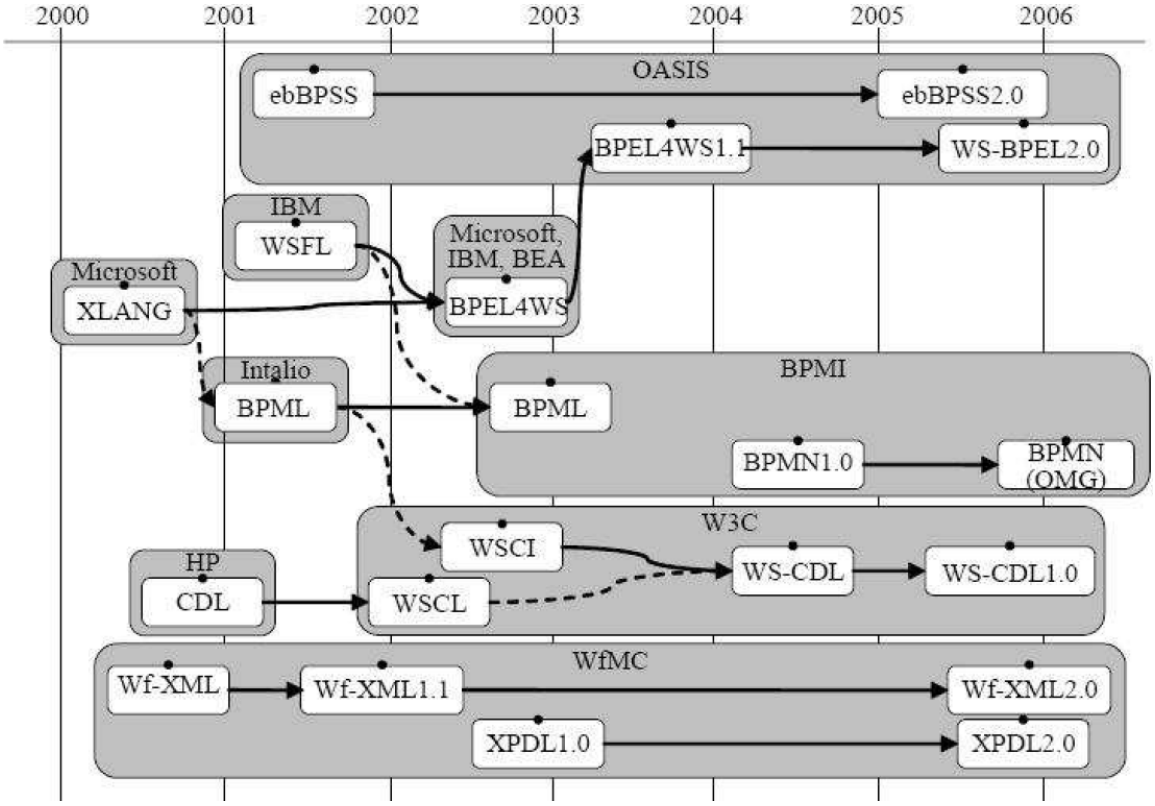


FIG. 2.8 – Généalogie des normes en description de workflows [Emmerich et al., 2008].

Parmi les organisations définissant des standards, nous trouvons, OASIS qui est chargée de la standardisation du langage d'orchestration de services Web BPEL. Une autre organisation est la WfMC qui spécifie le langage de modélisation de workflows XPDL visant à assurer l'interopérabilité des systèmes de workflows. Dans cette section, nous présentons ces deux formalismes de description syntaxique de workflows. Nous les avons choisis pour leur impact au niveau scientifique et industriel.

3.3.1 Business Process Execution Language (BPEL)

BPEL [Jordan et Evdemon, 2007] est un langage standard du consortium OASIS basé sur une syntaxe XML modélisant un processus métier par une composition d'un ensemble de services Web élémentaires. Le consortium OASIS est chargé de standardiser le modèle de composition BPEL qui est le résultat de la fusion de deux langages d'orchestration précurseurs, WSFL [Leymann, 2001] d'IBM et XLANG [Thatte, 2001] de Microsoft. Dans BPEL, chaque processus peut être considéré également comme un service Web. La spécification BPEL utilise des normes W3C : WSDL [W3C, 2004] pour la description des services Web partenaires, XML-Schema [W3C, 2001] pour la définition des structures de données et XPath [W3C, 1999] pour la récupération d'éléments XML. Les concepts les plus importants de BPEL sont :

- *PartnerLinks* : il fournit un canal de communication aux services Web distants utilisés dans le processus BPEL. Chaque *partnerLinks* est typé par un *partnerLinkType* et précise le rôle que joue un partenaire dans un *portType* ;
- *Variables* : elles sont utilisées pour stocker les messages d'interaction avec les services Web partenaires et les données internes du processus BPEL. Le type d'une variable référence un type de message dans un document WSDL ou un type de données défini dans un schéma XML ;
- *Simple activities* : les étapes de base d'un processus BPEL sont effectuées par les activités simples. Il y a des activités pour l'envoi et la réception de messages (*reply* et *receive*), l'invocation d'une opération d'un service Web partenaire (*invoke*), la modification du contenu d'une variable (*assign*), l'attente d'une date ou d'un délai (*wait*) ou la terminaison d'un processus BPEL (*exit*). La dernière version de BPEL a introduit une activité pour vérifier la conformité du contenu d'une variable à un schéma XML (*validate*) et l'ajout d'activités (*extensionActivity*) ;
- *Structured activities* : le contrôle de flux des activités simples est défini en utilisant les activités structurées. BPEL offre des activités pour spécifier l'exécution parallèle (*flow*), le branchement conditionnel (*if-then-else*) ou à la réception d'un message (*pick*), l'exécution séquentielle (*sequence*) et différentes itérations (*while*, *repeatUntil* et *forEach*). Les activités structurées peuvent être imbriquées. *Scope* est considéré comme une activité structurée particulière. Il définit un sous processus avec des variables locales et des gestionnaires d'erreurs et de compensation associés.

La figure 2.9 montre une description graphique du processus *Purchase_Order_Process* obtenu sur le plugin Eclipse BPEL Designer[13]. Sur réception de la commande d'un client (*Receive_Order*), le processus initie trois sous-processus en même temps : le calcul du prix final de la commande (*Compute_Price*), la sélection d'un expéditeur (*Arrange_Logistics*), et l'ordonnancement de la production et l'expédition de la commande (*Production_Scheduling*).

La figure 2.10 montre un fragment de description syntaxique en BPEL du processus *Purchase_Order_Process*. Cette description importe les définitions des services Web participants

[13]BPEL Designer Project : http://www.eclipse.org/bpel/

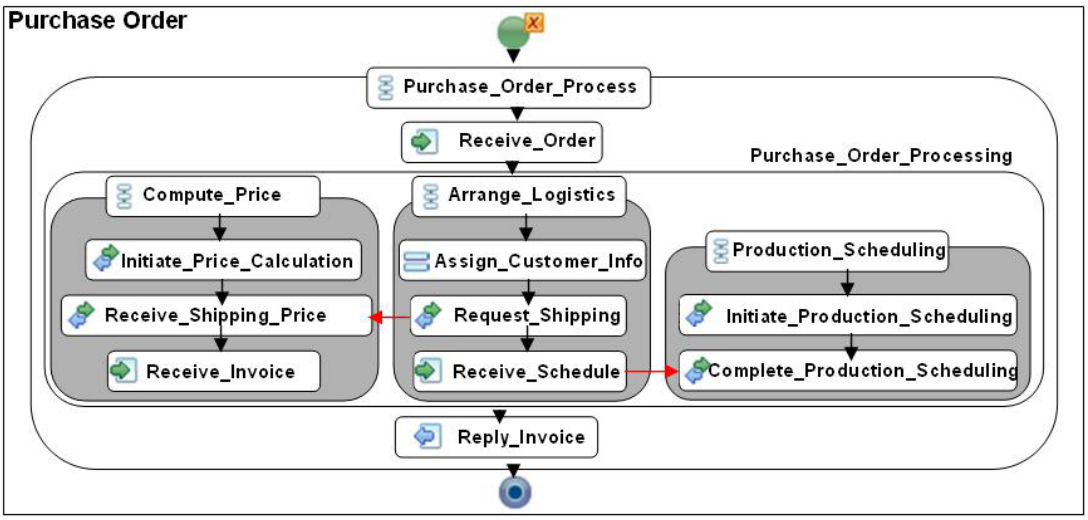


FIG. 2.9 – La description graphique du processus *Purchase Order*.

à travers leur fichier WSDL. Le comportement du processus est décrit par l'activité principale *Purchase_Order_Process* de type *sequence*. Les informations de la commande sont reçues dans le message *PO* par l'activité *Receive_Order* de type *receive*. Cette activité est lancée par l'appel de l'opération *sendPurchaseOrder* décrite dans le document WSDL référencé par le *namespace lns*. L'activité *Purchase_Order_Processing* de type *flow* lance le traitement de la commande par trois sous processus dont le premier est *Arrange_Logistics* qui est décrit par une activité de type *sequence*. Le processus *Arrange_Logistics* sélectionne un expéditeur à l'aide des activités *Assign_Customer_Info*, *Request_Shipping* et *Receive_Schedule* lancées en séquence.

3.3.2 XML Process Definition Language (XPDL)

XPDL [Workflow Management Coalition, 2002][Workflow Management Coalition, 2008] est un standard de la Workflow Management Coalition (WfMC) qui permet de définir un processus métier à l'aide du langage XML, processus métier qui sera ensuite utilisé par un moteur de workflow. Il a été conçu dans le but de servir de format d'échange entre les différents systèmes de gestion de workflows. Plus généralement, la WfMC promeut et développe l'utilisation de la technologie de workflow grâce à l'établissement de normes pour constituer une terminologie commune, améliorer l'interopérabilité et la connectivité entre les systèmes de gestion de workflow.

XPDL permet la définition d'un workflow basé sur une syntaxe XML comportant les principaux éléments suivants :

- les *processus* servent à définir des workflows ou des parties de workflow ;
- les *activités* sont des concepts de base dans XPDL. Il existe trois types d'activités, les activités de routage (*Route*) qui sont utilisées pour exprimer les contrôles de flux complexes

```
<process name="purchaseOrderProcess" ... xmlns:lns="http://manufacturing.org/wsdl/purchase">
...
<!- Déclaration des variables -------------- ->
<variables>
   <variable name="PO" messageType="lns:POMessage"/>
   <variable name="Invoice" messageType="lns:InvMessage"/>
   <variable name="shippingRequest" messageType="lns:shippingRequestMessage"/>
   <variable name="shippingInfo" messageType="lns:shippingInfoMessage"/>
   <variable name="shippingSchedule" messageType="lns:scheduleMessage"/>
</variables>
...
<!- Déclaration du gestionnaire d'erreur -------------- ->
...
<!- Déclaration du comportement du processus -------------- ->
<sequence name="Purchase_Order_Process">
   <receive name="Receive_Order" ...
            operation="sendPurchaseOrder" variable="PO"/>

   <flow name="Purchase_Order_Processing">

      <sequence name="Arrange_Logistics">
         <assign name="Assign_Customer_Info"...
               <copy><from>$PO.customerInfo</from><to>$shippingRequest.customerInfo</to></copy>
         </assign>
         <invoke name="Request_Shipping"... operation="requestShipping"
                inputVariable="shippingRequest" outputVariable="shippingInfo"/>
         <receive name="Receive_Schedule"...
                operation="sendSchedule" variable="shippingSchedule"/>
      </sequence>
...
```

FIG. 2.10 – Exemple de workflow BPEL.

entre des activités qui ne peuvent pas être exprimés seulement avec des transitions. Ensuite, les activités de bloc (*BlockActivity*) qui servent à exécuter un ensemble d'activités. Finalement, les activités de type implémentation (*Implementation*) qui servent à exprimer des tâches du workflow, ces tâches peuvent être implémentées en utilisant des procédures manuelles (réalisées par des êtres humains), automatiques (réalisées par une application) ou de type workflow (réalisées par un autre système de workflow) ;

- les *transitions* sont les relations entre les activités ;
- les *participants* décrivent les entités qui vont réaliser les tâches des activités.

XPDL inclut un mécanisme d'extension qui permet d'ajouter des attributs aux éléments qui font partie de la définition d'un workflow. Généralement ces attributs sont utilisés par les éditeurs pour inclure, par exemple, des informations de positionnement graphique des éléments. Le mécanisme d'extension est assez permissif, et donc la sémantique associée aux extensions est seulement comprise par les outils qui ont défini l'extension. Enfin, les informations graphiques permettent généralement la représentation sous forme d'un diagramme BPMN [White, 2004] qui est une norme de notation pour la modélisation de processus et dont l'objectif est de définir une notation graphique partagée par tous pour modéliser les processus métiers dans le but de les analyser.

La figure 2.11 montre une description graphique du processus *Approve_Purchase* obtenu sur

le logiciel Tibco [TIBCO, 2003]. Une condition sur le prix d'un produit est émise. Si le produit en question, est inférieur à 5000 unités alors il peut être approuvé par le manager. Sinon, il doit être approuvé par le directeur. Enfin, un email est généré automatiquement à la fin de la décision.

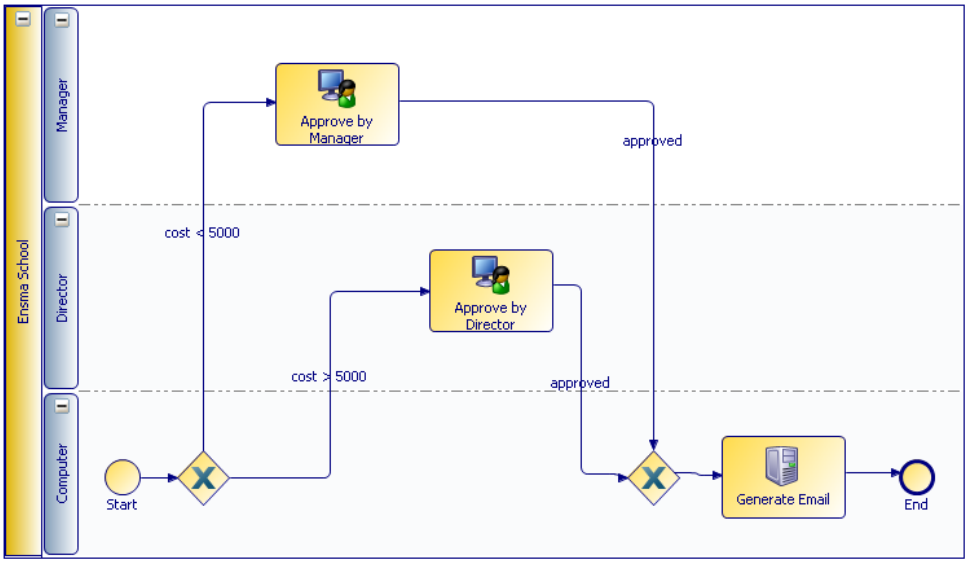


FIG. 2.11 – La description graphique du processus *Approve_Purchase*.

La figure 2.12 montre un fragment de description syntaxique en XPDL du processus *Approve_Purchase*. Cette description comporte des éléments relatifs à la description des activités (activités humaine, services Web, conditions, etc.), des transitions entre les activités, des informations graphiques, etc. Par exemple, l'activité *Generate Email* est décrite comme une implémentation de service Web.

3.4 Exécution de workflows

Décrire syntaxiquement un workflow informatique en termes de processus, d'acteurs, de données nécessaires et autres caractéristiques a comme objectif final et souvent unique de fournir toutes les informations suffisantes à son exécution. Le dispositif logiciel permettant l'exécution de workflows est appelé moteur de workflow. Ce dernier est lié à un langage de description syntaxique de workflows. Ainsi, les moteurs sont soit implémentés pour exécuter des descriptions standards de workflows ou bien pour exécuter des descriptions propriétaires. Par exemple, *Bonita* [Loridan et Rosell, 2006] est un moteur de workflows implémenté en Java et distribué en open-source. Il permet d'exécuter des instances de description de workflows en XPDL basé sur les standards de la WfMC. Un autre exemple est le moteur de workflow *Orchestra* [OrchestraTeam, 2010]. Il est également développé en Java sur les standards d'OASIS et permet d'exécuter des instances de descriptions de workflows BPEL.

```
<xpdl2:Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:simulation="http://www.tibc
"http://www.tibco.com/XPD/xpdExtension1.0.0" xmlns:xpdl2="http://www.wfmc.org/2004/XPDL2.0alpha" xsi:s
http://www.wfmc.org/standards/docs/TC-1025_bpmnxpdl_24.xsd" Id="_bsUXkBzfEeCITYFVvDH41g" Name="Process
  <xpdl2:PackageHeader>
  <xpdl2:RedefinableHeader PublicationStatus="UNDER_REVISION">
  <xpdl2:Pools>
  <xpdl2:WorkflowProcesses>
    <xpdl2:WorkflowProcess Id="_ce1ZoBzfEeCITYFVvDH41g" Name="Process">
      <xpdl2:ProcessHeader/>
      <xpdl2:RedefinableHeader/>
      <xpdl2:Activities>
        <xpdl2:Activity Id="_hUx04BzfEeCITYFVvDH41g" Name="Generate Email">
          <xpdl2:Implementation>
            <xpdl2:Task>
              <xpdl2:TaskService xpdExt:ImplementationType="WebService" Implementation="WebService">
            </xpdl2:Task>
          </xpdl2:Implementation>
          <xpdl2:NodeGraphicsInfos>
        </xpdl2:Activity>
        <xpdl2:Activity Id="_y2rlYBzfEeCITYFVvDH41g">
        <xpdl2:Activity Id="_zz7QwBzfEeCITYFVvDH41g">
        <xpdl2:Activity Id="_79xFsBzfEeCITYFVvDH41g" Name="Start">
        <xpdl2:Activity Id="_8kEQsBzfEeCITYFVvDH41g" Name="End">
        <xpdl2:Activity Id="_GQi9EBzgEeCITYFVvDH41g" Name="Approve by Manager">
        <xpdl2:Activity Id="_GtwdgBzgEeCITYFVvDH41g" Name="Approve by Director">
          <xpdl2:Implementation>
            <xpdl2:Task>
          </xpdl2:Implementation>
          <xpdl2:NodeGraphicsInfos>
        </xpdl2:Activity>
      </xpdl2:Activities>
      <xpdl2:Transitions>
    </xpdl2:WorkflowProcess>
  </xpdl2:WorkflowProcesses>
  <xpdl2:ExtendedAttributes>
</xpdl2:Package>
```

FIG. 2.12 – Exemple d'un workflow XPDL.

Des moteurs ont également été développés pour exécuter des descriptions propriétaires de workflows. *OpenFlow workflow engine*, développé par *IFP Énergies Nouvelles*, et commercialisé par *Beicip-Franlab* [Beicip-Franlab, 2010] fournit un éditeur de workflow permettant de définir un workflow métier à partir des activités disponibles dans un produit. Toutes les activités sont implémentées sous la forme de plugins de OpenFlow. OpenFlow fournit également un moteur pour exécuter les workflows et distribuer les activités sur différentes machines.

Dans ce qui suit, nous présentons les insuffisances des langages de description syntaxique de workflows orientés orchestration.

3.5 Insuffisances des descriptions syntaxiques de workflows

Les descriptions syntaxiques de workflows sont utiles à l'exécution et au monitoring de workflows. Les modèles de description syntaxique n'ont ainsi pas été initialement développés pour satisfaire aux exigences de la découverte, l'interopérabilité ou l'adaptativité des workflows. Pourtant, ces différentes exigences deviennent de plus en plus importantes de nos jours avec la

nécessité d'un faible couplage résultant du développement d'Internet et des Intranets. Au cœur des besoins se situe le besoin d'expliciter la sémantique des services Web et workflows.

De nombreuses approches ont été proposées pour prendre en compte les aspects liés à la sémantique et particulièrement aux fonctionnalités fournies par les services. Dans la section suivante, nous présentons quelques travaux émergents s'intéressant à l'enrichissement sémantique des services et des workflows informatiques.

4 Enrichissement sémantique des services et workflows informatiques

Les services Web peuvent être accédés et exécutés via le Web mais leurs descriptions ne sont que semi-formelles. Leur objectif final est de doter le Web avec des programmes distribués et en mesure d'interagir avec d'autres services, d'être récupérés automatiquement et composés en services plus complexes. En outre, l'automatisation de la découverte et de la composition est un élément clé de l'évolutivité, de la forte réactivité, de la réduction des coûts et de la qualité de service. Toutefois, le consensus sur les interactions mécaniques (informatiques) ne suffit pas pour permettre aux services Web d'interagir de façon non équivoque. De la sémantique explicite doit être injectée dans leur description. Une solution consiste à utiliser les technologies du Web sémantique [Berners-Lee et al., 2001][Fensel et Musen, 2001] (figure 2.13).

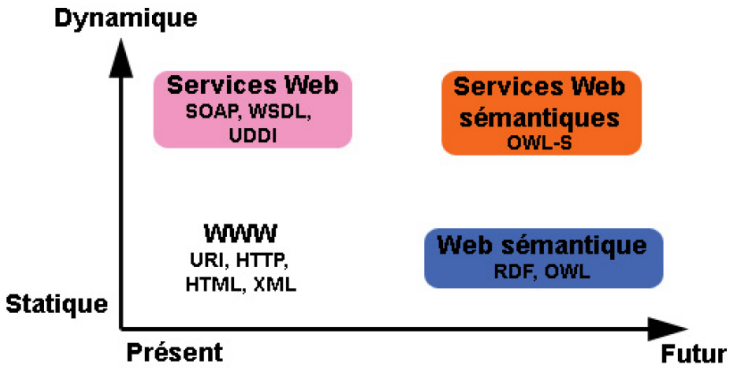


FIG. 2.13 – Services Web sémantiques.

De la même manière, l'un des facteurs qui empêche les experts des diverses disciplines scientifiques d'utiliser les technologies et infrastructures de gestion de workflows informatiques et particulièrement pour les exécuter, est la connaissance technique requise. Les détails concernant l'interopérabilité informatique, l'orchestration, le binding des méthodes abstraites aux services qui composent les workflows sont importants pour leur bonne exécution. Cependant, les experts ont d'autres besoins qui vont au-delà des considérations techniques tels que la découverte, l'interopérabilité sémantique, la composition sémantique etc. Pourtant, les systèmes de gestion de workflows informatiques n'ont pas été initialement développés pour satisfaire à ces

exigences sémantiques. En réponse à ces besoins, de nombreuses initiatives ont été mises en place pour enrichir sémantiquement les services et workflows informatiques et permettre ainsi la découverte, l'adaptativité ou l'interopérabilité au niveau sémantique de services Web et des workflows. Les ontologies dans ce domaine, en tant que conceptualisations formelles et consensuelles de domaines, et les annotations sémantiques permettent l'association d'une sémantique explicite à la description de services et de workflows informatiques. Dans cette section, nous présentons quelques travaux majeurs basés sur ces concepts.

4.1 Langages de description sémantique de workflows

4.1.1 Web Ontology Language for Web Services (OWL-S)

OWL-S [Martin et al., 2004] est un modèle d'ontologie basé sur OWL, permettant la description des propriétés et des capacités d'un workflow et d'une composition de services Web. L'objectif est de permettre à des utilisateurs et à des agents logiciels de découvrir, d'invoquer, de composer, et de contrôler des ressources Web offrant des services particuliers et ayant des propriétés particulières avec un degré élevé d'automatisation. OWL-S dérive d'OWL, ce qui signifie qu'elle est également construite en utilisant des ressources et des concepts hiérarchisés. De plus, OWL-S se base sur WSDL. En particulier, OWL-S utilise WSDL comme point de communication avec le service Web. Ainsi OWL-S ne connait pas les localisations physiques de services car celles-ci sont gérées par WSDL.

OWL-S propose également une façon de décrire le processus de composition de services. Grâce à ce processus, il est possible de déterminer si le service est un service composé de plusieurs autres ou si c'est un service atomique, c'est-à-dire qu'il n'a qu'une seule fonction. Il est à noter que de la perspective d'un utilisateur, même les services composés apparaitront comme un service unique.

OWL-S spécifie trois ontologies spécialisées appelées : *Profile*, *Model* et *Grounding* qui sont utilisées pour décrire différents aspects d'un service (figure 2.14) :

- l'ontologie *Profile* décrit les informations concernant les services offerts par un provider et qui vont être ensuite demandées par un client telles que l'entreprise qui fournit le service, la fonction calculée par le service et les caractéristiques du service ;
- l'ontologie *Model* représente le comportement d'un service. Cette ontologie contient une classe *Process* qui détermine l'utilisation du service comme un processus. Elle permet également d'indiquer les propriétés fonctionnelles, c'est-à-dire les entrées, les sorties, les préconditions et les effets. En ce qui concerne le type de processus, il existe trois categories : atomique, simple et composé. Une critique importante faite à OWL-S est qu'il n'impose aucune contrainte afin que l'ontologie *Profile* et l'ontologie *Model* soient cohérentes ;
- l'ontologie *Grounding* permet de déterminer comment accéder au service publié. Pour

cela, elle fournit les détails pour connecter la spécification abstraite à la spécification concrète.

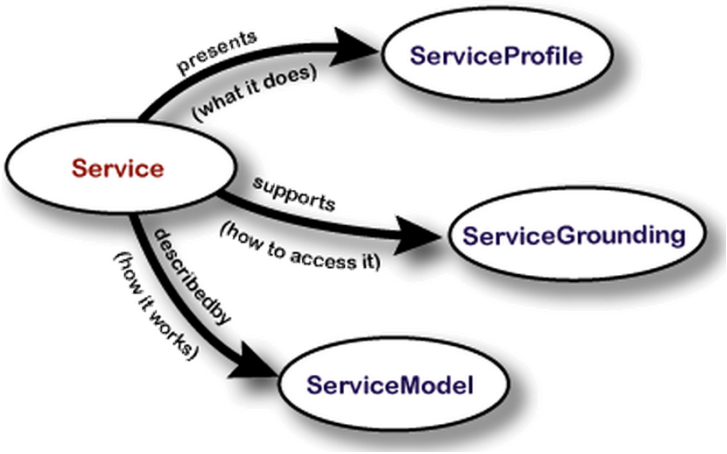


FIG. 2.14 – Modèle d'ontologie de services dans OWL-S.

Pour illustrer OWL-S, la figure 2.15 montre comment est spécifiée la description OWL-S d'un workflow *BravoAir_Process*. Il s'agit d'une séquence du processus atomique *GetDesiredFlightDetails* suivi du processus atomique *SelectAvailableFlight* suivi du processus composé *BookFlight*.

```
<process:CompositeProcess rdf:ID="BravoAir_Process">
 <rdfs:label>This is the top level process for BravoAir
                                          </rdfs:label>
  <process:composedOf>
    <process:Sequence>
      <process:components rdf:parseType="Collection">
        <process:AtomicProcess rdf:about=
                          "#GetDesiredFlightDetails"/>
        <process:AtomicProcess rdf:about=
                          "#SelectAvailableFlight"/>
        <process:CompositeProcess rdf:about=
                          "#BookFlight"/>
        </process:components>
    </process:Sequence>
  </process:composedOf>
</process:CompositeProcess>
```

FIG. 2.15 – Exemple de description OWL-S pour le workflow *BravoAir_Process*.

Un deuxième exemple est présenté dans la figure 2.16 pour représenter le processus atomique *LogIn*. Ses propriétés fonctionnelles sont spécifiées et peuvent être liées à des descriptions ontologiques.

Une des principales motivations de la création d'OWL-S, reste la possibilité de découvrir les services Web d'une manière automatique. En effet, un logiciel basé sur les agents interprète la description, et peut ensuite découvrir les entrées et les sorties demandées par le service. Une fois les entrées et les sorties découvertes, le service peut alors être exécuté automatiquement.

```
<process:AtomicProcess rdf:ID="LogIn">
  <process:hasInput rdf:resource="#AcctName_In"/>
  <process:hasInput rdf:resource="#Password_In"/>
</process:AtomicProcess>

<process:Input rdf:ID="AcctName_In">
  <process:parameterType rdf:resource=
                          "&concepts;#AcctName">
</process:Input>
<process:Input rdf:ID="Password_In">
  <process:parameterType rdf:resource=
                          "&concepts;#Password">
</process:Input>
```

FIG. 2.16 – Exemple de description OWL-S pour le service atomique *LogIn*.

Notons par contre qu'OWL-S ne permet pas de caractériser les services les uns par rapport aux autres de manière hiérarchique.

4.1.2 Web Service Modeling Ontology (WSMO)

WSMO [Bussler, 2004][Roman et al., 2005] est une ontologie qui décrit les différents aspects relatifs à la composition dynamique des services Web, y compris la découverte dynamique, la sélection, la médiation, l'invocation et l'interopérabilité. Elle est basée sur WSMF [Fensel et Bussler, 2002][Lara et al., 2004] (Web Service Modelling Framework) qui spécifie les éléments principaux pour décrire les services Web sémantiques.

WSMO a quatre composantes principales :

- *objectifs* : objectifs du client lors de la consultation d'un service Web. En d'autres termes, ils indiquent ce que l'utilisateur attend du service ;
- *ontologies* : description sémantique de l'information utilisée par tous les autres composants. Elles définissent la terminologie en termes de concepts, relations, fonctions, instances et axiomes ;
- *médiateurs* : connecteurs entre les composants avec facilités de médiation. Ils permettent l'interopérabilité entre les différentes ontologies ;
- *services Web* : description sémantique des services Web. Elle peut contenir des possiblités fonctionnelles offertes par le service et des descriptions d'interfaces d'utilisation.

Un autre objectif majeur de WSMO est d'offrir un découplage fort entre les composants et de jouer un rôle central pour la médiation. En effet, l'un des principes fondamentaux de WSMO consiste en la séparation totale entre les différents éléments impliqués dans la composition des services Web. A cet effet, WSMO distingue la formalisation de la demande du client de l'explicitation du service par son fournisseur. Sachant que la demande et l'offre sont décrites de façons différentes, un travail important doit être effectué afin d'établir la correspondance entre la demande et l'offre. Ce travail est le rôle des médiateurs sur lesquels l'architecture de WSMO

repose (figure 2.17).

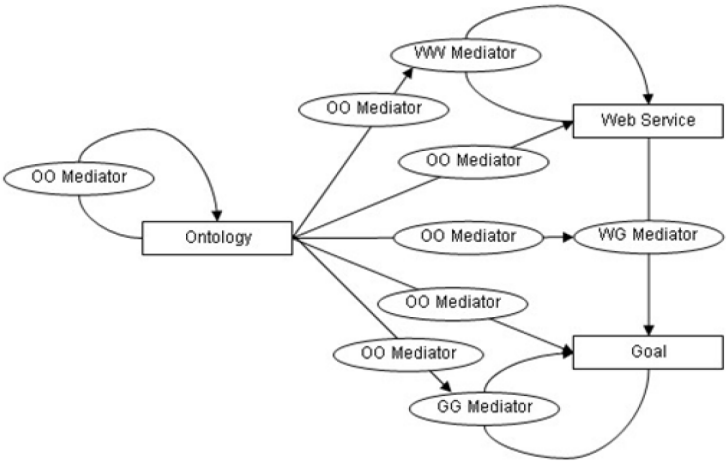


FIG. 2.17 – Architecture de WSMO.

Le langage WSML (Web service Modeling Language) [Bruijn et al., 2005] est utilisé pour décrire formellement tous les éléments définis dans l'architecture WSMO. D'autre part, l'environnement d'exécution WSMX (Web Service Execution Environment) [Bussler, 2004][Haller et al., 2005] qui offre un certain nombre de modules utilisables en run-time permettant la découverte, la sélection, la médiation et l'invocation des services Web sémantiques. Toutefois, les aspects relatifs à la description de compositions ne sont pas définis sémantiquement mais seulement au niveau de l'interopérabilité structurelle.

4.2 Annotation sémantique de langages de workflows existants

4.2.1 Semantic Annotation for WSDL (SAWSDL)

SAWSDL [Farrell et Lausen, 2007] est un langage d'annotation sémantique de description de services Web. Il est le successeur de WSDL-S (Web Service Description Language - Semantic) [Miller et al., 2004] qui permet d'ajouter des informations sémantiques sur une description WSDL y compris les définitions de condition, d'entrée, de sortie et les effets des opérations de service Web. Recommandée par le W3C, il s'agit d'une couche évolutive ajoutée au standard WSDL pour augmenter son expressivité en définissant sémantiquement les termes utilisés dans des descriptions WSDL et en référençant des ontologies RDF.

D'une part, SAWSDL fournit un mécanisme permettant d'annoter sémantiquement les types de données, les opérations, les entrées et les sorties de WSDL, et d'autre part, il ajoute des éléments pour spécifier les pré-conditions, les effets et les catégories des services Web. L'inconvénient de SAWSDL est la nécessité de modifier les outils existants autour de WSDL pour les adapter à SAWSDL même s'il a été conçu dans la perspective d'être un langage incrémental. Une autre limitation est qu'il n'aborde pas la composition de services.

Pour illustrer l'annotation sémantique, nous montrons dans la figure 2.18 la description WSDL d'une opération de *devis*. Cette opération a comme sortie un élément de type *devisDeclaration* qui représente un devis (ligne 2).

```
1 <wsdl:operation name="devis" >
2   <wsdl:output element="devisDeclaration" />
3 </wsdl:operation>
```

FIG. 2.18 – Opération devis de l'assuré en WSDL.

Dans la figure 2.19, nous présentons l'annotation du message produit en sortie par l'opération *devis* en l'associant au concept *DevisDeclaration* dans l'ontologie *AssuranceOnto* (lignes 3 à 4). Nous présentons également l'annotation de l'opération *devis* en l'associant au concept *Devis* dans l'ontologie *AssuranceOnto* (lignes 1 à 2).

```
1 <wsdl:operation name="devis"
2   psawsdl: preference="http://w3/ontology/AssuranceOnto/Operation#Devis"/>
3   <wsdl:output element="devisDeclaration"
4   psawsdl: preference="http://w3/ontology/AssuranceOnto#DevisDeclaration"/>
5 </wsdl:operation>
```

FIG. 2.19 – Description de l'opération devis de l'assuré en SAWSDL.

4.2.2 SEmantic Service MArkup (SESMA)

SESMA [Peer, 2005] est un langage de description de services Web sémantiques, qui a été conçu pour fournir un langage simple à utiliser, avec une syntaxe compacte et une intégration forte avec les langages existants WSDL, SOAP et WS-BPEL. SESMA a été construit avec les objectifs suivants :

- une syntaxe reposant sur le langage XML, pour une distribution du langage à large échelle ;
- une sémantique précise qui clarifie la signification des descriptions ;
- une forte complémentarité avec les langages existants ;
- des possibilités d'extension permettant une évolution vers d'autres langages éventuels.

SESMA fournit un support pour l'annotation de descriptions de workflows BPEL et peut précisément être utilisé pour :

- annoter des descriptions de services en se basant sur le WSDL (figure 2.20 (1)) ;
- annoter les descriptions WSDL qui décrivent les parties *Operation* et *Message* de descriptions BPEL (figure 2.20 (2)) ;
- annoter directement des fragments de description BPEL (figure 2.20 (3)).

Par exemple, la figure 2.21 illustre un fragment de description SESMA pour l'annotation sémantique de l'opération WSDL *buyItem*. Cette opération attend en entrée un produit, une carte

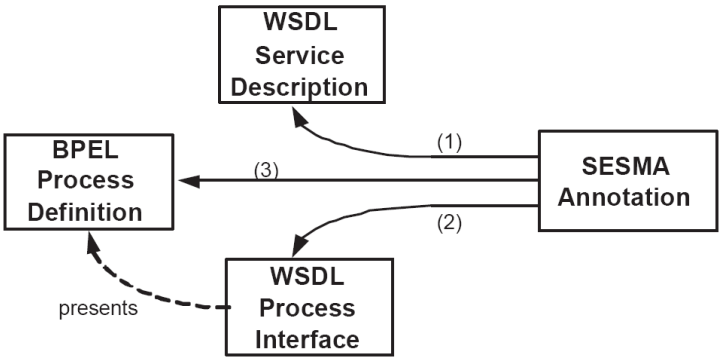


FIG. 2.20 – SESMA, WSDL et BPEL.

de crédit et une date d'expiration et comme résultat, rend au client le fournisseur du produit. Ainsi, nous montrons dans la figure 2.21 comment SESMA annote sémantiquement l'entrée de l'opération *buyItem*.

```
<annotation xmlns="http://schemata.org/sws/sesma"
xmlns:wsdl="http://schemata.org/sws/wsdl" xmlns:s="http://voc.org/preds#"
wsdl:url="http://sws.mcm.unisg.ch:8080/axis/services/MegashopService?wsdl"
wsdl:serviceName="MegashopServiceService">
    <functional-profile>
    <!-- buy an item -->
        <act-def wsdl:name="buyItem" wsdl:portType="MegashopService">
            <input>
                <var name="?item" wsdl:part="ean"/>
                <var name="?cc" wsdl:part="ccNr" />
                <var name="?ccexp" wsdl:part="ccExpDate" />
            </input>
...
```

FIG. 2.21 – Description SESMA pour l'annotation sémantique d'une opération WSDL.

SESMA modifie ainsi des descriptions standards pour les enrichir sémantiquement. Les outils existants doivent donc être modifiés pour pouvoir utiliser les nouvelles descriptions.

4.3 Workflows scientifiques et autres approches de gestion sémantique de workflows

D'autres systèmes de gestion de workflows permettent une meilleure gestion en se basant sur des techniques et des mécanismes d'enrichissement sémantique parmi lesquels WDO-It! [Pinheiro da Silva et al., 2007], Pegasus [Deelman et al., 2005], Kepler [Ludäscher et al., 2006], Taverna [Hull et al., 2006], Triana [Taylor et al., 2005], Vistrail [Scheidegger et al., 2008] et Askalon [Qin et Fahringer, 2008]. Nous les présentons dans cette sous-section.

4.3.1 WDO-It!

WDO-It! est un outil qui permet la construction de processus abstraits à travers des ontologies (WDOs workflow-driven ontologies). WDO-It! est d'abord utilisé pour capturer les connaissances et les encoder en ontologies de concepts et de méthodes (fonctions ou services). Ensuite, il est utilisé pour créer des processus abstraits. Cependant, aucun "rôle" précis n'est défini pour autoriser la modification de l'ontologie alors que seuls des experts, après consensualité, doivent être en mesure de faire évoluer l'ontologie. En outre, son but final ne concerne pas les aspects techniques de workflows exécutables. Il est plutôt de permettre aux scientifiques de créer des ontologies qui décrivent leur domaine de façon à ce qu'ils puissent générer un ensemble de processus abstraits dans ce domaine. Ainsi, leur travail ne concerne pas l'indexation de services ni de workflows pré-existants, ni la création de nouveaux workflows.

4.3.2 Pegasus

Pegasus est un framework qui permet de faire correspondre des processus abstraits à des workflows exécutables sur des environnements distribués. La construction de workflows scientifiques n'est pas le but principal. Pegasus fournit des fonctionnalités pour permettre aux utilisateurs de développer des applications indépendamment d'un système d'exécution donné. Des règles heuristiques sont utilisées par Pegasus pour lier les tâches spécifiées par le processus abstrait aux ressources de façon à minimiser le temps global d'exécution du workflow. En revanche, les processus abstraits que manipule Pegasus sont des *templates* de spécifications de workflow qui sont proches des spécifications techniques de workflows. La sémantique explicitée est donc faible. Pegasus s'appuie sur d'autres outils, tels que l'outil CAT (Composition Analysis Tool) [Kim et al., 2004] qui permet d'aider les scientifiques dans la conception de processus abstraits mais aucun ne représente une méthodologie stricte pour la construction de workflows partagés. Ces outils utilisent des bases de connaissances existantes mais ne sont pas conçus pour modifier les bases de connaissances en cours du processus de génération de workflow. De plus, l'objectif principal de CAT est d'assister la génération de workflows exécutables, il ne se soucie pas de la génération des processus abstraits. L'objectif est simplement de raffiner les processus abstraits jusqu'à l'obtention de workflows executables. En outre, il ne permet pas non plus l'indexation de workflows existants pour permettre de les réutiliser.

4.3.3 Kepler

Kepler est un système particulier de workflows guidé par des ontologies de domaine. Il fournit un environnement de développement dédié à la conception d'ontologie avec une partie navigateur/visualiseur sur celle-ci. Il permet ainsi à des utilisateurs de construire des workflows en utilisant une interface graphique pour programmer visuellement les spécifications de processus. Kepler génère et exécute des workflows exprimés dans le langage de modélisation MoML

[Lee et Neuendorffer, 2000]. Dans ce système, MoML qui est un langage simple, permet la spécification de workflows qui comprennent un ou plusieurs responsables *Director* qui planifient les tâches pour les *Actors* de l'organisation et ceci en s'appuyant sur une ontologie. Chaque *Actor* exécute une étape dans le workflow, et le *Director* donne la sémantique du déroulement du workflow. L'inconvénient est que les ressources qui sont utilisées en tant qu'acteurs dans la spécification de workflows nécessitent une inscription avec le système de Kepler et une formation de l'utilisateur en programmation de workflows afin d'être en mesure de produire des spécifications de workflow.

4.3.4 Taverna

Taverna est un système de gestion de workflows scientifiques. C'est la composante workflows du projet myGrid. Il a été conçu pour des utilisateurs qui n'ont pas nécessairement les connaissances techniques requises pour concevoir et exécuter des descriptions syntaxiques de workflows. Taverna supporte la découverte de services sémantiques à travers un plugin qui modélise les entrées et les sorties des opérations des services. La description sémantique des fonctions réalisées par les services se limite donc à la description des propriétés fonctionnelles. De plus, il n'y a pas de caractérisation hiérarchique entre ces fonctions. Enfin, il a été développé principalement pour le domaine de la bio-informatique et ne peut pas être appliqué à d'autres domaine où les ressources ne sont pas centralisées.

4.3.5 Triana

Triana est un système graphique open-source conçu pour définir, analyser, exécuter et gérer des workflows. Il permet à des utilisateurs de composer des workflows de manière graphique en renliant des "unités" par des "cables". Par exemple, la spécification d'un cable entre deux unités va mener à l'envoie d'un fichier d'une unité à l'autre. Triana supporte un grand nombres de modèles de données et de types de services tels que des code Java, des services Web et des langages de description syntaxique de workflows parmi lesquels WSFL et BPEL. Cela est notamment dû au fait qu'il soit utilisé dans plusieurs domaines. Toutefois, bien que ce système permette une grande flexibilité de composition, l'explicitation de la sémantique des services y est limitée.

4.3.6 Vistrail

Vistrail est un système qui a été développé pour visualiser et optimiser l'exécution de workflows. Il possède une infrastructure qui peut être combinée avec d'autre systèmes et librairies de visualisation car l'élément clé de VisTrail est la composition de manière simple et visuelle. De plus, il permet une séparation entre la spécification d'un "pipeline" qui est la composition

de "modules", et ses "instances" d'exécution. La sémantique des modules et des données n'est pas explicitée et ne peut pas être partagée de manière formelle.

4.3.7 Askalon

Askalon est un système de gestion de gestion workflows scientifiques. Il est basée sur une approche dont l'objectif est de séparer la sémantique des données de leur représentation informatique d'une part, et les fonctions réalisées des descriptions syntaxiques de services d'autre part. Ainsi, la composition de workflows se fait à travers les fonctions et non pas à travers leurs descriptions d'implémentation. D'autre part, la composition peut être établie de manière semi-automatique à l'aide des types de données d'une part, de leur sémantique d'autre part. Askalon permet aussi la conversion automatique entre certains types de données. Askalon peut être applique aux domaines de l'ingénierie dans lequel des ontologies peuvent être construites. Toutefois, la relation de subsomption implémentée n'est pas clairement explicitée et la construction de workflows à partir des ontologies décrites dans le domaine de la météorologie n'est pas formellement décrite.

Tout au long de cette section, nous avons présenté quelques propositions majeures d'enrichissement sémantique de services et de workflows informatiques. Dans ce qui suit, nous synthétisons ces approches afin de mettre en évidence leurs apports mais aussi leurs insuffisances.

4.4 Synthèse sur les descriptions sémantiques de workflows

La description sémantique de workflows vient enrichir la description syntaxique de workflows utile pour leur exécution. C'est une couche dont l'objectif est de permettre l'amélioration d'un ou plusieurs éléments liés à la gestion des descriptions syntaxiques de workflows tels que la recherche, l'interopérabilité ou l'adaptativité.

Plusieurs approches ont été proposées à cet effet. Nous les avons regroupées en deux catégories :

- les langages de description sémantique de services et de workflows tels que OWL-S [Martin et al., 2004] et WSMO [Bussler, 2004]. Ces langages permettent de proposer de nouveaux langages de description basés sur des standards. Ces nouveaux langages sont plus évolués et permettent une gestion sémantique des workflows (recherche sémantique à travers les données qu'ils manipulent, composition facilitée, etc.). Ces approches ne proposent par contre aucun moyen d'expliciter la sémantique qui existe au sein même des services et des workflows suivant leur fonctionnalité, leur précédence, etc. Elles ne proposent pas non plus de moyen de classer les services et les workflows au sein de catégories de services ;
- les annotateurs ou indexeurs sémantiques de descriptions de services et de composition de service tels que SAWSDL [Farrell et Lausen, 2007] et SESMA [Peer, 2005]. Ces systèmes

d'annotations permettent d'étendre des descriptions standards dans le but de définir sémantiquement certaines parties des descriptions syntaxiques de workflows. Les systèmes de gestion basés sur les standards sont plus facilement modifiables pour être adaptés aux descriptions ainsi annotées. Toutefois, ils ne proposent pas d'expliciter la sémantique au sein des services et des workflows d'un domaine ni de les caractériser hiérarchiquement les uns par rapport aux autres.

Il existe aussi des systèmes plus complets de gestion sémantique de services et de workflows tels que WDO-It! [Pinheiro da Silva et al., 2007] et Pegasus [Deelman et al., 2005]. Quelques travaux sur de tels systèmes de gestion permettent la caractérisation fonctionnelle des services mais s'y limitent dans le sens où ils n'indexent pas des services et des workflows pré-conçus. Ils n'ont pas non plus comme objectif l'exécution des workflows "abstraits" conçus. D'autres approches en revanche ont pour but de raffiner des workflows abstraits dans le but de construire des workflows exécutables. Ces approches ne semblent pas avoir une vue suffisamment complète pour intégrer toute la chaîne de gestion de workflows. D'autres propositions concernent l'enrichissement sémantique pour permettre la composition, la recherche ou l'adaptativité. [Mrissa, 2007] étudie ainsi la constitution de contextes pour décrire la sémantique des données échangées pour permettre la composition de services. [Benslimane et al., 2007][Maamar et al., 2009] décrivent des communautés de services comme des groupes rassemblant des services répondant à un même ensemble de besoins fonctionnels. Ainsi, une communauté peut être perçue comme offrant un niveau d'abstraction intermédiaire entre des applications clientes et les services qu'elle désirent utiliser. L'objectif principal du modèle de communautés est de fournir un cadre à la substitution d'un service par un autre. Cependant, les relations sémantiques entre les communautés ne sont ni structurées ni explicitées.

5 Conclusion

Les programmes ou services informatiques et particulièrement les services Web sont des fonctions atomiques exécutables localement ou à distance. Cependant, leurs descriptions syntaxiques ne permettent pas une gestion sémantique (découverte, interopérabilité, etc.). De même, leurs compositions en workflows informatiques représentées par des descriptions syntaxiques (BPEL, XPDL, etc.) utiles pour leur exécution ne permettent pas non plus leur gestion sémantique. De plus en plus de travaux s'intéressent donc à l'*enrichissement sémantique* des services et des workflows informatiques. Nous avons exposé quelques uns de ces travaux basés sur les ontologies et les annotations sémantiques (OWL-S, SAWSDL, etc.). Toutefois, la majorité des approches que nous avons étudiées présentent deux inconvénients majeurs :

- leurs mécanismes d'enrichissement sémantique procèdent soit par annotation sémantique de descriptions syntaxiques standards soit par extension de ces langages de description ;
- aucune classification hiérarchique ou relationnelle selon la sémantique des services et workflows informatiques n'est fournie.

Nous pensons que pour remédier à ces limitations, il faut permettre d'une part, une séparation claire entre les services et workflows informatiques et leur description sémantique et d'autre part, une hiérarchisation sémantique de ces services et workflows informatiques.

Dans le prochain chapitre, nous présentons le domaine de la modélisation géologique qui représente le contexte applicatif de nos travaux.

Chapitre 3

Modélisation géologique

Sommaire

1 Introduction 63
2 Géologie et objets géologiques simples 63
2.1 Interfaces sédimentaires 64
2.1.1 Horizon 64
2.1.2 Chenal 65
2.2 Contacts tectoniques 65
3 Modélisation géologique 66
3.1 Étapes de la modélisation géologique : plusieurs modèles géologiques 67
3.2 Interprétation sismique 68
3.2.1 Acquisition d'images sismiques 68
3.2.2 Marqueurs aux puits 70
3.2.3 Extraction d'objets d'intérêt 70
3.2.4 Construction de modèle structural 71
4 Gestion des processus en modélisation géologique 72
4.1 Processus d'interprétation sismique 72
4.2 Implémentation de processus en workflows informatiques 72
5 Conclusion 74

Résumé. Nous présentons dans ce chapitre le contexte de nos travaux de recherche, à savoir la modélisation géologique pour le stockage de CO_2. Ainsi, nous présentons quelques objets géologiques et les grandes étapes de la modélisation géologique. Dans cette description, nous prêtons un intérêt particulier à l'une de ces étapes : l'interprétation sismique. Nous nous appuyons ensuite sur ces notions pour montrer comment les processus de modélisation géologique sont actuellement gérés.

Ceci nous amènera, dans la prochaine section, à une synthèse générale de l'état de l'art où nous mettons en évidence les insuffisances des descriptions sémantiques des services et des workflows et les relions aux insuffisances liées à la gestion des processus de modélisation géologique.

1 Introduction

Dans ce chapitre, nous définissons le contexte dans lequel s'inscrivent nos travaux de recherche, à savoir la modélisation géologique pour le stockage de CO_2. Cette activité fait partie du domaine de la géologie pétrolière. La modélisation géologique pour le stockage de CO_2 permet de représenter de manière plus ou moins fidèle une zone de prospection pétrolière. Un modèle est ainsi créé et subit des simulations d'écoulement de fluides et particulièrement de CO_2. Ceci permet à terme un stockage efficace et sûr du CO_2. Cette activité est supportée par un ensemble de processus de modélisation géologique implémentés en programmes informatiques et composés en workflows informatiques. La description de ces processus nous permettra de montrer les insuffisances actuelles dans leur gestion. Répondre à ces insuffisances et appliquer notre proposition au contexte particulier de la modélisation géologique est l'objet de nos travaux.

Ce chapitre est organisé comme suit. Dans la section 2, nous présentons les principales notions de la géologie nécessaires dans la modélisation géologique. Nous y introduisons les termes utilisés dans le domaine et particulièrement les objets géologiques simples tels que les *horizons* ou les *failles*. Nous présentons dans la section 3, la modélisation géologique et particulièrement l'interprétation sismique en tant que domaine d'application de notre étude. Ensuite, nous exposons dans la section 4, un processus potentiel d'interprétation sismique et son implémentation en workflow avant de finir par une conclusion.

2 Géologie et objets géologiques simples

La géologie est la science qui traite de la composition, de la structure, de l'histoire et de l'évolution des couches internes et externes de la Terre, et des processus qui la façonnent. La géologie est une discipline importante parmi les sciences de la Terre et possède de nombreuses attaches avec d'autres sciences, parmi lesquelles la géochimie, la géophysique, la minéralogie, la paléontologie, la sismologie, etc. Elle intervient, en même temps que la géophysique appliquée, dans la recherche et/ou l'exploitation des ressources naturelles notamment le charbon, les minerais, les pierres précieuses, l'eau et les hydrocarbures.

Le domaine qui nous intéresse particulièrement est la représentation de la structure du sous-sol en vue d'y stocker du CO_2. Nous décrivons ici les propriétés des principaux objets géologiques ainsi que les relations entre ces objets qui nous permettent d'introduire le vocabulaire nécessaire à la bonne compréhension de l'application de cette thèse.

Divers types de modèles géologiques 3D sont utilisés pour les besoins de l'exploration pétrolière. Les plus simples de ces modèles, dont dérivent tous les autres, sont les modèles structuraux, qui représentent l'assemblage des différentes surfaces géologiques présentes dans une zone d'intérêt déterminée. Elles sont classées selon 2 grandes familles que sont les *interfaces*

sédimentaires et les *contacts tectoniques*.

2.1 Interfaces sédimentaires

Les *interfaces sédimentaires* séparent deux strates sédimentaires, c'est-à-dire deux couches du sous-sol déposées l'une au dessus de l'autre et chacune composée de roches du même type. Chaque interface sédimentaire correspond à un évènement géologique marquant comme une érosion, l'achèvement d'un épisode de dépôt ou encore le déplacement d'une aire de dépôt.

2.1.1 Horizon

Dans le domaine de la géologie pétrolière, les interfaces sédimentaires sont généralement désignées comme des *horizons* [Michum et al., 1977][Catuneanu, 2002]. Ces surfaces séparent des couches sédimentaires, qui sont regroupées, à plus grande échelle, en unités sédimentaires (encore appelées unités stratigraphiques). Les horizons sont généralement de grande taille (plusieurs kilomètres carrés) et sont souvent horizontaux ou faiblement inclinés. Du fait de leur plus ou moins grande horizontalité, les horizons sont dans la plupart des cas de bons *réflecteurs*[14] des ondes acoustiques. Pour cette raison, ils sont donc généralement identifiables de manière aisée sur les images sismiques. Un horizon peut être continu sur l'ensemble de sa surface. A l'inverse, du fait des accidents tectoniques qui l'affectent, il peut éventuellement être découpé en plusieurs morceaux.

La figure 3.1 schématise des horizons qui sont généralement disposés parallèlement les uns aux autres mais sous l'effet de divers facteurs liés aux particularités des environnements de dépôt ou à la tectonique locale, il arrive que cette propriété ne soit pas respectée.

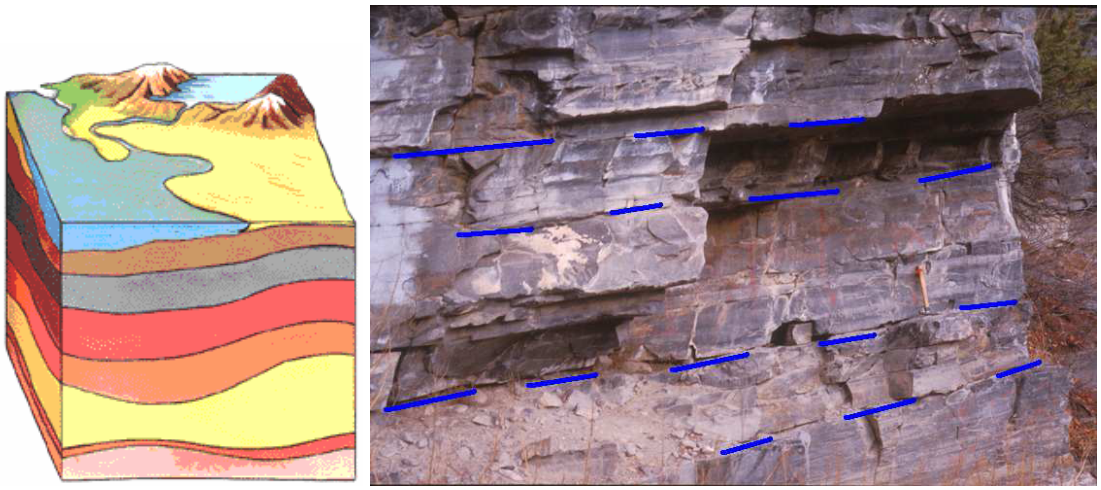


FIG. 3.1 – Exemples de successions d'horizons.

[14]Les réflecteurs sont des surfaces épaisses visibles sur l'image sismique.

2.1.2 Chenal

Un *chenal* est une dépression de forme généralement concave et linéaire dans laquelle les sédiments transportés par un flux aqueux constituent divers corps sédimentaires allongés [Schlumberger Oilfield Glossary, 2009]. De manière plus imagée, il s'agit du lit d'une ancienne rivière asséchée et remplie par des dépôts ultérieurs.

La figure 3.2 montre une représentation réelle de chenal.

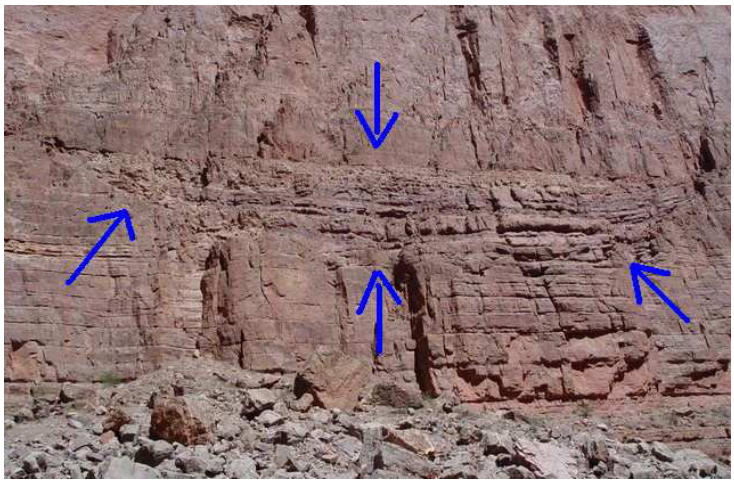


FIG. 3.2 – Exemple réel d'un chenal.

Un chenal a un profil transversal concave et un profil longitudinal toujours allongé qui peut être de type droit, méandreux ou tressé. Par ailleurs, un chenal peut s'étendre sur plusieurs centaines de mètres de largeur et sur plusieurs kilomètres de longueur.

2.2 Contacts tectoniques

Ils sont générés par les mouvements transversaux qui affectent la croûte terrestre et dont la cause première est le jeu des plaques tectoniques. Ces contacts correspondent à des surfaces de fractures le long desquelles les masses rocheuses glissent de manière différentielle sous l'effet du champ de contraintes locales auxquelles elles sont soumises.

Les contacts tectoniques appartiennent principalement à la catégorie des *failles*. Selon [Groshong, 2006], une faille est une discontinuité qui sépare les couches qu'elle affecte en deux compartiments qui ont subi un déplacement appréciable l'un par rapport à l'autre (voir figure 3.3). Les failles peuvent avoir des extensions longitudinales extrêmement variables, allant du mètre à la dizaine de kilomètres ou plus.

Il existe principalement 3 types de déplacements des panneaux le long d'une faille :

– normal : le déplacement est vertical, le panneau supérieur est déplacé vers le bas et la pente de la faille est d'environ 60 degrés par rapport à l'horizontal ;

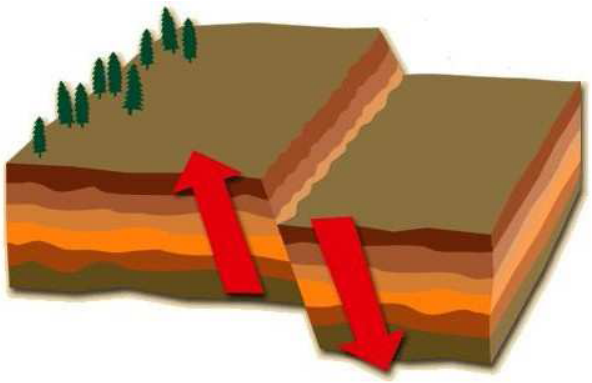


FIG. 3.3 – Une faille commune.

– inverse : le déplacement est vertical, le panneau supérieur est déplacé vers le haut et la pente de la faille est d'environ 45 degrés par rapport à l'horizontal ;
– décrochement : le déplacement est horizontal, la pente de la faille est d'environ 90 degrés par rapport à l'horizontal.

Une faille n'est généralement pas directement visible sur une image sismique. En effet, dans la mesure où une faille n'est pas une surface homogène du point de vue des propriétés physiques, son comportement n'est pas homogène du point de vue de la propagation des ondes sismiques.

Ayant défini certains concepts géologiques, nous présentons dans la section suivante la modélisation géologique qui fait appel à ces concepts.

3 Modélisation géologique

La modélisation géologique s'intéresse à la représentation du sous-sol sur ordinateur. Elle date des années 80 et a été rendue possible grâce aux progrès de la Conception Assistée par Ordinateur (CAO) et de l'informatique graphique.

La modélisation géologique est particulièrement utilisée dans le domaine de l'exploration pétrolière où le nombre de données et les volumineux calculs effectués nécessitent l'emploi d'ordinateurs. Dans ce cas, les modèles géologiques utilisés (*earth models*) sont de plusieurs types. Ils peuvent concerner soit des gisements, et donc des réservoirs individuels (modèles de réservoirs) soit des bassins dans leur ensemble (modèles de bassin).

Dans ce qui suit, nous présentons les étapes de la modélisation géologique et présentons plus en détail l'interprétation sismique qui représente la première grande étape de cette modélisation.

3.1 Étapes de la modélisation géologique : plusieurs modèles géologiques

Lors de la chaîne de traitements de l'exploration pétrolière (en vue d'extraire des hydrocarbures ou de stocker du CO_2), plusieurs *earth models* sont utilisés. La figure 3.4 représente une vue d'ensemble du processus global dédié à la modélisation géologique d'un gisement.

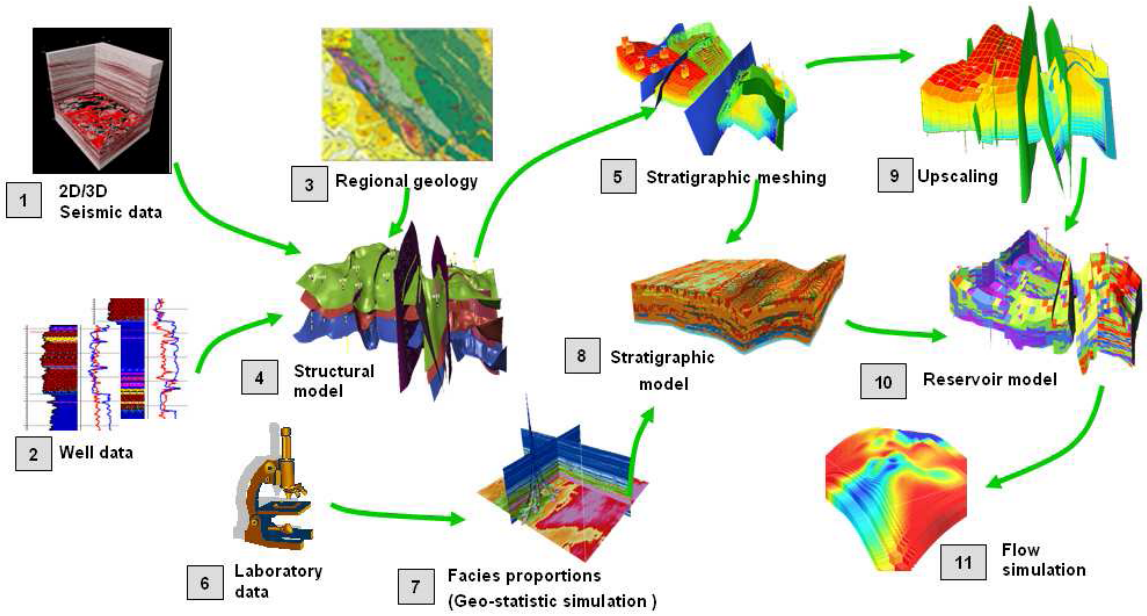


FIG. 3.4 – Vue d'ensemble de la modélisation géologique d'un gisement (le processus *earth modeling*).

Le processus *earth modeling* commence avec la définition d'un *prospect* (une perspective), qui correspond à une zone d'intérêt en 3D. L'acquisition initiale de données concernant un *prospect* est essentiellement réalisée au moyen de la réflexion sismique (figure 3.4(1)) et les forages aux puits (figure 3.4(2)). Les géologues tiennent également compte des études antérieures sur le *prospect*, tels que des documents concernant la géologie régionale, des cartes géologiques ou des coupes transversales (figure 3.4(3)). A partir de ces informations, une *interprétation sismique* est réalisée pour former un premier modèle d'assemblage de surfaces géologiques appelé *modèle structural*. Nous détaillons cette étape dans les sections suivantes car c'est sur elle que nous appliquons notre proposition.

Par la suite, dans l'activité de modélisation stratigraphique, des mailles stratigraphiques sont construites à l'intérieur de chacun des blocs du modèle structural (figure 3.4(5)). Des propriétés pétrophysiques doivent alors être affectées à chaque cellule de chacune des mailles. Pour cela, des outils de modélisation informatique examinent d'abord les propriétés acquises à partir de points isolés correspondant à des échantillonnages et à des études en laboratoire (figure 3.4(6)). Ces propriétés sont ensuite propagées à l'ensemble du volume à l'aide de simulations géostatistiques (figure 3.4(7)). Le modèle qui en résulte, où les mailles stratigraphiques sont enrichies par des propriétés sur la roche, est appelé modèle stratigraphique (figure 3.4(8)). Pour utiliser ce

modèle dans les simulations, il est nécessaire de transformer la géométrie du maillage du modèle stratigraphique afin d'obtenir un maillage de réservoir plus gros, et ensuite de faire évoluer les valeurs de propriétés (figure 3.4(9)). Il en résulte un modèle de réservoir (figure 3.4(10)), qui fournit un ensemble complet de caractéristiques dynamiques sur le réservoir (c'est-à-dire la porosité, la perméabilité ou la saturation en eau) pour chaque cellule de la grille 3D. Il sera utilisé par les ingénieurs de réservoir pour obtenir un calcul réaliste de simulation de migration des hydrocarbures liquides (figure 3.4(11)). Il peut ainsi à terme être utilisé pour estimer le montant des réserves d'hydrocarbures exploitables présentes dans le réservoir et la qualité de ces réserves (pétrole lourd ou léger, gaz, etc.) mais aussi pour une simulation de stockage de CO_2 dans des réservoirs pétroliers en fin de production.

3.2 Interprétation sismique

L'interprétation sismique permet à partir de données sismiques et de données de puits, de reconstruire un modèle du sous-sol géologiquement cohérent. Cette interprétation est à la base de la construction d'un modèle structural. Une interprétation sismique 3D est une opération qui nécessite couramment plusieurs mois de travail. Selon [Clark, 1998], l'interprétation sismique représente la majeure partie du travail d'un géophysicien.

3.2.1 Acquisition d'images sismiques

L'acquisition de données sismiques concernant un champ pétrolier *on shore* ou *off shore* est réalisée au moyen d'une technique appelée *réflexion sismique*. Le but est d'explorer le sous-sol en profondeur. La *réflexion sismique* est une technique de mesure indirecte qui consiste à enregistrer en surface à l'aide de capteurs des échos résultant de la propagation d'ondes acoustiques produites artificiellement (dynamite, canon à air, etc.) dans la mer ou dans le sous-sol. Ces échos sont générés par des interfaces sédimentaires qui correspondent à un changement assez brutal des propriétés physiques (densité, élasticité, etc.) du sous-sol et qui, de ce fait, réfléchissent les ondes sismiques. La figure 3.5 illustre cette technique.

Ainsi, dans les terrains stratifiés, des changements considérables des propriétés physiques des roches sont généralement observés lors du franchissement d'une frontière sédimentaire. De tels changements dans les propriétés du milieu physique provoquent une réflexion d'ondes acoustiques. Les ondes réfléchies par les diverses limites sédimentaires vers la mer ou la surface de la terre sont enregistrées et constituent une image sismique. De plus, deux propriétés fondamentales sont enregistrées par les capteurs : l'amplitude de l'onde réfléchie et la durée du trajet de l'onde. Ainsi, à chaque position de récepteur, est enregistré un signal qui représente l'amplitude de la réflexion en fonction du temps. Ce signal s'appelle une trace. Il constitue les colonnes d'une image sismique dont nous illustrons un exemple dans la figure 3.6.

Une telle image sismique montre des lignes de couleurs et de largeurs différentes correspon-

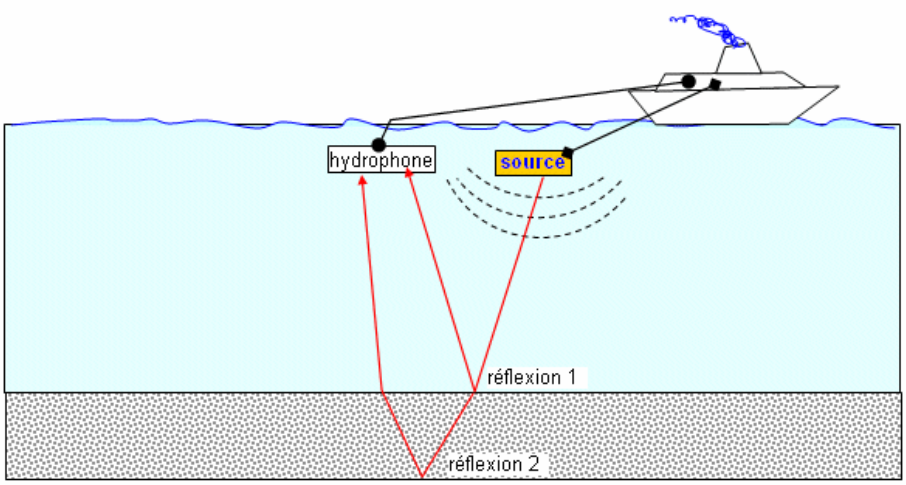


FIG. 3.5 – Réflexion sismique.

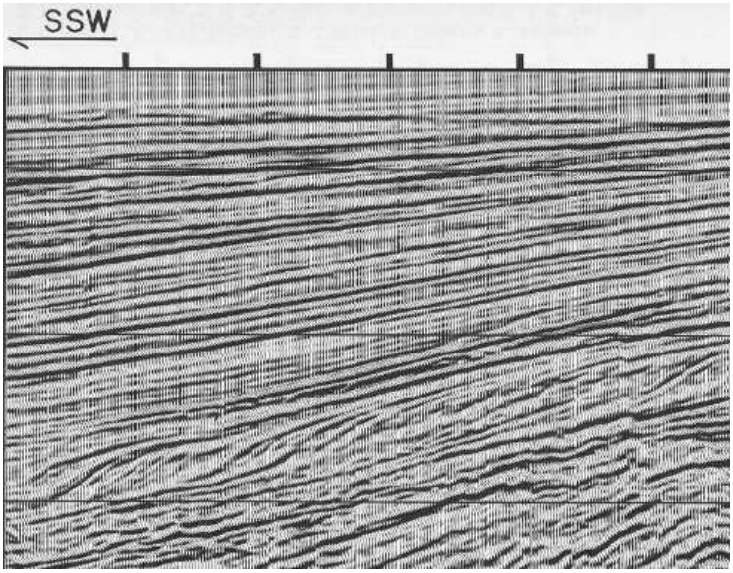


FIG. 3.6 – Exemple d'une image sismique.

dant à des variations des amplitudes des ondes. Chaque ligne correspond à un *réflecteur*, c'est-à-dire à une limite stratigraphique. L'image 3D est assimilable à une matrice à trois dimensions appelée bloc sismique. Les coordonnées latérales de ce bloc sont relatives à la position géographique des capteurs et les coordonnées verticales sont relatives au temps de retour de l'onde. Enfin, les valeurs associées à chaque cellule de cette matrice sont représentatives de l'amplitude de la réflexion en ce point.

Un bloc sismique est représentatif d'un champ pétrolier de plusieurs dizaines à plusieurs milliers de kilomètres carrés. Il s'étend en profondeur sur quelques kilomètres. Le pas latéral entre deux voxels est environ de 10 à 50 mètres. Il est à noter que la longueur de l'onde envoyée dans le sous-sol ne permet pas de détecter des différences de milieu dans des couches de moins

de 7,5 mètres d'épaisseur dans les meilleures conditions.

Bien que la dimension verticale soit initialement exprimée en temps (en secondes), il est aussi possible de la convertir en profondeur (en mètres) afin que les relations de distance entre les différents éléments de l'image soient plus proches de la réalité. Pour cela, il est nécessaire de connaître les vitesses de propagation des ondes dans les milieux traversés. En effet, si l'on connaît la vitesse de propagation d'une onde dans un milieu et le temps qu'elle a mis pour le traverser, on peut en déduire l'épaisseur de ce milieu. Un bloc temps est donc légèrement trompeur par rapport à la réalité. Cependant, la transformation en bloc profondeur est une opération lourde car elle nécessite des temps de calcul très importants.

3.2.2 Marqueurs aux puits

En parallèle aux données sismiques, le géologue examine en général une liste de *logs* (fichiers-journaux) sur les puits où sont stockés les changements des propriétés physiques des roches traversées par des trajectoires de forage. Chaque *log* de puits peut être ainsi interprété comme une succession de *marqueurs* au puits, chacun correspondant à une discontinuité lithologique. Une profondeur dans le trou de forage est associée à chacun de ces *marqueurs*. En conséquence, les *logs* de puits peuvent fournir des informations très précises sur la géométrie et sur la position des horizons sédimentaires dans la zone d'étude. Grâce à la tâche de corrélation des puits, les marqueurs sont utilisés pour ajuster la position verticale de chacune des surfaces identifiées à travers l'interprétation sismique. Les géologues effectuent des corrélations entre plusieurs *logs* de puits, des sections de coupe sismique, et les propriétés de base. Ce travail d'identification et de corrélation permet de retrouver les relations entre les structures déterminées sur l'image sismique, ce qui sera utile pour construire le modèle structural.

3.2.3 Extraction d'objets d'intérêt

A l'aide d'outils informatiques, les géologues effectuent une tâche d'interprétation sismique sur l'image sismique, qui consiste à identifier les *patterns* qui seront reconnus comme des surfaces (tels que des horizons et des failles) ou des assemblages de surfaces correspondant à des objets sédimentaires (tels que des chenaux). Après avoir identifié un objet, le géologue choisit des points pour préciser sa géométrie. Ces points sont comptabilisés comme correspondant à une ou plusieurs surfaces interprétées. L'extraction d'objets d'intérêt se fait par *picking* et *tracking*. Le *picking* est une opération qui vise à pointer manuellement un objet d'intérêt à l'aide de la souris. Il est généralement effectué sur une section 2D de l'image sismique. Le *tracking* (en 3D) ou *autopicking* (en 2D) est un algorithme dont le but est d'étendre la sélection faite lors du *picking* sur l'ensemble de l'image sismique dont on trouve une bonne illustration dans [Dorn, 1998].

Le résultat d'une telle opération fournit un ensemble de surfaces géologiques. Les profondeurs apparentes de ces surfaces sur l'image sismique ne correspondent pas strictement à des

profondeurs réelles car les enregistrements sismiques ne correspondent qu'à des temps de propagations (à savoir le délai pendant lequel une onde donnée atteint la mer ou la surface de la terre). Il faut donc encore assembler les surfaces obtenues de manière cohérente afin d'obtenir un modèle structural, sur lequel pourront être basés l'ensemble des *earth models* relatifs à la zone prospectée.

3.2.4 Construction de modèle structural

L'assemblage des surfaces extraites d'une image sismique permet la construction d'un modèle structural [Foucault et Raoult, 2005]. A l'aide d'outils de modélisation informatique (appelé *geomodelers*), les surfaces identifiées par l'interprétation sismique sont chargées dans un espace 3D sans autre information que leur géométrie. Tenant compte en plus de toutes les données disponibles issues des études de la géologie régionale (rapports, documents de recherche, cartes, etc), les géologues procèdent ensuite à une interprétation sur la structure. Cela permet de spécifier les relations spatiales et chronologiques entre les objets identifiés. Le géologue sélectionne les surfaces qu'il souhaite intersecter et informe le logiciel sur la façon d'opérer cette intersection (A intersecte B ou B intersecte A). Le logiciel réalise l'opération et enregistre la relation topologique dans un modèle structural. La topologie de l'assemblage d'objets est d'une importance capitale car elle dépend strictement de l'interprétation géologique [Perrin, 1998].

Le choix des surfaces à intersecter relève d'une interprétation du géologue car il peut exister, dans un cas déterminé, plusieurs solutions pour assembler des surfaces de manière géologiquement cohérente. L'interprétation doit donc être couramment effectuée au cas par cas par le géologue en fonction de sa connaissance du contexte géologique (qui n'est d'ailleurs pas toujours aisé à appréhender de manière correcte dans le cas où le nombre de surfaces entrant dans le modèle est élevé).

Cette étape est cruciale dans le processus de modélisation géologique parce que le modèle structural (figure 3.4(4)) est le "squelette" sur lequel les autres *earth models* seront construits. Il consiste essentiellement en un assemblage de surfaces géologiques qui marquent les limites des différents blocs géologiques.

Nos travaux s'intéressent à la gestion des processus de modélisation géologique et particulièrement aux processus potentiels d'interprétation sismique qui les composent. Dans la prochaine section, nous décrivons une représentation d'un processus métier d'interprétation sismique. Nous montrons ensuite comment les fonctions qui composent ces processus sont implémentées en programmes et logiciels informatiques et comment les workflows résultants sont exécutés.

4 Gestion des processus en modélisation géologique

4.1 Processus d'interprétation sismique

L'interprétation sismique permet de construire un modèle structural à partir de la composition d'objets géologiques basiques. La figure 3.7 montre un exemple d'un processus réalisant une interprétation sismique tel qu'il est représenté par des géologues.

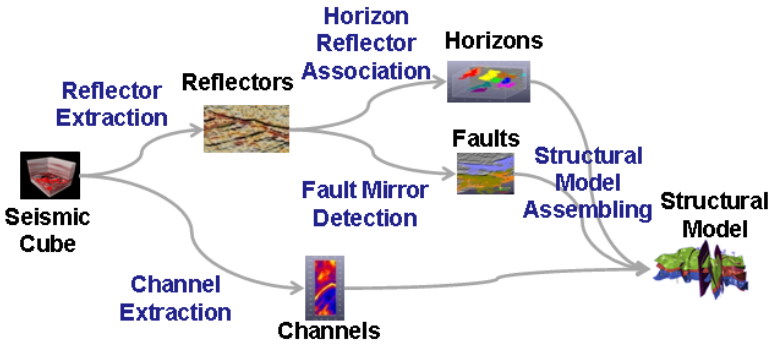


FIG. 3.7 – Un processus d'interprétation sismique représenté par des géologues.

Dans cette figure, à partir d'un *SeismicCube* qui représente les images sismiques d'un réservoir donné, une extraction de réflecteurs est effectuée (*ReflectorExtraction*) afin d'en extraire les *Reflector* qui sont de petites parties visibles des éléments géologiques. Deux actions sont réalisées en parallèle : une association de réflecteurs pour la formation d'horizons (*HorizonReflectorAssociation*) et la formation de failles (*FaultMiroirDetection*) pour obtenir respectivement les *Horizon* et les *Fault*. En parallèle, les chenaux (*Channel*) sont extraits du cube sismique (*SeismicCube*) par l'activité *ChannelExtraction*. Enfin, les trois objets géologiques *Horizon*, *Fault* et *Channel* sont combinés par l'activité *StructuraleModelAssembling* pour obtenir un modèle structural (*StructuralModel*).

La représentation du processus décrit dans la figure 3.7 permet de spécifier les fonctions et les relations entre ces fonctions caractérisant l'interprétation sismique ainsi que les objets géologiques manipulés. Ainsi, ces fonctions peuvent être formalisées et implémentées, ce que nous montrons dans la section suivante.

4.2 Implémentation de processus en workflows informatiques

Nous avons vu dans la section précédente que les géologues définissent des fonctions tels que *ReflectorExtraction* et *HorizonReflectorAssociation*, les objets géologiques que ces fonctions manipulent et les processus les composant. L'objectif est ensuite de fournir des solutions logicielles qui implémentent les fonctions et des modèles informatiques qui représentent les objets géologiques. Ainsi, des logiciels et des programmes informatiques sont conçus pour ré-

pondre à ces besoins et des workflows composés de ces logiciels et programmes informatiques sont construits pour implémenter les processus définis. La figure 3.8 est une représentation graphique d'un workflow implémentant le processus défini dans la figure 3.7. Ce workflow décrit l'enchaînement des tâches (logiciels et programmes informatiques) à exécuter par les géologues pour réaliser une interprétation sismique.

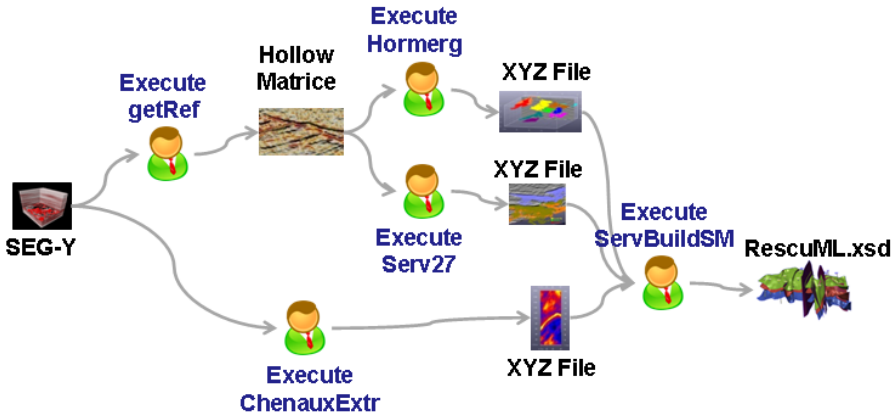


FIG. 3.8 – Représentation graphique d'un workflow informatique d'interprétation sismique.

Dans cette figure, à partir d'un fichier conforme à un modèle de données *SEG-Y* représentant un cube sismique, un géologue doit exécuter le logiciel *getRef* pour extraire des réflecteurs et générer en sortie un fichier conforme à un modèle *HollowMatrix* qui est une représentation de matrice creuse de coordonnées de points. Ensuite, deux activités peuvent être réalisées en parallèle par les géologues. La première consiste à combiner les réflecteurs du fichier *HollowMatrix* à l'aide du logiciel *Hormerg* et obtenir des horizons dans un fichier conforme à un modèle *XYZ File*, représentant des listes de coordonnées de points. Une autre activité consiste à détecter les failles et les enregistrer dans un fichier également conforme à un modèle *XYZ File*. En parallèle à toutes ces activités, un fichier conforme au modèle *XYZ File* représentant les chenaux est obtenu en exécutant le logiciel *ChenauxExtr*. Enfin, les trois fichiers conformes au modèle *XYZ File* sont utilisés par un géologue et entrés dans le logiciel *ServBuildSM* pour obtenir un fichier conforme au modèle *RescuML* représentant un premier modèle structural.

Certains systèmes permettent d'automatiser les workflows créés en composant des programmes informatiques ne nécessitant ainsi plus d'intervention humaine. *Workflow OpenFlow*, est un système de gestion de workflows développé par *IFP Énergies Nouvelles*, et commercialisé par *Beicip-Franlab* [Beicip-Franlab, 2010] reposant sur ce principe. La figure 3.9 représente une vue du logiciel *Workflow OpenFlow*.

Le système *Workflow OpenFlow* fournit un éditeur de workflows permettant de définir un workflow informatique à partir des activités (qui sont des programmes informatiques) disponibles dans un produit. Toutes les activités sont implémentées sous la forme de plugins d'*OpenFlow* et correspondent aux fonctions de modélisation géologique entièrement automatisables et ne nécessitant pas d'intervention humaine. Certaines fonctions sont implémentées par

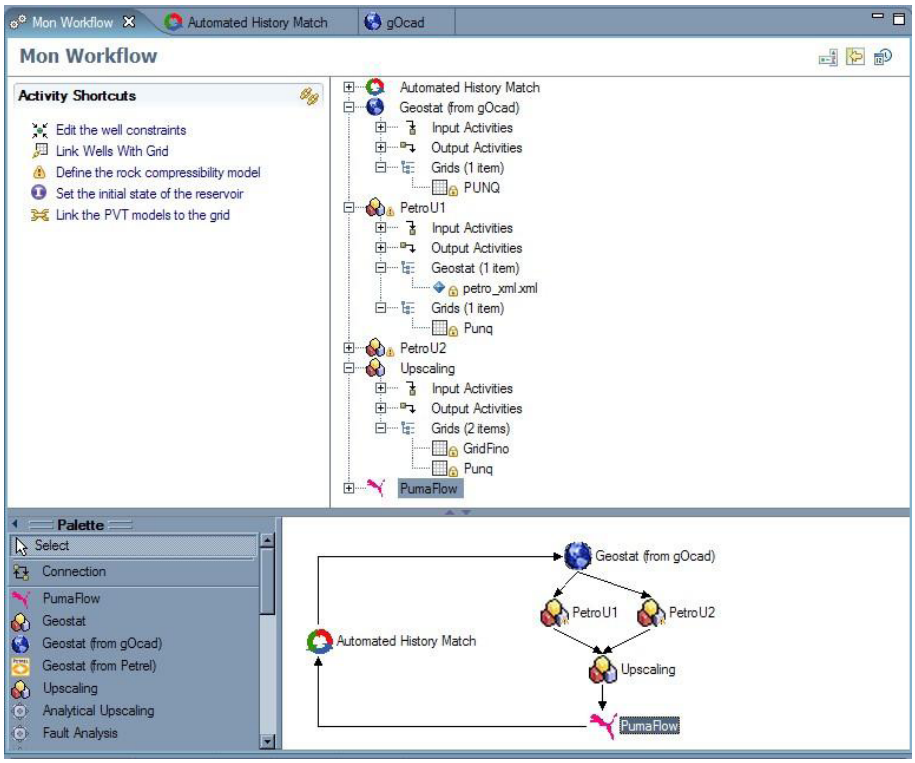


FIG. 3.9 – Vue de l'éditeur de workflow de OpenFlow de *IFP Énergies Nouvelles.*

des logiciels nécessitant des interactions avec des géologues. Les processus dont ces fonctions font partie ne peuvent pas être transformés en workflows informatiques automatiques.

Toutefois, l'automatisation n'est pas l'unique objectif. En effet, dans les systèmes actuels que les géologues utilisent, seules les descriptions syntaxiques et informatiques des processus et des objets géologiques manipulés persistent. Tous les logiciels, les programmes informatiques de même que les modèles de données qu'ils manipulent ne sont pas explicitement caractérisés de manière sémantique. Ainsi, les géologues ne peuvent pas rechercher des logiciels ou des workflows déjà créés par leurs sens mais uniquement par leurs noms. Ils ne peuvent avoir accès qu'à des descriptions syntaxiques utiles à leurs exécutions mais pas à la compréhension des fonctions qu'ils réalisent.

5 Conclusion

Dans ce chapitre nous avons défini le contexte dans lequel nous menons notre recherche. Nous avons présenté quelques objets géologiques simples comme les horizons qui représentent des couches de sédiments et les failles qui représentent des discontinuités qui séparent des parties de mêmes couches sédimentaires. Ceci constitue le vocabulaire nécessaire à la compréhension du contexte de nos travaux. Nous avons ensuite exposé la modélisation géologique qui vise

à représenter de manière plus ou moins fidèle un prospect pétrolier en fin de production (pour le stockage de CO_2). La modélisation géologique rend ainsi possible un stockage efficace et sûr en permettant la réalisation d'une simulation d'écoulement de fluides et plus particulièrement de CO_2. La modélisation géologique peut être représentée par un processus composé de grandes étapes dont l'interprétation sismique qui se concrétise par la construction du modèle structural. Notre choix s'est porté sur l'interprétation sismique en tant qu'étude de cas, que nous avons détaillée, car elle représente d'une part une étude de cas suffisamment complexe pour l'application de notre proposition, et d'autre part, nous disposons de nombreux éléments de modélisation de données à travers les études menées par [Verney, 2009][Mastella, 2010].

Dans une seconde partie, nous avons présenté la gestion actuelle des processus de modélisation géologique. Nous avons vu que l'interprétation sismique par exemple est modélisée en processus et chaque fonction qui compose le processus est implémentée en un programme ou logiciel informatique. Certains systèmes de gestion de workflows permettent de créer des workflows automatiques à condition que ces systèmes proposent des programmes informatiques ne nécessitant pas d'intervention humaine. Cependant, lors de l'implémentation des programmes et logiciels informatiques, le processus de conception n'est pas explicitement décrit. De plus, les modèles de données qui servent à stocker la grande quantité de données issue de la modélisation géologique ne sont pas explicitement liés aux concepts géologiques qu'ils représentent. Enfin, les implémentations informatiques et les modèles de données conçus dépendent des besoins ponctuels des géologues. Ainsi, aucune ressource, excepté les noms et les descriptions syntaxiques des workflows, des services qui les composent et celles des modèles de données qu'ils manipulent ne peuvent donner des informations sur leur signification et sur leurs rôles.

Dans ce qui suit, nous présentons une synthèse des trois premiers chapitres du mémoire qui constituent l'état de l'art. Nous exposons les insuffisances liées à la description et à la gestion de compositions de services informatiques. Nous mettons en évidence les besoins en enrichissement sémantique et les relions aux insuffisances de la gestion actuelle des services et des workflows en modélisation géologique.

Synthèse générale de l'état de l'art

Les trois premiers chapitres de ce mémoire sont relatifs à l'état de l'art qui permet de croiser le domaine des services et des workflows avec celui des ontologies et des annotations sémantiques et de relier ces deux domaines au contexte de nos travaux de recherche, à savoir la modélisation géologique pour le stockage du CO_2.

Dans les domaines de l'ingénierie, les logiciels permettent d'assister les scientifiques dans la réalisation de leurs tâches quotidiennes de modélisation, de traitement d'informations, etc. En ingénierie pétrolière par exemple, des solutions logicielles sont fournies, elles implémentent des fonctions décrites par des géologues pour leur permettre par exemple d'extraire des objets géologiques d'une image sismique.

Certains programmes informatiques permettent d'effectuer des travaux répétitifs ou nécessitant une grande précision, des calculs arithmétiques ou logiques, ou de simulation de manière entièrement automatique. En outre, avec l'apparition des services Web, la communication entre applications distantes à travers Internet indépendamment de tout langage de programmation est devenue possible. La composition de programmes informatiques et particulièrement de services Web a permis de créer des fonctions de plus en plus complexes apportant une plus grande valeur ajoutée. Cependant, dans certains domaines tels que l'ingénierie pétrolière, l'enchaînement des tâches reste souvent manuelle. Les géologues naviguent d'un logiciel à un autre en perdant du temps et de la précision. Des propositions d'algorithmes permettant d'automatiser certaines tâches des géologues rendant possible l'implémentation de programmes plus évolués et surtout automatiques. Ainsi, des compositions de ces programmes peuvent être effectuées rendant des parties de processus entièrement automatiques.

Les workflows permettent donc la systématisation des tâches à effectuer par les acteurs des domaines d'ingénierie. Quand ces workflows correspondent à des compositions de programmes ne nécessitant pas d'intervention humaine, ils deviennent entièrement automatiques. Cependant, ces workflows et compositions de services informatiques sont souvent décrits en ne suivant aucune méthodologie formelle. Ainsi, vu qu'ils ne sont plus explicitement liés aux fonctions et aux processus qu'ils implémentent, aucune ressource à l'exception de leurs descriptions syntaxiques ne peut fournir d'informations sur ce qu'ils réalisent. Cette absence de sémantique

explicite rend les logiciels, les programmes informatiques et les workflows qui les composent difficilement réutilisables. La composition de ces services et de ces workflows existant en vue d'en construire d'autres réalisant des tâches et des objectifs plus complexes est encore moins envisageable. De plus, il n'est pas non plus possible de retrouver un programme ou un workflow suivant ce qu'il réalise sans une connaissance précise de sa description ou de son nom. Enfin, exécuter ces programmes et workflows nécessite une connaissance informatique technique qui bien souvent fait défaut aux scientifiques des domaines d'ingénierie tels que l'ingénierie pétrolière. Ces scientifiques restent ainsi très dépendants des développeurs de logiciels que ce soit pour l'utilisation de programmes et de workflows existants ou pour la conception de nouveaux workflows même si ces derniers ne sont que la composition de programmes et de workflows existants.

Pour apporter une solution aux problèmes évoqués, les systèmes classiques de gestion de workflows doivent être sémantiquement enrichis pour permettre aux scientifiques et en particulier à ceux spécialisés en modélisation géologique, d'effectuer des recherches sémantiques sur les services et workflows mais aussi d'en construire de nouveaux sans disposer de connaissances techniques. Dans cette perspective, de récentes approches d'enrichissement de systèmes de workflows et de compositions de services Web ont été proposées. Les ontologies permettant d'expliciter la sémantique d'un domaine d'étude, se posent comme une solution pertinente pour enrichir sémantiquement la description de programmes et workflows informatiques. Nous avons passé en revue quelques unes de ces approches basées sur les ontologies et les annotations sémantiques telles que OWL-S, SAWSDL ou WSMO. Ces approches présentent en revanche des insuffisances. Leurs mécanismes d'enrichissement sémantique procèdent soit par annotation sémantique soit par extension des langages de descriptions syntaxiques. Ainsi, les services et les workflows surchargés ne peuvent plus être utilisés sans que de nouveaux outils soient développés. De plus, aucune classification hiérarchique ou relationnelle selon la sémantique des services et workflows informatiques n'est fournie rendant la recherche ou l'adaptativité restreinte.

Nous pensons que pour remédier à ces limitations, il faut permettre d'une part, une séparation claire entre les services et workflows informatiques et leur description sémantique, et d'autre part, une hiérarchisation sémantique de ces services et workflows informatiques. Nous proposons d'utiliser des concepts issus des ontologies et des annotations sémantiques pour enrichir les descriptions syntaxiques d'une couche sémantique sans les surcharger et permettant une meilleure gestion des services et des workflows informatiques.

L'approche que nous proposons se base sur la création d'*ontologies de services* dont les concepts sont des *services sémantiques*. La création d'ontologies de services nécessite une modélisation consensuelle des fonctions d'un domaine d'ingénierie particulier en caractérisant ses fonctions, les objets qu'elles manipulent et les relations entre ces fonctions. Notre démarche ne part pas de termes décrivant ces concepts, mais de caractérisations de concepts. D'autre part, la représentation d'une ontologie de services au sein de bases de données nécessite d'avoir accès au modèle d'ontologie et de pouvoir le modifier. Nous avons vu que seules les BDBO de type 3

permettent la modification du modèle d'ontologies utilisé. Nous exploiterons ainsi la BDBO de type 3 OntoDB ainsi que son langage OntoQL que nous avons présentés. Ainsi, nous utilisons OntoDB comme un entrepôt sémantique pour y stocker les services, les workflows informatiques et leurs descriptions sémantiques. Cela permet de rechercher et de composer des services et workflows par leur sens et les relations avec les autres services et workflows à travers la subsomption et leurs caractérisations fonctionnelles. Cela permet de remplacer des services et des workflows par d'autres services et workflows répondant mieux aux besoins spécifiés grâce à l'indexation des services et des workflows par des services sémantiques, etc.

Enfin, nous appliquons notre proposition dans le domaine de la modélisation géologique. Nous avons développé un logiciel que nous avons appelé GWE (Geological Workflow Editor) et qui permet la gestion des services et workflows de la modélisation géologique basée sur notre proposition.

Dans la seconde partie de ce mémoire, nous présentons notre approche de manière détaillée et montrons son implémentation sur OntoDB. Dans la troisième et dernière partie, nous appliquons notre approche à la modélisation géologique et présentons le prototype GWE (Geological Workflow Editor) que nous avons développé.

Deuxième partie

Notre proposition

Chapitre 4

Un modèle conceptuel pour une gestion sémantique de services et de workflows

Sommaire

1 Introduction 85

2 Enrichissement sémantique de modèles de données 87

2.1 Modèles de données 87

2.2 Ontologies de domaine 88

2.3 Indexation des modèles de données 90

3 Enrichissement sémantique des services et des workflows informatiques 93

3.1 Services informatiques 93

3.2 Ontologies de services 94

3.3 Indexation de services et de workflows informatiques 97

3.4 Modèle conceptuel global 98

3.5 Contrôle sur les ontologies de services 98

4 Construction de nouveaux workflows 100

4.1 Description de workflows sémantiques 101

4.2 Composition fonctionnelle dans les workflows sémantiques . . . 102

4.3 Évolution de l'ontologie des services 103

4.4 Construction et génération de workflows informatiques 104

5 Exploitation de l'approche 104

6 Conclusion 105

Résumé. Dans les chapitres précédents, nous avons présenté les travaux effectués pour la description sémantique de services et de workflows. Nous avons montré les points faibles et forts de chaque approche ce qui nous permet de les comparer à

notre approche. Dans ce chapitre, nous présentons notre approche d'enrichissement sémantique de services et de workflows de manière détaillée.

1 Introduction

Actuellement, les services et les workflows informatiques sont définis syntaxiquement dans des langages de définition tels que WSDL ou BPEL. Par contre, ces langages ne permettent pas d'associer de sémantique explicite à ces définitions syntaxiques. En effet, ces dernières se limitent à un nom de service, à un nom et à un type de paramètre, ce qui ne permet pas à une machine d'interpréter les services rendus. Cela entraîne d'une part, la difficulté de réutiliser des services et des workflows informatiques déjà définis, et d'autre part, il devient difficile de construire de nouveaux workflows informatiques à partir de services et de workflows informatiques existants. L'objectif final de notre proposition est de permettre une meilleure gestion des services et des workflows informatiques en offrant la possibilité de :

1. rechercher par leur sémantique et non plus par leur description syntaxique les services et les workflows informatiques ;

2. construire des workflows informatiques en composant des services et des workflows informatiques uniquement en se référant à leur sémantique.

L'idée principale qui caractérise notre proposition [Ait-Ameur, 2009], [Belaid et al., 2009c], [Belaid et al., 2009b], [Belaid et al., 2009a], [Belaid et al., 2010] est de permettre l'enrichissement sémantique des services et des workflows informatiques. Elle se base sur :

- l'explicitation de la sémantique des services et des workflows informatiques dans un domaine cible indépendamment des services et des workflows informatiques eux-même en créant des ontologies de services. Ces ontologies étant notamment basées sur la relation de subsomption, elles permettent de caractériser hiérarchiquement les concepts sémantiques de services et de workflows ;
- l'indexation ou l'annotation sémantique des workflows et des services informatiques existants par leurs descriptions sémantiques, c'est-à-dire par les concepts des ontologies de services créées. Dans notre proposition, nous procédons de même pour les modèles de données auxquels les services et les workflows informatiques font référence dans leurs descriptions.

Cette approche facilite la réutilisation de services et de workflows informatiques déjà conçus et utilisés. En effet, ces derniers sont indexés à travers des *services sémantiques* qui sont des concepts d'ontologies de services. Ceci permet de faciliter leur recherche et leur interprétation. Par ailleurs, ceci permet d'indexer des descriptions hétérogènes de services et de workflows informatiques sans modifier leur description syntaxique. Cette approche facilite également la construction de nouveaux workflows à partir de workflows et de services informatiques préalablement indexés. Cette construction est possible grâce au concept de *workflow sémantique* qui consiste à construire des workflows en composant des services sémantiques, et ensuite en construisant, voire en générant, des workflows informatiques.

La Figure 4.1 montre la conceptualisation de notre approche.

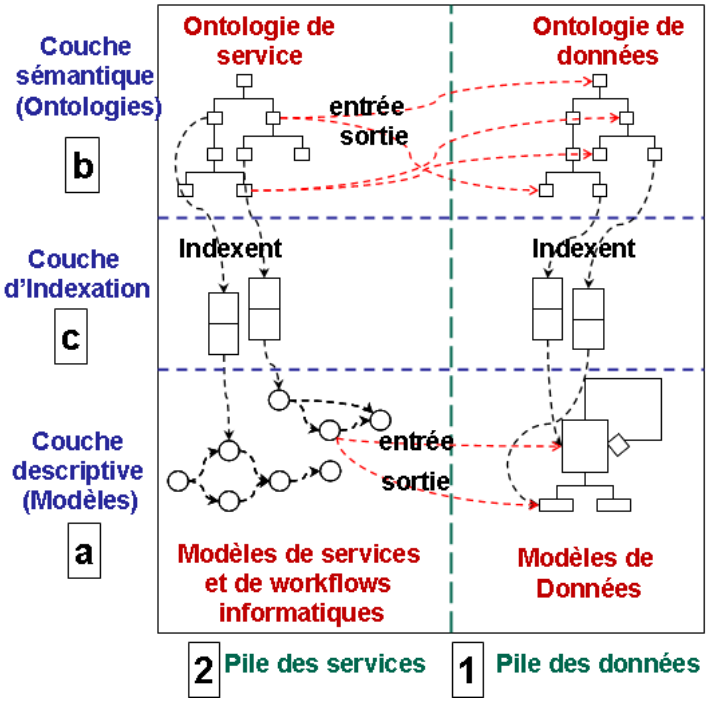


FIG. 4.1 – Approche à base ontologique pour la gestion de workflows.

Sur la figure 4.1, nous constatons qu'il y a trois couches (a), (b) et (c) suivant un découpage horizontal et deux piles (1) et (2) suivant un découpage vertical. Les deux piles de l'approche concernent la séparation entre les modèles de données et leur définition sémantique d'une part, et entre les services, les workflows informatiques et leur définition sémantique d'autre part. Les trois couches de l'approche concernent la séparation entre la couche des modèles de données et des descriptions de services et de workflows informatiques (couche syntaxique), la couche de description sémantique (couche sémantique) et la couche reliant les deux autres (couche d'indexation). Elle montre que notre approche consiste en un enrichissement sémantique de descriptions de services informatiques et de workflows référençant des modèles de données en utilisant un mécanisme d'indexation.

Dans ce qui suit, nous détaillons l'approche que nous proposons. Nous commençons par expliquer comment l'enrichissement sémantique des modèles de données par les concepts d'ontologies de domaine est réalisé. Ensuite, nous décrivons le mécanisme permettant l'enrichissement sémantique des descriptions de services et de workflows informatiques existants par les concepts d'ontologies de services (services sémantiques). Nous expliquons comment sont construits les workflows sémantiques et comment nous pouvons en "déduire" de nouveaux workflows informatiques. Enfin, nous montrons comment nous pouvons exploiter cette approche pour la réalisation des buts précédemment évoqués et permettre une meilleure gestion des workflows et des services informatiques.

2 Enrichissement sémantique de modèles de données

Les services informatiques sont décrits en référençant des descriptions de modèles de données (voir la colonne (1) dans la figure 4.1). Nous montrons dans cette section notre approche pour représenter les modèles de données du domaine d'application ainsi que leurs définitions sémantiques. C'est la partie concernant les données statiques de notre approche.

2.1 Modèles de données

Les modèles de données (voir la partie (1)(a) dans la Figure 4.1) représentent les concepts métiers manipulés ou produits par les services informatiques. Il s'agit de modèles de fichiers, de descriptions de données ou de formats. Ils représentent les éléments de données tels qu'ils sont manipulés ou produits par les services informatiques. Ils peuvent contenir des types simples comme les types de bases, c'est-à-dire entier, chaînes de caractères, etc. Ils peuvent également contenir des types complexes, par exemple, des fichiers de description tels que les XML-Schema qui décrivent la représentation de fichiers XML et permettent de représenter les éléments du domaine d'application.

Pour représenter les modèles de données utilisés pour décrire les services, nous proposons d'utiliser un modèle commun auquel les modèles de données seront conformes : le méta-modèle de données. Ainsi, ce méta-modèle de données utilisé devra contenir l'ensemble des caractéristiques nécessaires pour décrire de façon uniforme les modèles de données (nom de fichier, identification, principaux objets composés, etc.). La Figure 4.2 présente les principaux éléments du méta-modèle de données défini par [Mastella, 2010], que nous proposons de réutiliser.

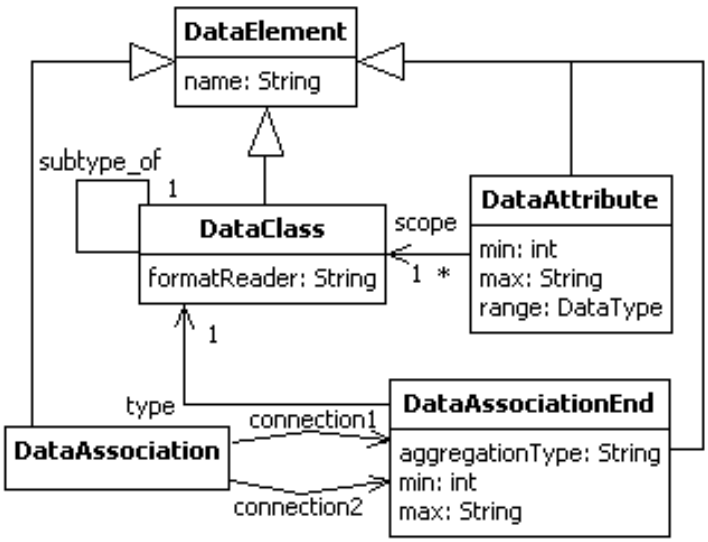


FIG. 4.2 – Un méta-modèle de données.

L'élément *DataElement* est la super-entité abstraite qui regroupe toutes les autres entités du

méta-modèle. *DataClass* est la primitive de modélisation des catégories de données. Les instances de *DataClass* ont chacune un nom et elles peuvent être organisées dans une hiérarchie de spécialisation/généralisation au moyen du lien *subtype_of*. Les attributs d'instance du *DataClass* sont définis grâce à l'entité *DataAttribute*, qui peut avoir une cardinalité minimale et maximale (champs *min* et *max*) et un champ *range* dont le type est défini par l'entité *DataType*. *DataAssociation* permet de représenter des relations binaires entre des *DataClass*. Les instances de *DataAssociation* sont des liens entre des instances de *DataClass* qui n'ont ni état ni identité. *DataAssociation* est reliée à deux entités *DataAssociationEnd*, qui spécifient les instances du *DataClass* auxquelles l'association est reliée, la multiplicité (champs min et max) et le type de l'association (champs *aggregationType*) à chacun des points d'extrémité.

Ces éléments sont les blocs constitutifs qui permettent de représenter n'importe quelle entité utilisée dans des modèles de données. Ainsi, à partir de ce méta-modèle de données, il est possible de représenter n'importe quel modèle de données comme une instance de ce méta-modèle. Notons que ce méta-modèle est très proche du MOF proposé par l'OMG [OMG, 2006]. Nous avons choisi ce méta-modèle plutôt que d'utiliser le MOF car il présente l'avantage d'être plus simple et d'avoir déjà été implémenté sur la plateforme OntoDB que nous utilisons pour valider notre approche. Cependant, le MOF pourrait également être utilisé.

Les modèles de données utilisés sont souvent spécifiques à une application donnée et en conséquence ne décrivent généralement que les concepts nécessaires à cette application sans en donner une description précise (par exemple les unités de mesure ne sont pas spécifiées). Pourtant, une telle description est nécessaire pour faciliter la recherche et la composition de services. Aussi, nous proposons d'indexer ces modèles de données par des ontologies de domaine.

2.2 Ontologies de domaine

Les ontologies de domaine (voir la partie (1)(b) dans la Figure 4.1) étant des conceptualisations partagées, elles permettent de décrire explicitement la sémantique d'un domaine de connaissance particulier. Par exemple, l'ontologie que nous présentons dans la figure 4.3 représente un fragment d'une ontologie sur la documentation construite à des fins didactiques. Une ontologie contient des descriptions précises et partagées sur un domaine. L'ontologie que nous présentons n'est qu'un exemple simplifié et restreint qui nous permet d'expliquer notre approche.

Ainsi, dans cette figure, des documents peuvent être des documents scientifiques, administratifs, etc. Les documents scientifiques peuvent être des documents scientifiques de recherche ou des documents scientifiques techniques. Les documents administratifs sont soit des documents de marketing, des documents financiers ou des documents de ressources humaines. Les documents financiers sont composés de factures. Des factures peuvent être plus précisément des factures électroniques.

Les ontologies de domaine sont définies selon un modèle appelé le modèle d'ontologie. Ce

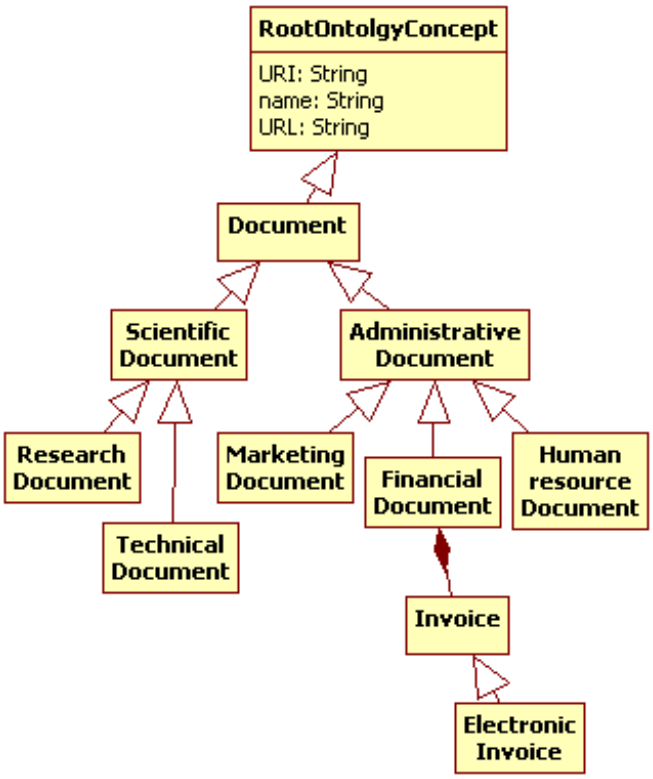


FIG. 4.3 – Une ontologie de domaine sur la documentation.

modèle d'ontologie contient un ensemble de primitives permettant de décrire les concepts d'une ontologie de domaine. La figure 4.4 représente un exemple de modèle d'ontologie contenant les principales primitives. De nombreux modèles d'ontologie réels existent tels que OWL [Dean et Schreiber, 2004] et PLIB [ISO13584-42, 1998].

L'élément *Entity* est la super-entité qui regroupe toutes les autres entités du modèle d'ontologie. *Ontology* est l'entité qui modélise les différentes ontologies. L'élément *Concept* permet de modéliser les éléments qui font partie des ontologies. De l'entité *Concept*, deux entités filles sont dérivées : *Class* et *Property*. *Class* permet de modéliser les concepts des ontologies et *Property* les relations entre ces concepts. Les *Property* sont d'un type de données défini *DataType*. De l'entité *Class*, l'entité *DomainOntologyConcept* est dérivée pour modéliser les concepts d'ontologies de domaine.

Ces éléments permettent de représenter les concepts définis dans l'exemple d'ontologie précédent. Nous avons montré comment nous avons représenté un modèle de données et une ontologie dans notre proposition, en utilisant une approche de méta-modélisation. Nous présentons maintenant le mécanisme utilisé pour faire le lien entre ces deux éléments.

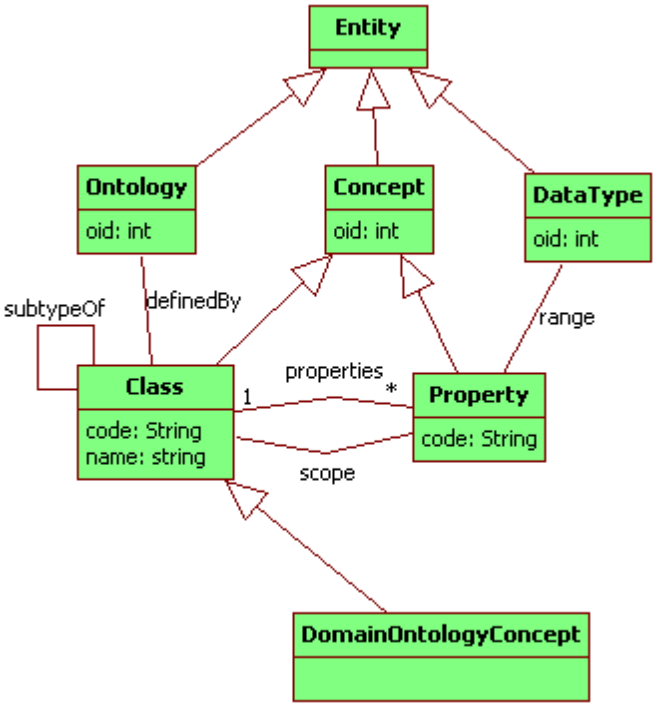


FIG. 4.4 – Modèle d'ontologie de domaine.

2.3 Indexation des modèles de données

Pour enrichir la description fournie par un modèle de données, nous proposons de l'indexer par des concepts d'une ontologie de domaine (voir la partie (1)(c) de la Figure 4.1). L'indexation des modèles de données permet ainsi l'enrichissement sémantique de ces modèles en vue d'une exploitation plus complexe.

Dans [Mastella, 2010], l'objectif est de remédier au manque de connaissances explicites dans les domaines de l'ingénierie. Une approche d'*annotation sémantique* est développée pour assigner des significations explicites aux objets identifiés par les expert dans les modèles de données. Par ailleurs, l'annotation dans l'approche décrite est considérée comme une entité de niveau supérieur, indépendante du méta-modèle de données. Ainsi, il est proposé de capturer les structures données et de les exprimer suivant le méta-modèle défini (voir figure 4.2). Les ontologies de domaine qui représentent la sémantique d'un domaine de connaissance particulier sont représentées suivant un modèle d'ontologie particulier. Enfin, il est proposé un modèle d'annotation pour permettre d'enrichir sémantiquement les instances des modèles de données par les instances ontologiques. La figure 4.5 montre le méta-modèle d'annotation proposé et le lien entre le méta-modèle de données ainsi que le modèle d'ontologie construits.

Dans la partie gauche de la figure, l'entité *ClassConstruct* et ses entités filles représentent un modèle simplifié d'ontologie qui permet de représenter des concepts d'ontologies de domaine.

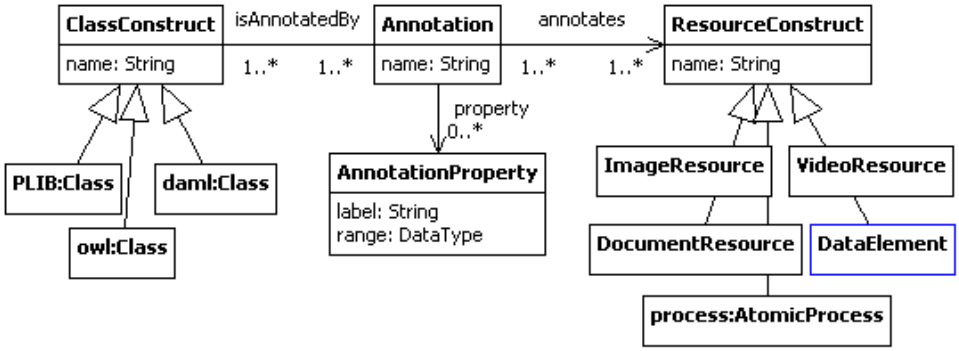


FIG. 4.5 – Méta-modèle d'annotation et la liaison entre un méta-modèle de données et un modèle d'ontologie de domaine.

Dans la partie droite de la figure, l'entité *ResourceConstruct* et ses entités filles représentent les primitives des ressources : documents, vidéos, images, et celles des modèles de données (l'entité *DataElement*). Dans le centre de la figure, l'entité *Annotation* crée un lien entre la primitive des concepts d'ontologies et la primitive des concepts de ressource (particulièrememnt la primitive *DataElement*) au moyen des relations *annotates* et *isAnnotatedBy*. Ces relations ont une cardinalité multiple, ce qui signifie qu'un élément d'annotation peut annoter des entités de modèles de données multiples avec un même concept d'ontologie, ou employer des multiples concepts d'ontologie pour annoter un même élément de modèle, ou encore que ces deux cas de figure peuvent se produire simultanément.

L'approche présentée par [Mastella, 2010] suggère d'isoler les annotations des modèles de données à annoter. Notre approche suggère également une indépendance entre les modèles de données et leur indexation pour les enrichir sémantiquement sans les modifier. Néanmoins, nous proposons d'indexer les modèles de données par les concepts d'ontologies de domaine en instanciant ces concepts. La figure 4.6 montre comment un modèle de facture électronique est indexé par le concept *ElectronicInvoice* de l'ontologie de la documentation en le rendant instance de ce concept.

Dans la figure, l'élément *Document* est le super concept qui généralise tous les autres concepts de l'ontologie de documentation. Il a comme propriétés *uri*, *name* et *url* qui renseignent respectivement sur l'identifiant unique du modèle de document, le nom du modèle de document et le chemin physique où se trouve la description du modèle de document. Ainsi, chaque concept de cette ontologie de documentation, qu'il s'agisse du concept *Document* ou de ses sous-concepts, indexe plusieurs modèles de données. Par exemple, le modèle de données qui représente la structuration des factures des auto-entrepreneurs possède l'uri "uri27" et sa description se trouve à l'adresse "//MDFact.xsd". Ce modèle de facture est une instance du concept *ElectronicInvoice* et est donc sémantiquement indexé par lui.

L'avantage de notre approche se trouve dans l'exploitation de cette description en utilisant la subsomption. Par exemple, des modèles de facture peuvent être établis et seront soit indexés

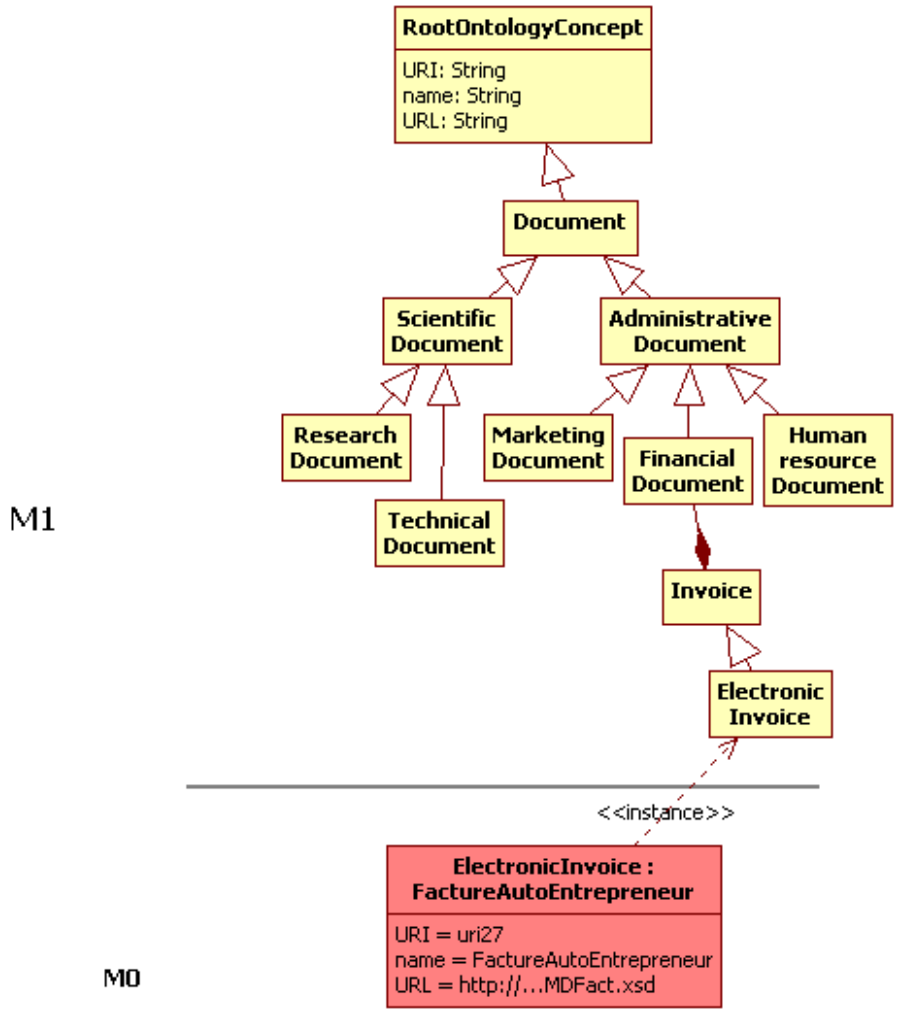


FIG. 4.6 – Indexation sémantique d'un modèle de facture électronique par un concept ontologique.

par le concept *Invoice* ou bien par son concept fils *Electronic Invoice* suivant que ces modèles représentent des factures ou bien, plus précisément, des factures électroniques. La recherche sémantique sur le concept *Invoice* donne comme résultat tous les modèles qu'ils soient indexés par le concept *Invoice* ou bien par son concept fils *Electronic Invoice* en vertu de la relation de subsomption. Une telle recherche est plus difficilement réalisable et maintenable dans l'approche proposée par [Mastella, 2010]. De plus, notre objectif est d'indexer les modèles de données qui sont référencés par les descriptions des services et des workflows informatiques (voir section 3.1). Cependant, les deux approches sont complémentaires lorsqu'il s'agit d'indexer les modèles de données et de pouvoir indexer également les instances de ces modèles de données.

Il est possible pour le même concept d'une ontologie d'indexer de multiples modèles de données. C'est le cas lorsque plusieurs modèles de données peuvent représenter le même concept

d'ontologie de domaine. Par exemple, un autre modèle de description de facture électronique pour les auto-entrepreneurs peut être formalisé. Dans ce cas, le même concept *ElectronicInvoice* indexerait ce modèle. Il est également possible pour plusieurs concepts d'indexer le même modèle de données. C'est le cas lorsqu'un modèle est assez générique pour représenter des concepts ontologiques différents. Par exemple, un modèle qui représente plusieurs points dans l'espace peut être indexé par plusieurs concepts ontologiques : *Cube*, *Pyramide*, etc.

3 Enrichissement sémantique des services et des workflows informatiques

Dans cette section (voir la colonne (2) dans la figure 4.1), il s'agit de représenter les descriptions des services et des workflows informatiques qui manipulent les modèles de données définis précédemment, ainsi que leur définition sémantique. C'est la partie concernant les données dynamiques de notre approche.

3.1 Services informatiques

Les services et les workflows informatiques (voir la partie (2)(a) dans la Figure 4.1) sont décrits dans des langages standards tels que WSDL pour la description de services Web ou bien BPEL et XPDL pour la description de compositions ou d'orchestrations de services informatiques en workflows. Leurs paramètres d'entrée et de sortie sont définis dans des documents de type XML. Dans leurs descriptions, des paramètres décrivent leurs emplacements, leurs langages de description, leurs entrées et sorties identifiées par des URI qui correspondent à des fichiers XML-Schema.

Les descriptions de services et de workflows informatiques sont avant tout des modèles qui décrivent de façon uniforme les exécutions de services et de workflows. Ces modèles de descriptions sont eux-mêmes conformes à des méta-modèles généralement standard de services et de workflows. Ces méta-modèles définissent l'ensemble minimal des entités et des attributs pour l'échange de définitions de services et de workflows informatiques. Par exemple, nous présentons le méta-modèle de processus défini par [Workflow Management Coalition, 2008] dans la figure 4.7.

Dans la figure, *Process* et *Activity* correspondent respectivement à la modélisation de workflows et de services. *Process* est un conteneur pour le workflow lui-même et fournit des informations associées à l'administration (par exemple une date de création ou l'auteur) ou pour être utilisés au cours de l'exécution des processus (par exemple le paramètres d'initialisation à utiliser, la priorité d'exécution, les délais à l'enregistrement, la personne à prévenir ou les informations de simulation). Un *Process* se compose d'une ou plusieurs *Activity*, chacune comprenant une unité d'exécution logique. L'entité *Transition* représente la séquence entre deux

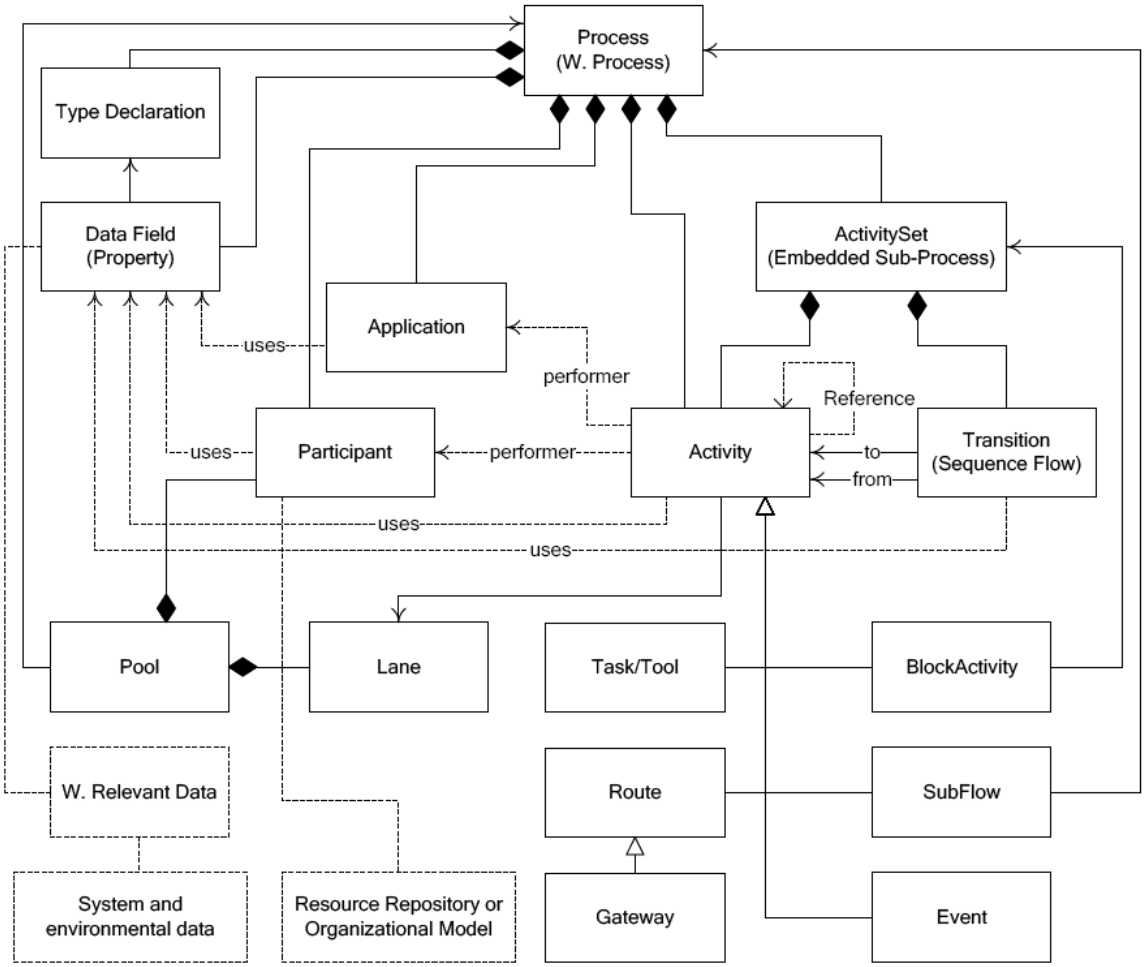


FIG. 4.7 – Méta-modèle de processus défini en XPDL.

Activity. L'entité *Pool* peut être considérée comme la représentation d'une organisation ou une entreprise et agit en tant que conteneur pour les objets *Activity* et *Transition*. L'entité *Lane* permet de subdiviser des *Pool* et peut être utilisée pour représenter des fonctions particulières au sein d'une même organisation ou entreprise.

Ce modèle permet de représenter n'importe quel service ou workflow informatique. Cependant, la description fournie ne permet pas de définir des hiérarchies et des comparaisons de services. Pour y remédier, nous proposons de les indexer par des ontologies de services. Cette approche est ainsi duale à celle suivie pour les modèles de données.

3.2 Ontologies de services

Les ontologies de services (voir la partie (2)(b) dans la Figure 4.1) sont les caractérisations sémantiques en termes de concepts des services et workflows dans un domaine de connaissance particulier. Ces ontologies définissent des hiérarchies et des compositions de services. Notre

approche ne consiste ainsi pas à envelopper des services par des annotations ni à étendre des langages de description de services et de workflows informatiques, mais à définir des ontologies de services indépendamment de tout langage de description et sans modification des descriptions.

Nous appelons *service sémantique*, un concept d'une ontologie de services. Un service sémantique est indépendant de toute implémentation. Par exemple, l'ontologie que nous présentons dans la figure 4.8 (partie de droite) représente un fragment d'une ontologie de services construite à des fins didactiques sur l'achat de pièces *Part*.

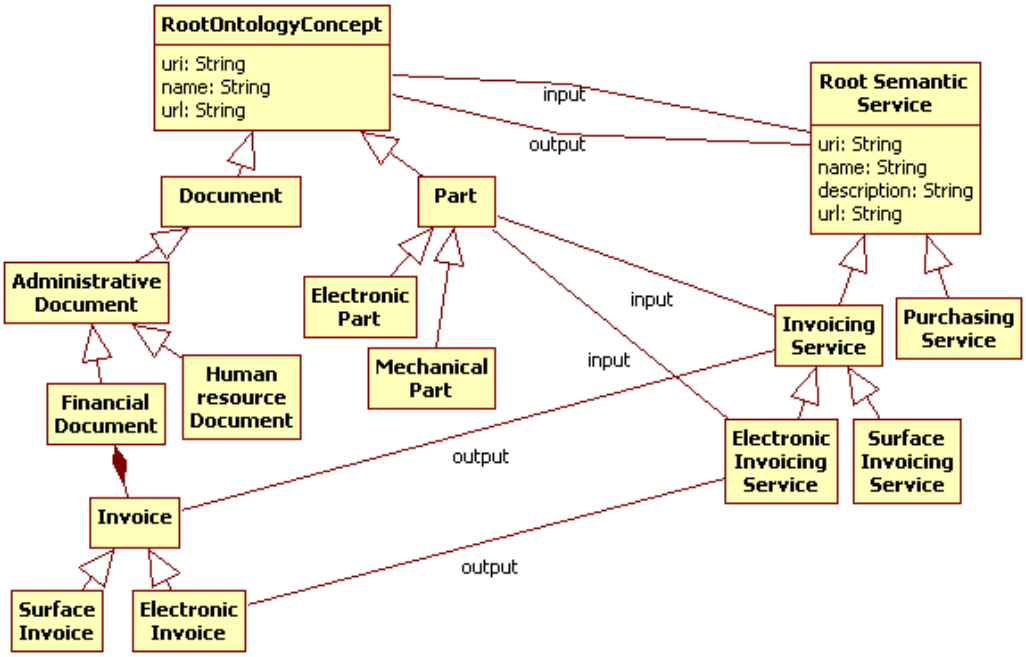


FIG. 4.8 – Une ontologie de services sur l'achat des pièces.

Dans cette ontologie de services, nous représentons un service sémantique racine *Root Semantic Service* qui peut être soit un service de paiement *Purchasing Service* qui concerne les services d'achat, soit un service de facturation *Invoicing Service*. Un service de facturation est plus précisément soit un service de facturation papier *Surface Invoicing Service* soit un service de facturation électronique *Electronic Invoicing Service*.

Les services sémantiques qui forment les concepts de cette ontologie de services référencent des concepts d'ontologies de domaines. Par exemple, le service sémantique de facturation a comme paramètre d'entrée une pièce *Part* et retourne en sortie une facture *Invoice* (voir la figure 4.8). Autre exemple, le service sémantique de facturation électronique a comme paramètre d'entrée une pièce *Part* et rend en sortie une facture électronique *Electronic Invoice*.

La construction d'une ontologie de services est un processus similaire à la construction d'une ontologie de domaine classique. Il s'agit de caractériser d'une manière consensuelle les

services et les workflows informatiques d'un domaine cible. Des différences, dues aux caractéristiques spécifiques des services, sont néanmoins à l'origine d'une modélisation particulière de ce type d'ontologies. En effet, nous avons montré que les services sémantiques référencent d'autres entités sémantiques comme les concepts d'ontologies de domaine pour définir leur entrée et sortie. Ils se référencent entre eux pour exprimer des relation d'antécédence. Par exemple, un service sémantique d'achat est prérequis à un service sémantique de facturation, etc. Ces propriétés sont partagées entre les services sémantiques (entrée, sortie, services prérequis, etc.). Les ontologies de services nécessitent bien une modélisation particulière. Nous avons donc conçu un modèle plus adéquat pour représenter les concepts de ces ontologies, c'est-à-dire les services sémantiques. Nous représentons le modèle que nous avons conçu sur la figure 4.9.

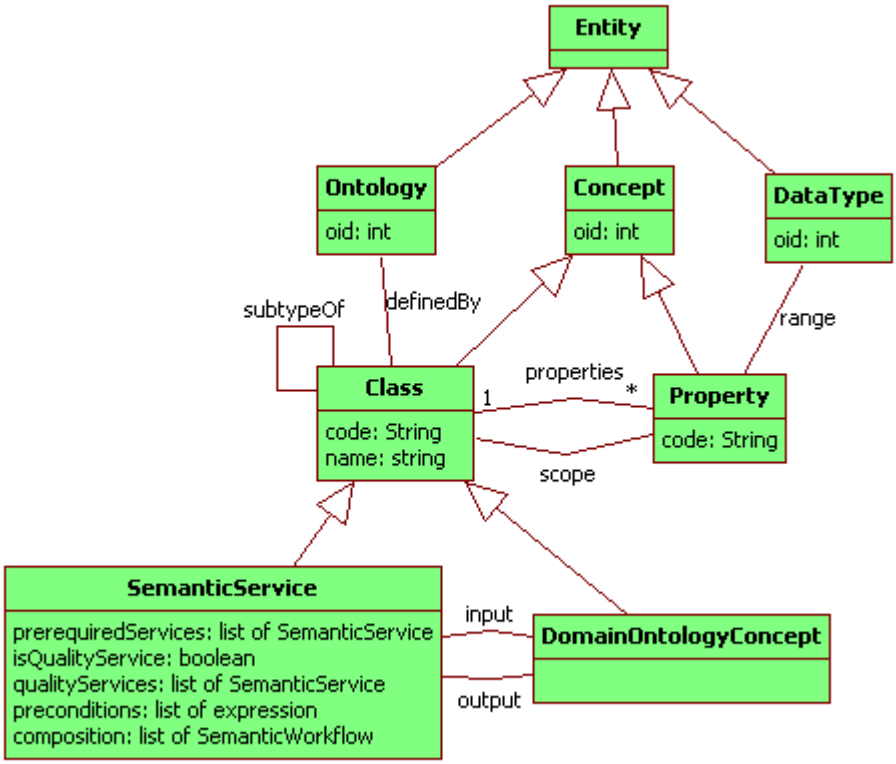


FIG. 4.9 – Extension du modèle d'ontologie pour la prise en compte de services sémantiques.

Les services sémantiques sont caractérisés par :

1. une liste d'entrées sémantiques qui référence des concepts d'ontologie de domaine ;
2. une liste de sorties sémantiques qui référence des concepts d'ontologie de domaine ;
3. une liste de services sémantiques pré-requis ;
4. une liste de services sémantiques de contrôle de qualité ;
5. une liste d'expressions représentant des préconditions ;

6. une liste de compositions correspondant aux différents workflows sémantiques (voir la section 4) ;

7. le lien hiérarchique entre les services sémantiques.

Dans cette section, nous montrons comment représenter des services sémantiques. Ils permettent d'enrichir la description syntaxique fournie par un service informatique. Cet enrichissement est réalisé en utilisant la technique d'indexation présentée dans la partie suivante.

3.3 Indexation de services et de workflows informatiques

L'objectif est d'indexer les descriptions de services et de workflows informatiques par des services sémantiques (voir la partie (2)(c) dans la Figure 4.1). L'indexation des services et des workflows informatiques permet l'enrichissement sémantique de ces descriptions en vue d'une exploitation plus complexe.

Notre approche suggère d'indexer les services et les workflows informatiques de la même manière que nous avons procédé à l'indexation des modèles de données. Elle implique une indépendance entre les descriptions des services et des workflows informatiques et leur indexation pour les enrichir sémantiquement sans les modifier. Nous proposons d'indexer et ainsi de caractériser sémantiquement les services et les workflows informatiques par des services sémantiques en instanciant ces derniers. Par exemple, un service particulier de facturation développé à l'intérieur d'une entreprise peut être un service Web qui porte le nom *FacturationAuto*. S'il est considéré comme réalisant sémantiquement une action de facturation électronique, il peut donc être indexé par le service sémantique *Electronic Invoice Service*. La figure 4.10 montre comment un service Web est indexé par le service sémantique *Electronic Invoicing Service* de l'ontologie de service d'achat de pièces en le rendant instance de ce concept.

Cette façon d'indexer les services et workflows informatiques par des services sémantiques garde bien une stricte séparation entre la description syntaxique des services et des workflows informatiques, utile pour leur exécution, et la description sémantique, utile pour leur localisation, leur découverte, etc.

Il est possible pour le même service sémantique d'indexer de multiples descriptions de services et de workflows. Ce cas de figure correspond à l'implémentation multiple du même service sémantique. Par exemple, plusieurs implémentations réalisant une facturation électronique seraient indexées par le même service sémantique *Electronic Invoicing Service*. Réciproquement, il est aussi possible pour plusieurs services sémantiques d'indexer un unique service ou workflow informatique qui serait générique. Par exemple, un service Web qui calcule une moyenne arithmétique peut être indexé par un service sémantique de moyenne de température ou de moyenne de pression atmosphérique.

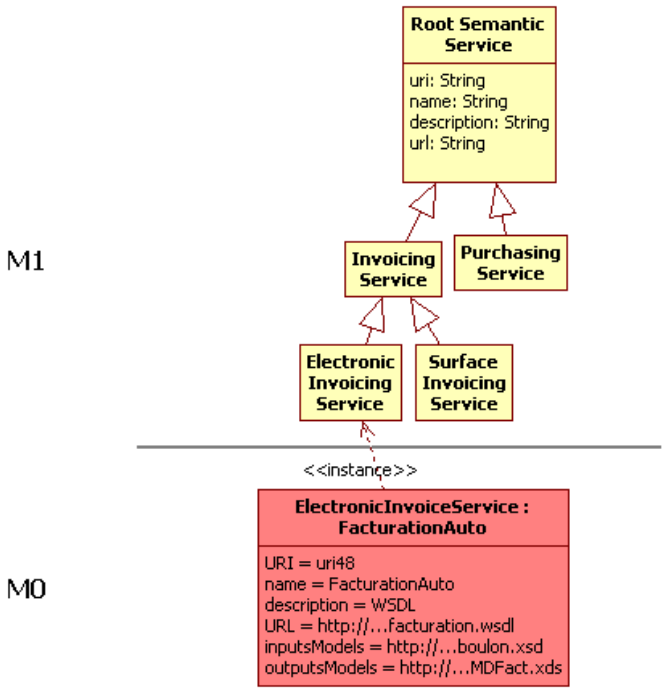


FIG. 4.10 – Indexation sémantique d'un service Web par le service sémantique *Electronic Invoicing Service*.

3.4 Modèle conceptuel global

Dans la figure 4.1, nous avons présenté notre approche pour l'indexation sémantique des modèles de données, des services et des workflows informatiques. Ayant présenté les différents modèles, ontologies et modèles d'ontologies, nous présentons et résumons, dans la figure 4.11, une modélisation plus détaillée de notre approche sur l'exemple qui nous a servi à l'expliquer.

Ainsi, dans la couche méta-modèle (M2), nous montrons le modèle d'ontologie enrichi pour pouvoir décrire ses instances qui sont les ontologies de domaine et de services dans la couche d'ontologie (M1). Enfin, les instances des concepts d'ontologies de domaine sont les modèles de données qu'ils indexent et les instances des concepts d'ontologies de services sont les descriptions de services et de workflows informatiques qu'ils indexent. Ils sont représentés dans la couche des instances (M0).

3.5 Contrôle sur les ontologies de services

Dans le modèle d'ontologie enrichi présenté sur la figure 4.9, des relations lient les services sémantiques qui sont des concepts des ontologies de services et les concepts des ontologies de domaine. En effet, les services sémantiques consomment en entrée et produisent en sortie

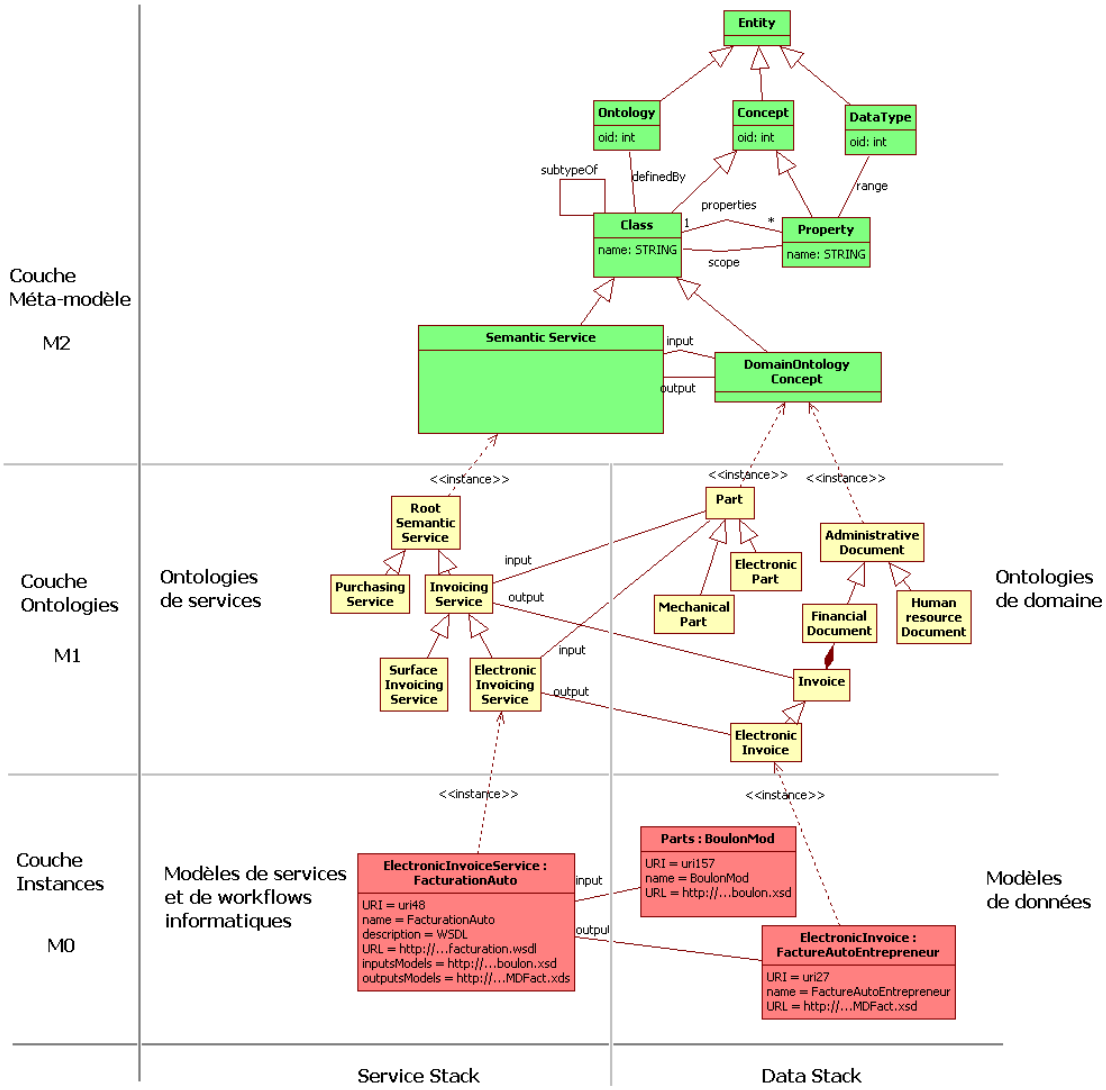


FIG. 4.11 – Modèle complet de l'approche à base ontologique pour la gestion sémantique de workflows.

des concepts d'ontologie de domaine. Toutefois, des contraintes doivent être rajoutées pour assurer la cohérence de certains liens entre les services sémantiques. Par exemple, si un service sémantique dans une ontologie de service subsume un autre service sémantique, les entrées et les sorties du premier service sémantique subsument respectivement les entrées et les sorties du second service sémantique. Une telle relation est appelée relation de covariance [Castagna et al., 1995] [Castagna, 1995]. Les services sémantiques pouvant être considérés comme des fonctions qui transforment des entrées en sorties, nous définissons la relation suivante de sumbsomption basée sur la covariance :

Si le service sémantique SS_i défini comme suit :

$$SS_i : DOC_{i,1} \times DOC_{i,2} \times ... \times DOC_{i,n} \longrightarrow DOC'_{i,1} \times DOC'_{i,2} \times ... \times DOC'_{i,n}$$

subsume le service sémantique SS_j défini comme suit :

$$SS_j : DOC_{j,1} \times DOC_{j,2} \times ... \times DOC_{j,n} \longrightarrow DOC'_{j,1} \times DOC'_{j,2} \times ... \times DOC'_{j,n}$$

alors

$$DOC_{i,1} \subseteq DOC_{j,1}, DOC_{i,2} \subseteq DOC_{j,2}, ..., DOC_{i,n} \subseteq DOC_{j,n}$$

et

$$DOC'_{i,1} \subseteq DOC'_{j,1}, DOC'_{i,2} \subseteq DOC'_{j,2}, ..., DOC'_{i,n} \subseteq DOC'_{j,n}$$

Avec :

– SS : SemanticService ;
– DOC : DomainOntologyConcept.

Nous illustrons la covariance que nous avons formalisée précédemment à travers l'ontologie des services que nous montrons dans la figure 4.8. Le service sémantique de facturation *Invoice Service* subsume le service sémantique *Electronic Invoice Service*. Cela implique bien que l'entrée sémantique du service sémantique *Invoice Service*, c'est-à-dire *Parts*, subsume l'entrée sémantique du service sémantique *Electronic Invoice Service*, c'est-à-dire *Parts*. De même, cela implique aussi bien que la sortie sémantique du service sémantique *Invoice Service*, c'est-à-dire *Invoice*, subsume la sortie sémantique du service sémantique *Electronic Invoice Service*, c'est-à-dire *Electronic Invoice*.

Dans la prochaine section, nous abordons les aspects liés à la construction de nouveaux workflows informatiques en se basant sur l'approche d'indexation que nous avons définie.

4 Construction de nouveaux workflows

L'indexation sémantique de workflows doit être distinguée de la construction de workflows sémantiques. Dans le premier cas, l'objectif est d'indexer des workflows existants construits par composition des services informatiques afin de simplifier leur éventuelle recherche en vue d'une utilisation future potentielle. Les workflows existants peuvent être indexés par les *services sémantiques* qui sont des concepts d'ontologies de services. Dans le second cas cependant, l'objectif est de simplifier la construction de nouveaux workflows en construisant d'abord des *workflows sémantiques*. La construction de workflows sémantiques se fait en composant des services sémantiques. Une telle tâche est naturelle pour les experts d'un domaine donné puisque

la sémantique des services est déjà explicitée. Composer des services sémantiques consiste à définir l'ordre d'exécution de services ou workflows informatiques déjà construits en vue de réaliser des workflows informatiques plus complexes.

Dans ce qui suit, nous montrons comment nous décrivons des workflows sémantiques. Nous exposons également dans quelle mesure la composition de services sémantiques peut être contrôlée ou vérifiée. Ensuite, nous montrons comment l'ontologie de services est enrichie par l'ajout du workflow sémantique construit en tant que service sémantique. Enfin, nous expliquons comment un workflow informatique, c'est-à-dire une implémentation de workflow sémantique, peut être obtenue.

4.1 Description de workflows sémantiques

Pour pouvoir représenter des workflows sémantiques, nous avons défini un ensemble d'opérateurs de composition décrit dans le modèle de description de workflows. Ce modèle est représenté dans la Figure 4.12.

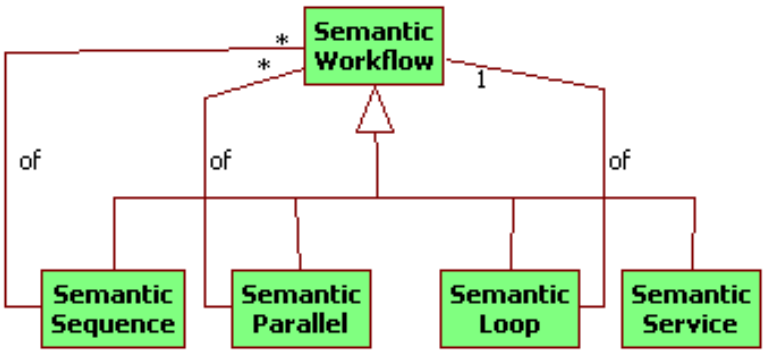


FIG. 4.12 – Modèle de description de workflows sémantiques.

Un workflow sémantique peut être dérivé en service composite sémantique :
- *SemanticSequence* : représente un déroulement en séquence de deux ou plusieurs services ou workflows sémantiques ;
- *SemanticParallel* : représente deux ou plusieurs services ou workflows sémantiques qui se déroulent en parallèle ;
- *SemanticLoop* : représente un service ou un workflow sémantique qui se déroule en boucle ;
- *SemanticService* : représente un service sémantique (atomique).

Un tel modèle permet la description d'un workflow sémantique.

Par exemple, nous montrons dans la figure 4.13 la description d'un achat de pièces et une double facturation papier et électronique.

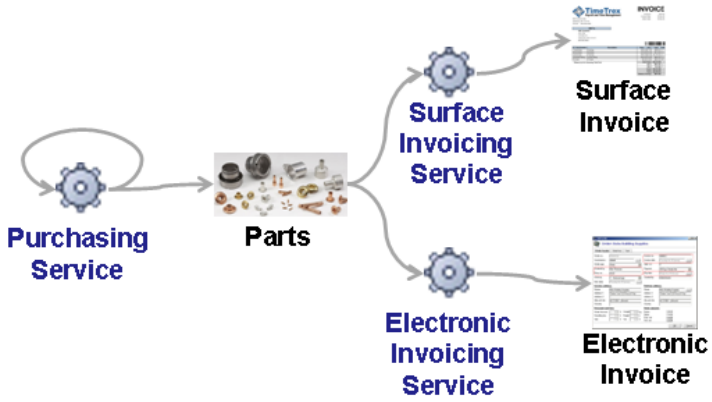


FIG. 4.13 – Un workflow sémantique sur un achat de pièces et une double facturation.

Dans la figure, le service sémantique *Purchasing Service* est effectué en boucle pour l'achat de plusieurs pièces. Ensuite, en parallèle, les services sémantiques de *Surface Invoicing Service* et *Electronic Invoicing Service* sont effectués pour une facturation papier ainsi qu'une facturation électronique.

Par ailleurs, la figure 4.14 représente le workflow sémantique 4.13 en tant qu'instance du modèle de description de workflows sémantiques montré dans la figure 4.12.

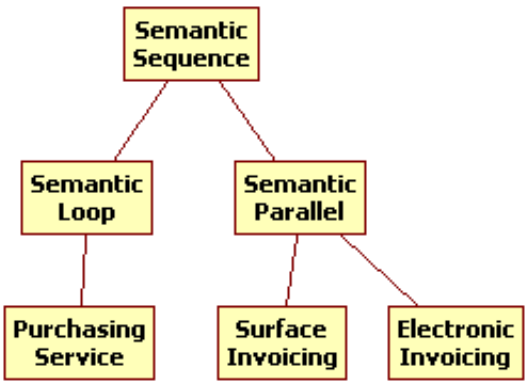


FIG. 4.14 – Description du workflow sémantique dans le modèle défini.

4.2 Composition fonctionnelle dans les workflows sémantiques

Nous avons montré dans les sections précédentes que les services sémantiques sont définis principalement à travers des propriétés fonctionnelles (ainsi que des propriétés non-fonctionnelles qui ne sont pas l'objet de cette thèse mais dont certains résultats sont publiés dans les travaux de [Tondello et Siqueira, 2008], [Kritikos et Plexousakis, 2007] et [Zhou et al., 2004]).

La composition des services sémantiques se fait par un expert du domaine dans lequel les services sémantiques ont été définis. Elle implique que les services sémantiques communiquent à travers leurs entrées et leurs sorties sémantiques. Toutefois, la validation de la composition de services sémantiques n'est pas triviale, et spécialement au niveau sémantique (la validation de la composition n'est pas l'objet de cette thèse). Des résultats sur la validation syntaxique sont néanmoins l'objet d'étude d'une thèse complémentaire [Ait-Sadoune et Ait-Ameur, 2009] [Ait-Sadoune et Ait-Ameur, 2010].

Certaines vérifications sont cependant trivialement décidables. C'est le cas des services sémantiques de contrôle de qualité. Ainsi, si un service sémantique requiert un service sémantique de contrôle de qualité préalable, un tel service doit se trouver directement en séquence avant le service sémantique. Un autre exemple concerne les services sémantiques pré-requis. Ainsi, si des services sémantiques sont pré-requis au déroulement d'un service sémantique donné, il est possible de vérifier qu'ils sont bien effectués avant ce service dans la description du workflow sémantique, ou bien de les calculer.

4.3 Évolution de l'ontologie des services

Losqu'un workflow sémantique est construit, deux possibilités peuvent se présenter. La première possibilité est que conceptuellement, le workflow sémantique est nouveau et devient donc à son tour un service sémantique. Il est ensuite possible d'enrichir l'ontologie de services avec un nouveau service sémantique.

Lorsqu'un workflow sémantique est ajouté en tant que service sémantique, ses propriétés fonctionnelles telles que les entrées sémantiques, les sorties sémantiques et les services sémantiques pré-requis doivent être calculées. Par exemple, si le workflow sémantique décrit sur la figure 4.13 représentait un concept nouveau de service, il serait ajouté en tant que service sémantique pour enrichir l'ontologie des services de la figure 4.8. Ils n'aurait aucune entrée sémantique et aurait comme sortie une facture papier représentée par le concept ontologique *Surface Invoice* et une facture électronique représentée par le concept ontologique *Electronic Invoice*.

Si le workflow sémantique correspond conceptuellement déjà à un service sémantique, ce dernier est modifié et enrichi par une autre description de composition sémantique. Par exemple, si le workflow sémantique décrit dans la figure 4.13 correspondait déjà à un service sémantique dans l'ontologie des service de la figure 4.8, la description de ce service sémantique serait enrichie pour prendre en compte le fait qu'il peut aussi représenter une composition de plusieurs services sémantiques.

4.4 Construction et génération de workflows informatiques

Quand un workflow sémantique a été construit, il est possible de "déduire" un workflow informatique. Cela peut être fait en sélectionnant un service ou un workflow informatique pour chaque service sémantique qui compose le workflow sémantique. Des critères de qualité de services peuvent être utilisés lors de cette sélection.

Bien que les services sémantiques composés pour obtenir un workflow sémantique forment une composition cohérente, cela n'implique pas que les services informatiques sélectionnés sont composables pour obtenir un workflow informatique.

Dans ce cas, des services informatiques d'adaptation qui représentent principalement des services de *conversion* de modèles de données sont introduits dans le workflow informatique pour rendre la composition cohérente. Par exemple, si dans un workflow sémantique, deux services sémantiques sont en séquence et qu'ils se transmettent une facture électronique, les services ou workflows informatiques qui leurs correspondent peuvent ne pas être compatibles car ils manipulent (l'un produit et l'autre consomme) des factures électroniques qui n'ont pas le même format. Ainsi, il faut pouvoir intercaler en séquence, entre les deux services ou workflows informatiques, un service informatique de conversion de format.

5 Exploitation de l'approche

Notre approche permet de représenter les modèles de données, les services et workflows informatiques ainsi que les concepts ontologiques et services sémantiques qui en définissent le sens et qui les indexent. Pour permettre une exploitation de cet ensemble de données en assurant la scalabilité de l'approche, une structure de base de données particulière doit être utilisée. Cette structure doit permettre de stocker des ontologies, leurs instances et de faire évoluer le modèle d'ontologie. De plus, un langage de requêtes de cette structure doit être disponible. Ce langage permet la modification du modèle d'ontologie, la création de concepts ontologiques ainsi que leurs instances. Il permet la recherche, la mise à jour et la suppression pour faire évoluer les représentations. Plus particulièrement, il permettra d'écrire les requêtes suivantes :

- requêtes de découverte de services à partir de leurs caractérisations fonctionnelles : il est possible de rechercher des services ou des workflows informatiques indexés par un service sémantique particulier. De plus, il est possible de restreindre la recherche aux propriétés fonctionnelles qui vérifient des contraintes particulières. Par exemple, à une entrée ou sortie qui référence un concept particulier de l'ontologie de domaine ;
- requêtes de découverte de services de substitution : il est possible de rechercher des services ou des workflows qui remplissent un rôle similaire à un service ou workflow donné. La requête consiste à rechercher les services et workflows indexés par le même service sémantique qui indexe le service ou workflow donné, ou par un service sémantique qui le subsume ;

– requêtes de découverte de services à partir de leurs propriétés non-fonctionnelles : il est possible de rechercher des services ou des workflows informatiques qui remplissent des conditions particulières telles que la disponibilité, la localité, la performance ou le concepteur.

Dans le prochain chapitre, nous montrons comment nous avons validé notre approche en l'implantant sur la structure de base de données à base ontologique OntoDB. Nous montrons comment nous avons modifié le modèle d'ontologie, créé des ontologies de domaine et de services et enfin indexé des modèles de données, des services et des workflows informatiques en vue d'une exploitation complexe.

6 Conclusion

Dans ce chapitre, nous avons introduit les modèles supports de l'approche que nous proposons pour une meilleure gestion de services et de workflows informatiques ainsi que des modèles de données qu'ils manipulent. Notre approche est basée sur la construction d'ontologies de domaine et de services et sur l'indexation de modèles de données, de services et de workflows informatiques par des concepts ontologiques.

La construction d'ontologies de domaine nécessite une modélisation complète et consensuelle d'un domaine particulier. De même, la construction d'ontologies de services nécessite la caractérisation des services et des workflows informatiques d'un domaine en termes de services sémantiques et de relations entre eux. A la fin de cette étape, il est possible d'indexer sémantiquement les modèles de données, les services et les workflows informatiques ce qui permet leur meilleure gestion. Ainsi, une telle indexation permet la réutilisation de services et de workflows informatiques déjà conçus. Plus précisément, elle permet par exemple la recherche de services et de workflows informatiques par leur sens et la recherche de services ou de workflows informatiques de substitution.

Il est également possible de créer de nouveaux workflows informatiques. Ainsi, à partir des concepts de l'ontologie des services, c'est-à-dire des services sémantiques, il est possible de construire des workflows sémantiques en composant les services sémantiques. Enfin, un workflow informatique est dérivé en faisant correspondre un service ou un workflow informatique à chaque service sémantique qui compose le workflow sémantique.

Dans le prochain chapitre, nous montrons comment notre approche est validée en l'implantant sur la structure de base de données à base ontologique OntoDB. Nous montrons comment nous y définissons les ontologies de domaine et de services. Nous montrons aussi comment nous y stockons des descriptions de modèles de données ainsi que des services et des workflows informatiques. Nous explicitons enfin les possibilités d'exploitation qui en émergent.

Chapitre 5

Modélisation logique de notre approche sur OntoDB

Sommaire

1 Introduction 109
2 Modélisation logique de données sur OntoDB 110
2.1 Création de concepts d'ontologie de domaine 110
2.2 Indexation de modèles de données 111
3 Modélisation logique de services sur OntoDB 112
3.1 Modèle d'ontologie de services sur OntoDB 113
3.2 Instanciation de services sémantiques 113
3.3 Indexation de services et de workflows informatiques 114
4 Extension globale d'OntoDB pour la gestion de services sémantiques 114
5 Exploitation du modèle logique sur OntoDB 116
5.1 Recherche de services sémantiques à travers leurs attributs sémantiques 118
5.2 Recherche de services sémantiques à travers des services et workflows informatiques indexés 119
5.3 Recherche de services ou de workflows informatiques 119
5.4 Recherche de services ou de workflows informatiques avec restriction sur les propriétés et attributs sémantiques 120
5.5 Adaptativité de services 121
5.5.1 Adaptativité parfaite 121
5.5.2 Adaptativité spécialisée 122
6 Extension d'OntoQL 123
7 Implantation des contraintes 124
7.1 Implantation de contraintes de covariance 124
7.2 Implantation de contraintes sur la relation entre données sémantiques et modèles de données 126

8 Conclusion 128

Résumé. Dans le chapitre précédent, nous avons présenté notre approche de façon conceptuelle. Dans ce chapitre, nous présentons l'adaptation de notre approche sur la structure de BDBO OntoDB que nous utilisons comme entrepôt sémantique.

1 Introduction

L'approche que nous proposons possède les caractéristiques suivantes :

- elle nécessite la gestion de trois niveaux de modélisation comme cela est décrit dans la section 3.4 du chapitre 4. D'abord, il y a le méta-modèle (M2) ou modèle d'ontologie pour définir les primitives permettant de créer les concepts de données et les services sémantiques. Ensuite, il y a le niveau (M1), c'est-à-dire le niveau ontologique où les concepts de données et les concepts de services sémantiques sont définis. Enfin, il y a le niveau instance (M0) dans lequel les modèles de données, les services et workflows informatiques sont définis en tant qu'instances respectivement des concepts de données et de services sémantiques ;
- elle comprend la gestion de deux types de modèles conceptuels. Il y a, d'une part, les modèles syntaxiques qui décrivent les structures de données, et d'autre part, les services informatiques et leurs compositions en workflows informatiques ;
- elle sépare la gestion de la partie statique, c'est-à-dire celle des données de la partie dynamiques, c'est-à-dire celle des services.

Pour valider notre approche, il est nécessaire de choisir une plateforme sur laquelle il sera possible d'implanter ces caractéristiques et de les exploiter. La structure de bases de données OntoDB permet la création et la gestion d'ontologies et de leurs instances. Elle permet également de modifier et d'enrichir le niveau méta-modèle ou modèle d'ontologie. Grâce à OntoDB, il est possible de modifier le méta-modèle et ainsi de permettre la gestion de services sémantiques. D'autre part, OntoDB permet de poser des requêtes au niveau ontologique de même qu'au niveau des instances ontologiques avec le langage de requêtes OntoQL. OntoDB nous permet donc, à travers ses caractéristiques, d'exploiter pleinement le pouvoir descriptif de notre approche.

Dans ce chapitre, nous expliquons d'abord comment nous créons les ontologies de domaine qui caractérisent les données de domaines particuliers et comment nous stockons ces représentations dans une base de données à base ontologique OntoDB. Nous montrons ensuite comment sont stockées en base les indexations des modèles de données par les concepts des ontologies de domaine. Nous expliquons également comment nous faisons évoluer le modèle d'ontologies pour permettre la modélisation de services sémantiques et la création d'ontologies de services dans OntoDB. Puis, nous montrons comment nous indexons les services et les workflows informatiques par les services sémantiques. Ensuite, nous détaillons comment nous pouvons créer des workflows sémantiques et les ajouter en tant que services sémantiques dans OntoDB. Nous montrons également comment nous exploitons toutes ces modifications, créations et insertions dans OntoDB pour répondre à une meilleure gestion des services et des workflows informatiques dans un domaine cible. Enfin, nous montrons les extensions du langage OntoQL pour la représentation de services sémantiques et mettrons en évidence les contraintes à vérifier pour assurer la cohérence des diverses représentations.

2 Modélisation logique de données sur OntoDB

L'objectif est de pouvoir représenter la partie statique, c'est-à-dire la partie des données sur OntoDB. Plus précisément, il s'agit de pouvoir représenter les modèles de données ainsi que les concepts ontologiques qui en définissent le sens dans la base de données à base ontologique OntoDB. Dans son état initial, OntoDB permet uniquement de créer des classes d'ontologies. Dans notre cas, nous devons représenter des classes pour indexer des modèles de données. Pour bien distinguer les classes des ontologies des classes indexant des modèles de données, nous avons choisi de spécialiser le constructeur (ou entité) de classe standard pour la création de concepts ontologiques *#DomainOntologyConcept*. La figure 5.1 montre comment la création de l'entité *#DomainOntologyConcept* au niveau du méta-modèle ou modèle d'ontologie est réalisée.

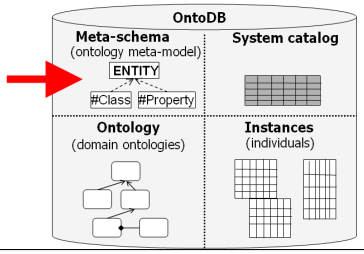
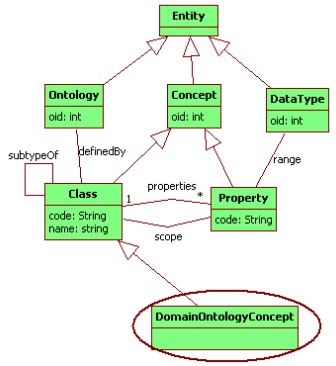
<table>
<tr><td>Requête de modification du modèle d'ontologie</td><td> 
</td></tr>
<tr><td>CREATE ENTITY #DomainOntologyConcept UNDER #Class ()</td><td> 
</td></tr>
</table>

FIG. 5.1 – Requête OntoDB pour la création de concepts ontologiques.

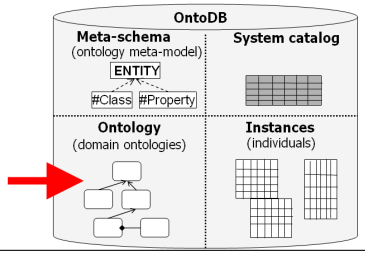
Une fois ce nouveau constructeur ajouté, nous pouvons créer des concepts d'ontologies de domaine par des création de classes de type *#DomainOntologyConcept*. Enfin, les modèles de données sont ajoutés en tant qu'instances de ces concepts en instanciant les classes créées sur OntoDB. Ceci permet d'indexer le modèle de données par une classe d'ontologie de domaine.

2.1 Création de concepts d'ontologie de domaine

La création des concepts d'ontologie de domaine se fait en instanciant des classes du modèle d'ontologie dans OntoDB. Ces classes sont de type *#DomainOntologyConcept*.

Par exemple, nous montrons la création du concept de facture électronique *Electronic Invoice* sur la figure 5.2. Lors de cette création, nous attribuons des propriétés au concept de facture électronique. A chaque fois qu'une instance est ajoutée, il faut renseigner ses propriétés. Tout modèle ou format de données qui permet de représenter des factures est une instance potentielle du concept de facture électronique. La propriété *URI* permet de renseigner de manière unique une instance de ce concept. La propriété *name* renseigne sur le nom du format de description du modèle de données ajouté. La propriété *URL* permet de représenter le lien vers la description du modèle de données ajouté.

Requête d'enrichissement d'ontologies de domaine




```
CREATE #DomainOntologyConcept
RootOntologyConcept (
  PROPERTIES (
     URI STRING,
     name STRING,
     URL STRING))
...
CREATE #DomainOntologyConcept
ElectronicInvoice UNDER Invoice ()
...
CREATE EXTENT OF ElectronicInvoice (URI, name,
URL)
```

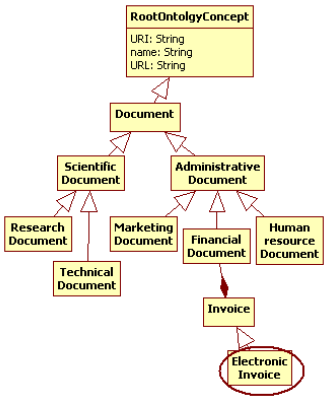


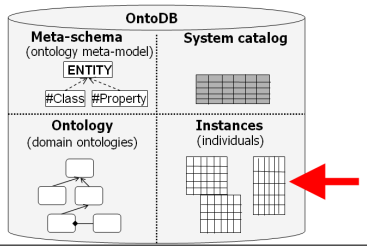
FIG. 5.2 – Exemple de création d'un concept d'une ontologie de domaine.

2.2 Indexation de modèles de données

L'indexation de modèles de données se fait en instanciant un concept d'une ontologie de domaine. En rendant un modèle de données instance d'un concept particulier, on annote le modèle sémantiquement.

Par exemple, puisqu'il existe un modèle de facture électronique pour les auto-entrepreneurs, nous pouvons indexer ce modèle de facture électronique par le concept de facture électronique *Electronic Invoice* de l'ontologie de documentation. Nous représentons cette indexation de ce modèle par son insertion en tant qu'instance du concept *Electronic Invoice* sur la figure 5.3.

Requête d'insertion d'instances d'ontologies de domaine (modèles de données)




```
INSERT INTO ElectronicInvoice (
    URI, name, URL)
 VALUES (
    'uri27', 'Facture Auto Entrepreneur',
'http://.../modele- officiel- facture- auto-
entrepreneur.pdf')
```

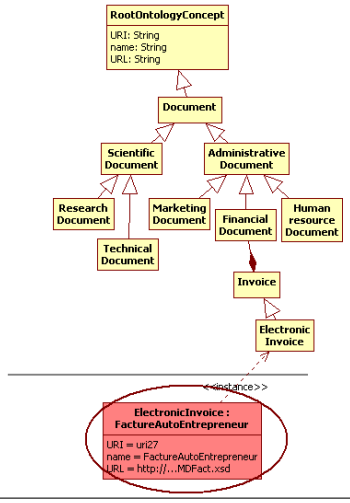


FIG. 5.3 – Exemple d'indexation d'un modèle de données par un concept ontologique.

Nous avons montré dans cette section comment nous avons représenté la partie correspondant aux données de notre approche. Nous présentons dans la section suivante la représentation de la partie correspondant aux services.

3 Modélisation logique de services sur OntoDB

Ayant représenté la partie statique de notre approche, nous nous intéressons à présent à la représentation de la partie dynamique, c'est-à-dire la partie des services et des workflows informatiques sur OntoDB. Plus précisément, il s'agit de pouvoir représenter les services et les workflows informatiques ainsi que les concepts d'ontologies de services qui en définissent le sens, c'est-à-dire des services sémantiques dans une base de données à base ontologique OntoDB.

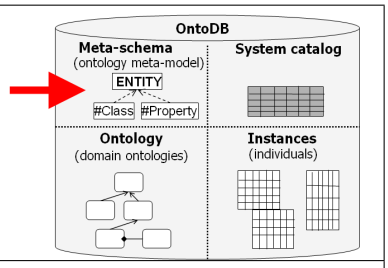
Nous avons déjà dit qu'une ontologie de service est une caractérisation particulière en ce sens que ses concepts ont des attributs communs particuliers. Dans ce qui suit, nous présentons donc d'abord l'enrichissement du modèle d'ontologie pour permettre la caractérisation des services sémantiques à travers l'entité *#SemanticService*. Ensuite, nous présentons la création des concepts d'ontologies de services par des création de classes de type *#SemanticService* sur OntoDB. Enfin, les services et workflows informatiques sont ajoutés en tant qu'instances des

services sémantiques créés sur OntoDB.

3.1 Modèle d'ontologie de services sur OntoDB

Les services sémantiques sont caractérisés par des valeurs de propriétés et peuvent être organisés dans une hiérarchie utilisant la relation de subsomption. Ils se comportent donc comme des classes de l'ontologie standard. Cependant nous souhaitons bien les distinguer des classes des ontologies. Aussi, pour permettre la création de services sémantiques, nous spécialisons la classe standard pour la création de concepts ontologiques en la classe *#SemanticService*. La figure 5.4 montre comment la création de l'entité *#SemanticService* au niveau du méta-modèle ou modèle d'ontologie est réalisée. Cette création est conforme au modèle de services sémantiques présenté sur la figure 4.9 du chapitre 4.

Requête de modification du modèle d'ontologie




```
CREATE ENTITY #SemanticService UNDER #Class (
  #inputs REF (#DomainOntologyConcept) Array,
  #outputs REF (#DomainOntologyConcept) Array,
  #prerequiredServices REF (#SemanticService)
Array,
  #qualityServices REF (#SemanticService)
Array,
  #preconditions REF (#Expression) Array,
  #composition REF (#SemanticWorkflow) Array)
```

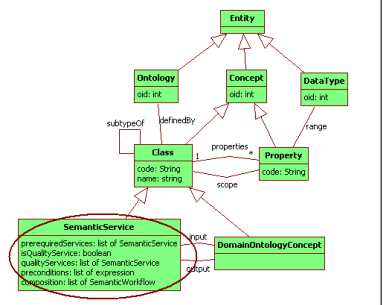


FIG. 5.4 – Requête OntoDB pour la création de services sémantiques.

3.2 Instanciation de services sémantiques

La création des concepts de l'ontologie de services se fait en créant des classes dans OntoDB. Ces classes sont des instances de *#SemanticService*.

Par exemple, nous montrons la création du service sémantique de facturation électronique *Electronic Invoicing Service* sur la figure 5.5. Il a comme attribut en entrée une pièce *Parts* et en sortie une facture *Electronic Invoice*. Lors de cette création, nous attribuons à ce service

sémantique des propriétés qui seront valuées lors de l'ajout d'instances, c'est-à-dire de services ou de workflows informatiques. A chaque fois qu'une instance est ajoutée, il faut renseigner ses propriétés. Tout service ou workflow informatique qui réalise une facturation électronique est une instance potentielle du service sémantique *Electronic Invoicing Service*. Il possède les propriétés suivantes :

- *URI* permet de renseigner de manière unique une instance de ce service sémantique ;
- *description* renseigne sur une description textuelle générale du service ou du workflow informatique ajouté ;
- *URL* permet de représenter le lien vers la description syntaxique du service ou du workflow informatique ajouté ;
- *inputsModel* permet de représenter le modèle de données référencé en entrée dans la description du service ou du workflow informatique. Cette propriété référence une liste de modèles de données par le concept ontologique racine *RootOntologyConcept* et qui subsume tous les autres ;
- *outputsModel* permet de représenter le modèle de données référencé en sortie dans la description du service ou du workflow informatique. Cette propriété référence une liste de modèles de données par le concept ontologique racine *RootOntologyConcept*.

3.3 Indexation de services et de workflows informatiques

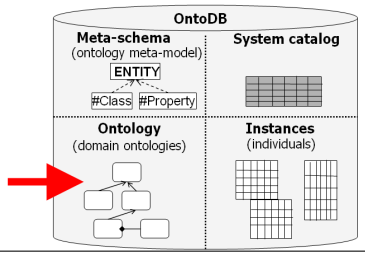
L'indexation de services et de workflows informatiques se fait en instanciant un concept d'une ontologie de service, c'est-à-dire de service sémantique. En rendant un service ou un workflow informatique instance d'un service sémantique particulier, on l'annote sémantiquement sans le surcharger.

Par exemple, un service informatique de facturation électronique peut être indexé par le service sémantique de facturation électronique *Electronic Invoicing Service*. Ceci permet d'enrichir ce service informatique par l'ensemble des informations décrites par le service sémantique comme par exemple la description de ses entrées et sorties, le fait que c'est un service de facturation particulier, etc. Nous représentons cette indexation de ce service informatique par son insertion en tant qu'instance du service sémantique *Electronic Invoicing Service* dans la figure 5.6.

4 Extension globale d'OntoDB pour la gestion de services sémantiques

Pour valider notre approche, nous devons stocker toutes les informations qui caractérisent les services et les workflows informatiques dans une base de données à base ontologique de type OntoDB. La figure 5.7 représente l'extension d'OntoDB que nous proposons par notre approche et qui permet de manipuler les modèles de données, les services et workflows informatiques,

Requête d'enrichissement d'ontologies de services




```
CREATE #SemanticService RootSemanticService (
 DESCRIPTOR (
    #input = ARRAY(
      SELECT doc.#oid
      FROM #DomainOntologyConcept doc
      WHERE doc.#name='RootOntologyConcept')
    #output = ARRAY(
      SELECT doc.#oid
      FROM #DomainOntologyConcept doc
      WHERE doc.#name='RootOntologyConcept')
 PROPERTIES (
    URI STRING,
    name STRING,
    description STRING,
    URL STRING,
    inputsModels REF (RootOntologyConcept)
ARRAY,
    outputsModels REF (RootOntologyConcept)
ARRAY))
...
CREATE #SemanticService
ElectronicInvoicingService (
 DESCRIPTOR (
    #input = ARRAY(
      SELECT doc.#oid
      FROM #DomainOntologyConcept doc
      WHERE doc.#name='Parts')
    #output = ARRAY(
      SELECT doc.#oid
      FROM #DomainOntologyConcept doc
      WHERE doc.#name='ElectronicInvoice'))
...
CREATE EXTENT OF ElectronicInvoicingService
(URI, name, description, URL, inputsModels,
outputsModels)
```

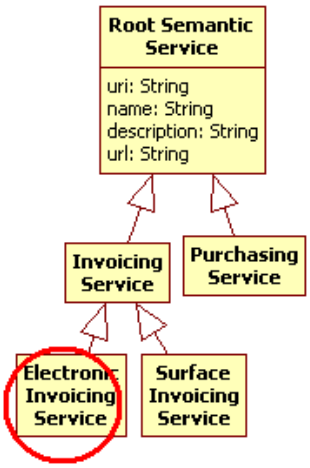


FIG. 5.5 – Exemple de création de service sémantique.

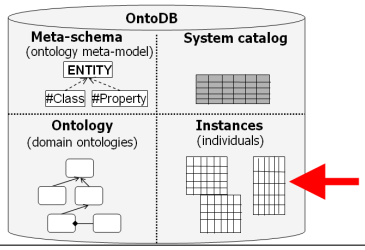
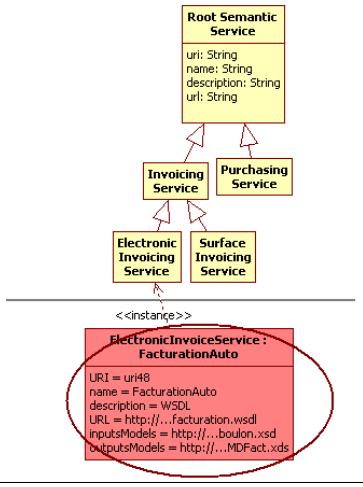
Requête d'insertion d'instances d'ontologies de service (services et workflows informatiques)	
INSERT INTO ElectronicInvoiceService (URI, name, description, URL, inputsModels, outputsModels) **VALUES** ('uri48', 'FacturationAuto', 'WSDL', 'http://.../facturation.wsdl ', (**SELECT** oid **FROM** RootOntologyConcept **WHERE** URL = 'http://.../boulon.xsd'),(**SELECT** oid **FROM** RootOntologyConcept **WHERE** URL = 'http://.../MDFact.xsd'))	

FIG. 5.6 – Exemple d'indexation d'un service informatique par un service sémantique.

les ontologies ainsi que les indexations. Cette extension est composée des principales parties suivantes :

- La figure 5.7.A stocke le modèle d'ontologie qui a été étendu et qui a été présenté dans la section 3.1 du chapitre précédent ;
- La figure 5.7.B stocke les ontologies de domaine et de services définies selon le modèle d'ontologie étendu. Des exemples d'ontologie de domaine et d'ontologie des services ont été donnés dans la section 2.2 du chapitre précédent ;
- La figure 5.7.C stocke les modèles de données qui sont des instances des concepts d'ontologies de domaine et stocke des services et des workflows informatiques qui sont des instances de services sémantiques, c'est-à-dire de concepts d'ontologies de services.

Le stockage de toutes ces informations dans un référentiel commun et unique permet de nouvelles fonctionnalités pour gérer des services informatiques, qui sont basées sur des requêtes sémantiques exposées dans la section suivante.

5 Exploitation du modèle logique sur OntoDB

Nous montrons dans cette section, l'intérêt de notre approche pour faciliter la gestion de services et de workflows informatiques. Pour cela, nous présentons les requêtes sémantiques

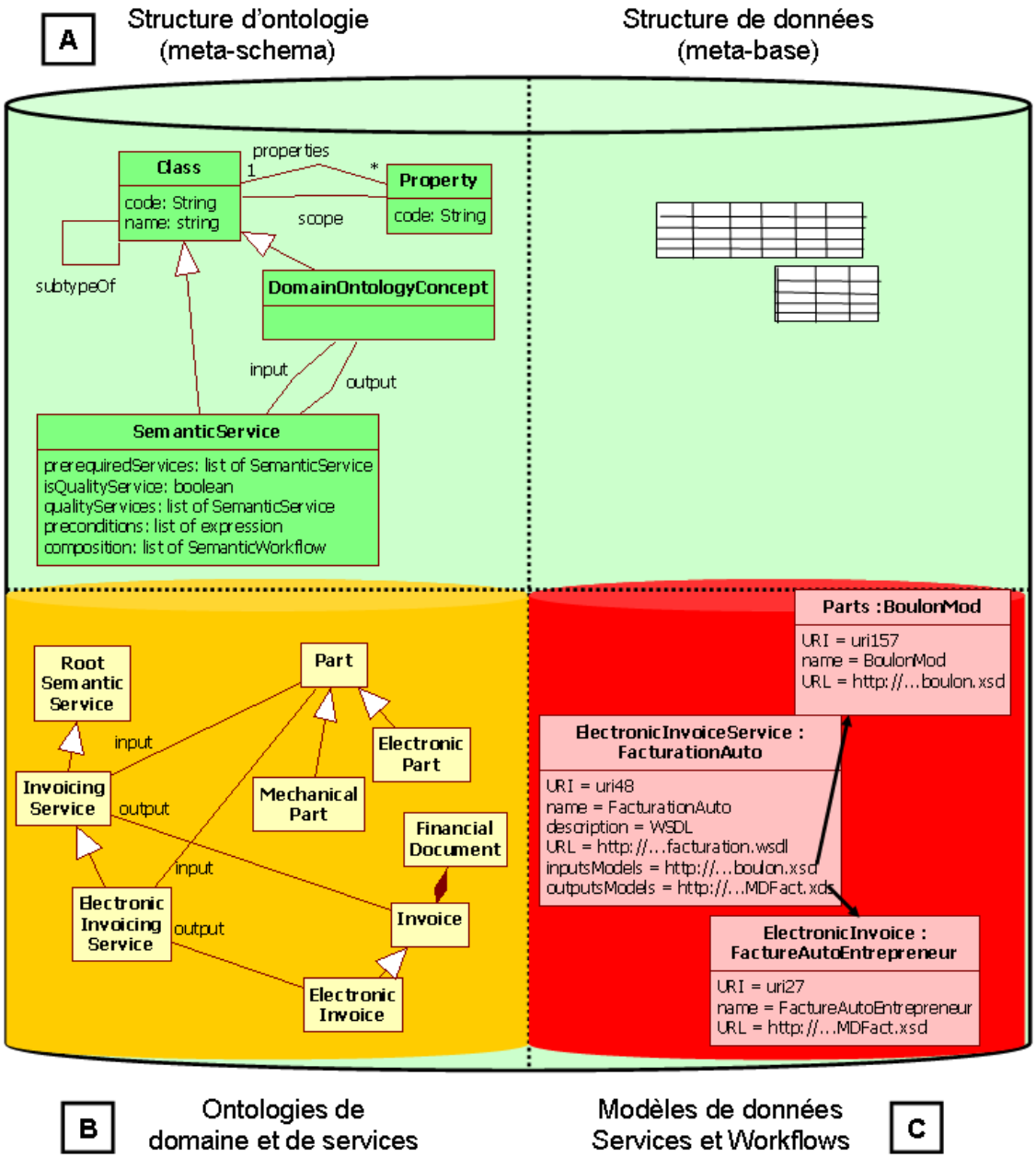


FIG. 5.7 – Extension d'OntoDB pour la représentation d'ontologies de services et entreposage.

rendues possibles par le stockage de l'ensemble des informations dans OntoDB. Il est ainsi possible par exemple de rechercher des services et des workflows informatiques par leur sens ou de remplacer un service ou un workflow informatique par un autre remplissant la même fonction.

La complexité des requêtes réside dans le fait qu'elles interrogent la base de données à base ontologiques à différents niveaux (méta-modèle ou modèle d'ontologies), différentes piles (pile de données ou pile de service) et différentes représentations conceptuelles (modèles et ontologies).

Dans la suite de cette section, nous ne présentons que quelques requêtes permettant de rechercher les services et les workflows informatiques grâce aux indexations sémantiques. Cependant, ces requêtes peuvent également être croisées et combinées pour former des requêtes plus complexes.

5.1 Recherche de services sémantiques à travers leurs attributs sémantiques

Notre approche permet de rechercher des services sémantiques à travers leurs attributs fonctionnels sémantiques. Par exemple, des services sémantiques peuvent être retrouvés suivant les concepts ontologiques qu'ils prennent en entrée. Ce type de requête est utile pour trouver les services sémantiques manipulant un concept ontologique donné.

Requête. Retrouver les services sémantiques qui prennent en entrée une facture, c'est-à-dire le concept ontologique *Invoice*.

```
SELECT SS.#name
FROM #SemanticService SS, #DomainOntologyConcept DOC
WHERE DOC.#name = 'Invoice'
 AND DOC.#oid IN SS.#inputs
```

Explication. Rechercher les noms (`SS.#name`) des services sémantiques (`SS`) qui ont comme entrée sémantique (`DOC.#oid IN SS.#inputs`) le concept ontologique *Invoice* (`DOC.#name = 'Invoice'`).

Un autre exemple est la recherche des services sémantique qui peuvent potentiellement précéder ou succéder au déroulement d'un service sémantique donné.

Requête. Retrouver les services sémantiques qui sont les successeurs potentiels du service sémantique *PurchasingService*.

```
SELECT SS2.#name
FROM #SemanticService SS1, #SemanticService SS2
WHERE SS1.#name = 'PurchasingService'
 AND SS1.#oid IN SS2.#prerequiredServices
```

Explication. Rechercher les noms (`SS2.#name`) des services sémantiques (`SS2`) qui succèdent (`SS1.#oid IN SS2.#prerequiredServices`) au service sémantique *PurchasingService* (`SS1.#name = 'PurchasingService'`).

5.2 Recherche de services sémantiques à travers des services et workflows informatiques indexés

L'indexation de services et de workflows informatiques par des services sémantiques permet de savoir à quel service sémantique appartient un service ou un workflow informatique donné et ainsi de savoir quelle est sémantiquement sa fonction.

Requête. Retrouver quel service sémantique indexe le service ou le workflow informatique dont l'URI est *uri25*.

```
SELECT SS.#name
FROM #SemanticService SS, RootSemanticService RSS
WHERE TYPEOF(RSS).#oid = SS.#oid
 AND RSS.uri = 'uri25'
```

Explication. *RootSemanticService* est le service sémantique racine, qui subsume tous les autres services sémantiques. `TYPEOF(RSS)` indique qu'on recherche le concept ontologique duquel le service informatique `RSS` est instance. Ainsi, il s'agit de rechercher le ou les noms (`SS.#name`) du ou des services sémantiques (`SS`) qui indexent (`TYPEOF(RSS).#oid = SS.#oid`) le service ou workflow informatique dont l'URI est *uri25* (`RSS.uri = 'uri25'`).

5.3 Recherche de services ou de workflows informatiques

Les services et les workflows informatiques sont indexés par des services sémantiques en instanciant ces derniers. Ainsi, rechercher sémantiquement des services ou des workflows informatiques se fait en recherchant un service sémantique particulier dont la fonction nous intéresse. Ensuite, il s'agit de chercher dans ses instances les services et les workflows informatiques qui réalisent la fonction cible. De plus, les instances d'un service sémantique donné sont non seulement ses instances directes, mais également les instances des services sémantiques qui le subsument. Ainsi, il est possible de rechercher des services ou des workflows informatiques soit par les fonction qu'ils réalisent, soit par des fonctions plus spécialisées à celles réalisées.

Requête. Retrouver dans OntoDB tous les services et workflows informatiques qui réalisent une facturation.

```
SELECT EIS.uri, EIS.url
FROM InvoicingService EIS
```

Explication. Rechercher les identifiants (`EIS.uri`) ainsi que les chemins (`EIS.url`) des services ou des workflows informatiques qui sont instances du service sémantique *InvoicingService* (`FROM InvoicingService EIS`). Notons que cette requête retournera également

les services instances d'une sous-classe de `InvoicingService`. Par défaut, le comportement est en effet de traverser la relation de subsomption.

Requête. Retrouver dans OntoDB tous les services et workflows informatiques qui réalisent *exactement* une facturation.

```
SELECT EIS.uri, EIS.url
FROM #SemanticService SS, InvoicingService EIS
WHERE TYPEOF(EIS).#oid = SS.#oid
 AND SS.#name = 'InvoicingService'
```

Explication. Rechercher les identifiants (`EIS.uri`) ainsi que les chemins (`EIS.url`) des services ou des workflows informatiques qui sont instances du service sémantique *InvoicingService* (**FROM** `InvoicingService EIS`) et qui ne sont instances que de ce service sémantique (`TYPEOF(EIS).#oid = SS.#oid` **AND** `SS.#name = 'InvoicingService'`).

Les exemples précédents ont montré l'utilisation de la relation de subsomption pour rechercher un service ou workflow informatique. Nous nous intéressons maintenant à l'utilisation des attributs qui caractérisent un service sémantique.

5.4 Recherche de services ou de workflows informatiques avec restriction sur les propriétés et attributs sémantiques

Notre approche permet d'effectuer une recherche sur les attributs des services sémantiques indexant des services ou des workflows informatiques donnés. Ceci est réalisé en testant la valeur des attributs sémantiques des services sémantiques recherchés (entrée sémantique, sortie sémantique, service sémantique pré-requis, etc.).

Requête. Retrouver dans OntoDB tous les services et workflows informatiques qui réalisent une facturation ou une fonction plus spécialisée et dont le service sémantique indexant a comme sortie sémantique une facture *Invoice*.

```
SELECT CI.uri, CI.url
FROM #SemanticService SS, #DomainOntologyConcept DOC, SS AS CI,
 InvoicingService EIS
WHERE DOC.#name = 'Invoice'
 AND DOC.#oid IN SS.#outputs
 AND EIS.uri = CI.uri
```

Explication. OntoQL permet l'interrogation conjointe des ontologies et des données. En effet, le constructeur **AS** permet de calculer les instances de classes déterminées dynamiquement. Ainsi, la requête précédente permet de retrouver tous les services sémantiques

`#SemanticService SS` qui ont comme entrée le concept ontologique *Invoice* (`DOC.#name = 'Invoice' AND DOC.#oid IN SS.#outputs`). Ensuite, tous les services sémantiques obtenus sont parcourus (`SS AS CI`). Enfin, seuls les services et workflows informatiques qui réalisent une facturation (`InvoicingService EIS`) sont gardés (`EIS.uri = CI.uri`).

Il est également possible de restreindre une recherche sur les propriétés des services ou des workflows informatiques. Elle est réalisée en recherchant les instances d'un service sémantique particulier qui vérifient une ou plusieurs contraintes sur les propriétés.

Requête. Retrouver dans OntoDB tous les services et workflows informatiques qui réalisent une facturation ou une fonction spécialisée et qui ont comme modèle de données en sortie celui que décrit le ficher XML-Schema dont l'URL est *http://../MDFact.xsd*.

```
SELECT EIS.uri, EIS.url
FROM InvoiceService EIS, RootOntologyConcept ROC
WHERE ROC.oid IN EIS.outputsModels
 AND ROC.URL = 'http://.../MDFact.xsd'
```

Explication. Il s'agit de retrouver tous les identifiants des services et des workflows informatiques de facturation (`EIS.uri`) ainsi que leurs chemins (`EIS.url`) et dont le modèle de données en sortie (`ROC.oid IN EIS.outputsModels`) est *http://../MDFact.xsd* (`ROC.URL = 'http://.../MDFact.xsd'`).

Les requêtes présentées précédemment montrent que notre approche permet non seulement de rechercher des services informatiques selon leurs caractéristiques syntaxiques en s'appuyant sur le modèle de données utilisé mais aussi selon leurs caractéristiques sémantiques fournies par les ontologies qui indexent les services. Nous présentons maintenant un autre intérêt de notre approche, celui de pouvoir faciliter l'adaptativité des services.

5.5 Adaptativité de services

L'adaptativité est la possibilité de retrouver automatiquement un service ou un workflow informatique remplaçant un service donné. Par service ou workflow remplaçant, nous entendons un service ou un workflow qui remplirait une fonction sémantiquement équivalente.

Si les services qui remplissent une fonction parfaitement équivalente sont indexés par le même service sémantique, nous pouvons écrire une requête pour retrouver les services répondant à l'adaptativité parfaite.

5.5.1 Adaptativité parfaite

L'adaptativité parfaite est le fait de retrouver uniquement les services ou les workflows informatiques qui remplissent une fonction parfaitement équivalente, c'est-à-dire des services ou

des workflows informatiques qui sont indexés par le même service sémantique.

Requête. Retrouver dans OntoDB les services ou les workflows informatiques qui remplissent une fonction équivalente au service ou workflow informatique dont l'uri est *uri25*.

```
SELECT RSS1.uri, RSS1.url
FROM RootSemanticService RSS1, RootSemanticService RSS2
WHERE (TYPEOF(RSS1).#oid = TYPEOF(RSS2).#oid)
 AND (RSS2.uri = 'uri25')
 AND NOT (RSS1.oid = RSS2.oid)
```

Explication. Il s'agit de retrouver tous les identifiants des services informatiques `RSS1.uri` ainsi que les liens physiques vers leurs descriptions `RSS1.url` qui sont indexés par le même service sémantique (`TYPEOF(RSS1).#oid = TYPEOF(RSS2).#oid`) que le service ou workflow informatique dont l'uri est *uri25* (`RSS2.uri = 'uri25'`) et qui sont bien évidemment différents de ce même service (**NOT** `(RSS1.oid = RSS2.oid)`).

5.5.2 Adaptativité spécialisée

L'adaptativité spécialisée est le fait de retrouver tous les services ou les workflows informatiques qui remplissent une fonction parfaitement équivalente ou plus spécialisée. Dans notre approche, ceci s'interprète comme les services ou les workflows informatiques qui sont indexés par le même service sémantique ou un service sémantique subsumé.

Requête. Retrouver dans OntoDB les services ou workflows informatiques qui remplissent une fonction plus spécialisée que celle remplie par le service ou le workflow informatique dont l'uri est *uri25*.

```
SELECT RSS1.uri, RSS1.url
FROM RootSemanticService RSS1, RootSemanticService RSS2
WHERE (TYPEOF(RSS2).#oid IN TYPEOF(RSS1).#oid.#superclasses)
 AND (RSS2.uri = 'uri25')
 AND NOT (RSS1.oid = RSS2.oid)
```

Explication. Il s'agit de retrouver tous les identifiants des services informatiques `RSS1.uri` ainsi que leurs chemins `RSS1.url` qui sont indexés par le même service sémantique ou par un service sémantique subsumé par celui-ci (`TYPEOF(RSS2).#oid` **IN** `TYPEOF(RSS1) .#oid. #superclasses`) que le service ou workflow informatiques dont l'uri est *uri25* (`RSS2.uri = 'uri25'`) et qui sont bien évidemment différents de ce même service (**NOT** `(RSS1.oid = RSS2.oid)`).

Les requêtes de cette section montrent comment les notions d'adaptativité parfaire et spécialisée peuvent être interprétées et mis en œvre dans notre approche. Nous pourrions également

écrire d'autres requêtes pour répondre à d'autres formes d'adaptativité (les services dont les paramètres d'entrée sont une spécialisation de ceux recherchés, etc.)

Ayant présenté l'implantation du modèle logique sur OntoDB et les requêtes OntoQL correspondantes, nous présentons dans la prochaine section, les développements effectués pour étendre le langage OntoQL qui ont permis cette implantation.

6 Extension d'OntoQL

Le langage OntoQL est un langage de requêtes, il permet entre autre l'exploitation de BDBO de type 3. Une implémentation est disponible sur OntoDB. Toutefois, des insuffisances quand à son implémentation ne permettent pas l'extension du modèle d'ontologie d'OntoDB pour la représentation de certains aspects relatifs aux services sémantiques. Nous les avons implémentés et présentons dans ce chapitre l'extension d'OntoQL pour la représentation d'attributs multivalués.

Nous montrons dans la figure 5.8 la représentation dans OntoDB de l'attribut multivalué *inputs* de l'entité *SemanticService*. Sur la partie gauche de cette figure se trouve le fragment du modèle d'ontologie simplifié pour la représentation des services sémantiques. Sur la partie droite se trouve le modèle logique correspondant utilisé par OntoDB et l'illustration de l'instanciation du service sémantique *InvoicingService* (voir la figure 4.8 du chapitre 4) avec référencement de ses entrées sémantiques (en supposant qu'il référence deux concepts : une pièce *Part* et une entreprise *Company*). La représentation dans OntoDB pour les attributs multivalués se fait comme suit. Une table d'association *SemanticService_2_inputs* est créée pour stocker les relations entre l'entité *SemanticService* et son attribut multivalué *inputs*. Lors de l'instanciation de l'entité *SemanticService* pour créer le service sémantique *InvoicingService* dont l'oid est *21*, la table d'association *SemanticService_2_inputs* est d'abord modifiée. Deux lignes sont ajoutées pour représenter respectivement la correspondance entre *InvoicingService* et *Part* d'une part, et *InvoicingService* et *Company* d'autre part. Enfin, les deux nouvelles instances de la table d'association sont les éléments de la liste correspondante à la valeur de *inputs* pour le service sémantique *InvoicingService*.

Nous avons étendu le langage OntoQL pour qu'il permette la manipulation d'attributs multivalués. Nous présentons dans la figure 5.9 un fragment de code correspondant aux développements réalisés. Ainsi, dans le cas de l'instanciation d'un attribut multivalué, nous recherchons d'abord l'oid de l'instance de l'entité, puis les oid des concepts référencés par ce même attribut. Ensuite, la table d'association est mise à jour avec les relations entre les oid des concepts référencés et l'instance de l'entité. Enfin, les oid correspondant à ces insertions sont les valeurs de la liste insérée dans l'attribut de l'instance de l'entité à ajouter.

Cette extension d'OntoQL nous a permis de créer des services sémantiques ayant une liste d'entrées, de sorties et de services sémantiques prérequis. Nous avons également pu les exploi-

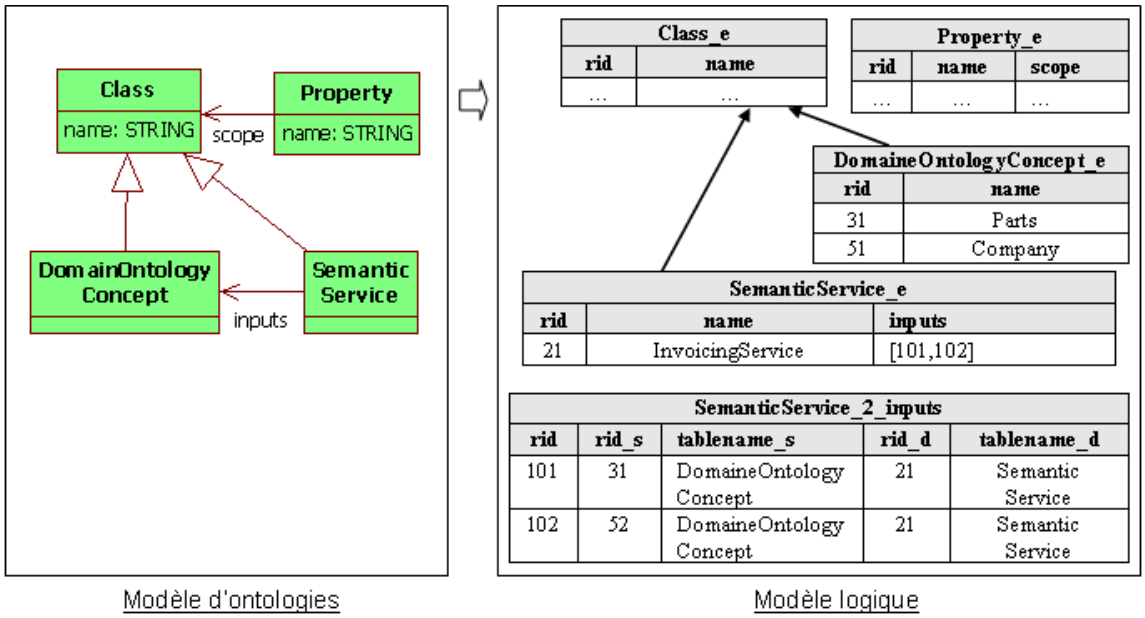


FIG. 5.8 – Exemple de représentation d'attributs multivalué dans OntoDB.

ter au niveau découverte de services sémantiques comme nous l'avons montré précédemment. Enfin, la modification du méta-modèle a entraîné la régénération de l'API OntoAPI, décrite dans le chapitre 1, permettant à des programmes JAVA d'accéder à OntoDB car nous l'utilisons pour le développement du logiciel de gestion de services et de workflows que nous présentons dans le dernier chapitre de ce mémoire. Dans la prochaine section, nous présentons les développements effectués pour la vérification de contraintes qui ne sont pas exprimables au niveau du modèle logique d'OntoDB.

7 Implantation des contraintes

L'établissement des contraintes vise à assurer d'une part la cohérence dans la définition des ontologies de services, et d'autre part, la conformité entre les services sémantiques et les services et workflows informatiques qu'ils indexent. Dans cette section, nous présentons les développements effectués pour implanter ces contraintes.

7.1 Implantation de contraintes de covariance

La covariance (voir chapitre 4, section 3.2) permet de vérifier la conformité des propriétés fonctionnelles des services sémantiques avec celles du service sémantique subsumant. La figure 5.10 montre un exemple de covariance. Dans cette figure, nous avons le service sémantique *ElectronicInvoicingService* qui est subsumé par le service sémantique *InvoicingService*. La covariance implique que les entrées et les sorties sémantiques de *ElectronicInvoicingService*

```
...
protected void processArrayInUpdate(AST updateStatement) {
    ...

    // tester le cas où c'est une référence à une "liste" d'attributs
    if (isArrayInUpdate(updateStatement)) {
        ...
        // rechercher l'oid de l'entité dont l'attribut de type array est à valuer
            //variable oidOfEntity = 21
        ...
        // rechercher les oids qui sont référencé par l'attribut
            //variable arrayChildren = [31,51] et numChildren = 2
        ...
        //retrouver le nom de la table d'association
            //variable arrayAttributeTableName = SemanticService_2_inputs
        ...

        // insérer les éléments dans la table d'association
            //iinsérer dans la table SemanticService_2_inputs les deux lignes(101, 31, 21) et (101, 51, 21)
        for (int i = 0; i < numChildren; i++) {
            sql = "insert into " + arrayAttributeTableName
                    + " (rid_s, rid_d) values (" + oidOfEntity
                    + ", " + arrayChildren[i] + ");";
            st = session.connection().createStatement();
            st.executeUpdate(sql);}

        // rechercher les oid des éléments de la tables d'association
            // variable arrayChildrenSql = [101,102]
        sql = "select rid from " + arrayAttributeTableName
                + " where rid_s=" + oidOfEntity;
        st = session.connection().createStatement();
        rSet = st.executeQuery(sql);
        for (int i = 0; i < numChildren; i++) {
            rSet.next();
            arrayChildrenSql[i] = rSet.getInt(1);}

        // insérer dans le tableau de l'attribut la liste d'oids de la table d'association
            //insérer dans inputs de la table SemanticService_e la valeur [101,102] à la ligne correspondante
        arrayNode.getFirstChild().setText(String.valueOf(arrayChildrenSql[0]));
        remainArrayQuery = arrayNode.getFirstChild();
        for (int i = 1; i < numChildren; i++) {
            remainArrayQuery = remainArrayQuery.getNextSibling();
            remainArrayQuery.setText(String.valueOf(arrayChildrenSql[i]));}
    }
}
...
```

FIG. 5.9 – Fragment de code JAVA pour l'extension du développement d'OntoQL.

doivent respectivement être subsumées par les entrées et les sorties sémantiques de *Invoicing-Service*. Ainsi, *Part* est bien subsumé par lui-même et *ElectronicInvoice* est bien subsumé par *Invoice*.

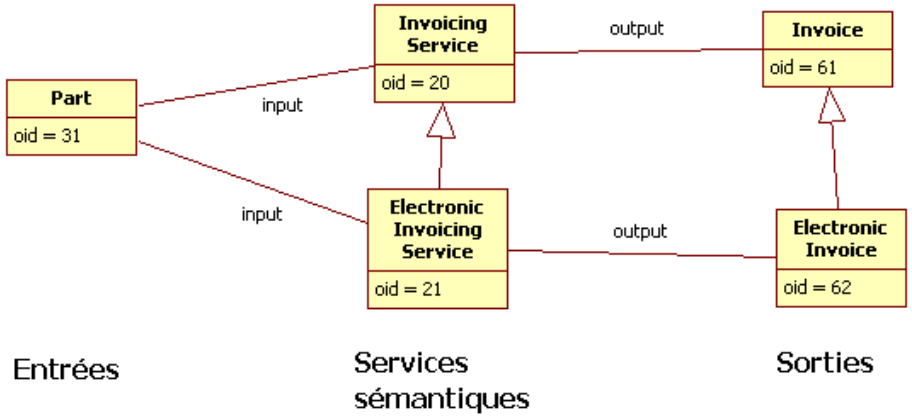


FIG. 5.10 – Exemple de covariance.

Cette contrainte n'est pas exprimée au niveau du modèle logique d'OntoDB. Pour assurer le respect de la covariance, nous avons défini des *déclencheurs* (ou *Triggers*) qui effectuent des vérifications lors de la création d'un nouveau service sémantique. Nous montrons dans la figure 5.11 un fragment de code en PL/SQL du *déclencheur* global *check_covariance_outputs* permettant de vérifier la covariance au niveau des concepts en sortie d'un service sémantique. Ainsi, lors de la création d'un nouveau service sémantique, le programme recherche les oid des concepts en sortie du service sémantique, puis éventuellement l'oid de son service sémantique subsumant. Ensuite, pour chaque oid de concept en sortie, le programme vérifie que l'un de ses concepts subsumant est un concept en sortie du service sémantique parent. Si l'un des oid de concept en sortie ne vérifie pas cette condition, alors le service sémantique à créer ne vérifie pas la propriété de covariance.

7.2 Implantation de contraintes sur la relation entre données sémantiques et modèles de données

Les services sémantiques font référence à des concepts d'ontologies de domaine en entrée et en sortie (niveau ontologie). Les services et les workflows informatiques sont les instances de ces services sémantiques (niveau instance). Lors de l'insertion d'un service ou d'un workflow informatique en tant qu'instance d'un service sémantique pour indexation, il faut vérifier que les modèles de données référencés par le service ou workflow informatique sont bien indexés par les concepts des ontologies de domaine référencés par le service sémantique indexant. La figure 5.12 montre un exemple d'indexation du service Web *FacturationAuto* par le service

```
CREATE OR REPLACE FUNCTION check_covariance_outputs() RETURNS "trigger" AS $BODY$
DECLARE
    ...
BEGIN
    ...
    --retrouver la liste des oids des entrées du "SemanticService_e" à ajouter
        --variable listofoidofoutputs = [62]
    i:= 0;
    oidofassociation:= NEW.outputs[i+1];
    WHILE NOT (oidofassociation IS NULL) LOOP
        sql := 'SELECT ss.* FROM SemanticService_e ss, SemanticService_2_outputs ssi
        WHERE '||NEW.rid_bsu||' = ssi.rid_s AND ss.rid_bsu = ssi.rid_d
        AND '||NEW.outputs[i]||' = ssi.rid;';

        FOR tuple IN EXECUTE sql LOOP tmp := tuple ;END LOOP;
        listofoidofoutputs[i] := tmp.rid_bsu;

        i := i + 1;
        oidofassociation:= NEW.outputs[i+1];
    END LOOP;

    --retrouver l'oid du service sémantique parent du "SemanticService_e" à ajouter
        --variable oidofsuper = 20
    ...

    --si le "SemanticService_e" à ajouter possède un père, alors il faut vérifier si la covariance est respectée
    IF NOT (oidofsuper IS NULL) THEN
        --retrouver la liste des oids des entrées du service sémantique parent du "SemanticService_e" à ajouter
            --variable listofoidofoutputsofsuper = [61]
        ...

        --maintenant, vérifier la covariance pour tous les éléments du tableau listofoidofoutputs
        FOR k IN 1..numinputs LOOP
            respected:=false;

            --oidofsuperofoutput : tableau des oid des parents de la "k"ième entrée du "SemanticService_e" à ajouter
            oidofsuperofoutput := superclasses(listofoidofoutputs[k]);
            ...
            --oidofoutputofsuper : oid de l'entrée du services sémantique parent à la "k" ième position
            oidofoutputofsuper:=listofoidofoutputsofsuper[k];

            FOR h IN 1..dim(oidofsuperofoutput) LOOP
                IF (oidofoutputofsuper = oidofsuperofoutput[h]) THEN
                    respected := TRUE;END IF;END LOOP;

            IF NOT respected OR NOT (oidofinputofsuper = listofoidofinputs[k]) THEN
                RAISE EXCEPTION 'The multi-covariance is not respected!';END IF;
    ...
```

FIG. 5.11 – Fragment de code PL/SQL pour la création d'un déclencheur pour la vérification de la covariance.

sémantique *ElectronicInvoicingService*. La vérification de l'indexation des modèles de données en entrée et en sortie du service Web *FacturationAuto* par les concepts en entrée et en sortie du service sémantique *ElectronicInvoicingService* n'est pas effectuée au niveau du modèle logique d'OntoDB. Nous avons donc défini et implémenté des contraintes qui permettent de vérifier la conformité des indexations au niveau sémantique avec les indexations au niveau informatique.

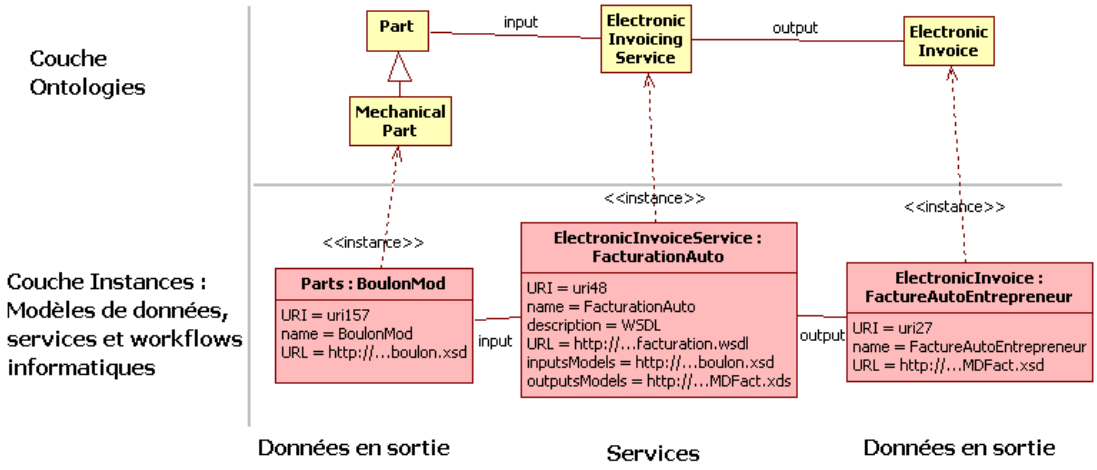


FIG. 5.12 – Exemple de contraintes entre le niveau ontologique et le niveau instance.

8 Conclusion

Dans ce chapitre, nous avons exploité notre approche sur la base de données à base ontologique OntoDB. Le modèle d'ontologies présent dans OntoDB ne permettant pas la représentation d'ontologies de services au sens où nous les avons définies, nous avons ainsi étendu le partie méta-schéma d'OntoDB. Nous avons ensuite créé les services sémantiques qui composent les ontologies de services conçues. Enfin, nous avons permis l'indexation sémantique de services et de workflows informatiques en instanciant les services sémantiques créés. Nous avons créé de nouvelles fonctionnalités basées sur des requêtes OntoQL permettant une meilleure gestion des services et des workflows informatiques. Ainsi, ces requêtes permettent entre autre, d'exploiter la subsomption en recherchant les services et workflows informatiques indexés par un service sémantique et les services sémantiques qu'il subsume. Elles permettent de rechercher les services et les workflows informatiques par les fonctions qu'ils réalisent, par les propriétés fonctionnelles sémantiques (concepts en entrée, concepts en sortie, services sémantiques prérequis, etc.), par les propriétés fonctionnelles syntaxiques (modèles de données en entrée, en sortie, etc.) et d'autres propriétés non-fonctionnelles (performance, disponibilité, sécurité etc.)[15]. Les requêtes permettent également l'adaptativité des services et workflows informatiques en

[15]Les propriétés non-fonctionnelles ne sont pas l'objet de cette thèse.

recherchant les services ou workflows remplissant des fonctions à des niveaux de similitude variés. Enfin, nous avons étendu OntoQL pour permettre la représentation des services sémantiques dans OntoDB et développé des requêtes de vérification de contraintes à l'origine non exprimables au niveau logique d'OntoDB.

Dans la prochaine partie, nous montrons l'application de nos travaux à la modélisation géologique et présentons l'application logicielle d'aide à la gestion sémantique de services et de workflows informatiques que nous avons développée.

Troisième partie

Application à la modélisation géologique

Chapitre 6

Application de notre proposition à la modélisation géologique

Sommaire

1 Introduction 135

2 Gestion des données en modélisation géologique 135

2.1 Modèles de données en modélisation géologique 135

2.2 Ontologies de domaine en modélisation géologique 138

2.3 Indexation des modèles données de modélisation géologique . . 138

3 Gestion des services en modélisation géologique 139

3.1 Services et workflows en modélisation géologique 141

3.2 Ontologies de services en modélisation géologique 141

3.3 Indexation des services et workflows informatiques de modélisation géologique 144

4 Construction de workflows en modélisation géologique 146

4.1 Processus de construction d'un workflow sémantique pour l'interprétation sismique 146

4.2 Composition sémantique en interprétation sismique 147

4.3 Mise à jour de l'ontologies des services de l'interprétation sismique par addition de services sémantiques composés 148

4.4 Construction d'un workflow informatique pour l'interprétation sismique 149

5 Conclusion 150

Résumé. L'approche que nous avons proposée dans les chapitres précédents est une méthode générique pour la gestion de services et des workflows informatiques. Pour la valider, nous l'appliquons dans ce chapitre au domaine particulier de la

modélisation géologique. Nous décrivons l'indexation sémantique des services et workflows informatiques dans ce domaine et présenterons les ontologies conçues. Nous montrons dans une seconde partie l'utilisation du travail réalisé en indexation pour la conception de nouveaux workflows.

1 Introduction

Dans la partie précédente, nous avons présenté notre proposition pour la caractérisation sémantique des services et des workflows informatiques. Nous l'avons implantée sur la base de données à base ontologique OntoDB en proposant une extension du langage d'exploitation OntoQL pour supporter la relation de subsomption de services sémantiques proposée.

Nous présentons dans ce chapitre, l'application de notre approche au domaine de la modélisation géologique et particulièrement en interprétation sismique. Notre objectif est d'expliciter la sémantique des modèles de données utilisés en modélisation géologique, ainsi que des services et des workflows informatiques qui les manipulent. Nous nous basons pour cela sur les ontologies de domaine et de services en modélisation géologique. Expliciter la sémantique des modèles et des services permet de les indexer sémantiquement en vue de faciliter leur découverte, adaptativité et interopérabilité. L'indexation sémantique permet également la construction de nouveaux workflows informatiques en construisant préalablement des workflows sémantiques et ensuite des workflows informatiques.

Dans un premier temps, nous présentons la partie statique de notre modélisation, c'est-à-dire la partie concernant les ontologies de domaine et les modèles de données de modélisation géologique. Ensuite, nous présentons la partie dynamique, c'est-à-dire la partie concernant les ontologies de services ainsi que les services et workflows informatiques de modélisation géologique. Dans chaque partie, nous expliquons comment nous indexons les modèles de données, les services et les workflows informatiques de modélisation géologique par leur description ontologique. Nous montrons également comment des workflows sémantiques en modélisation géologique sont construits pour en "déduire" de nouveaux workflows informatiques.

2 Gestion des données en modélisation géologique

Dans la partie qui correspond aux données de la modélisation géologique, il s'agit de représenter les modèles de données utilisés pour représenter les données de la modélisation géologique ainsi que leur définition sémantique à travers les ontologies de domaine. La figure 6.1 illustre les trois parties de la gestion des données en modélisation géologique suivant notre approche. Dans cette figure, les données et les modèles de données en modélisation géologique (voir section 2.1) sont caractérisés sémantiquement par des concepts d'ontologie de domaine (voir section 2.2) à travers des indexations sémantiques (voir section 2.3).

2.1 Modèles de données en modélisation géologique

L'industrie pétrolière dépend fortement des modèles informatiques de données. Chacun des workflows informatiques conçus en ingénierie pétrolière en général et en modélisation géolo-

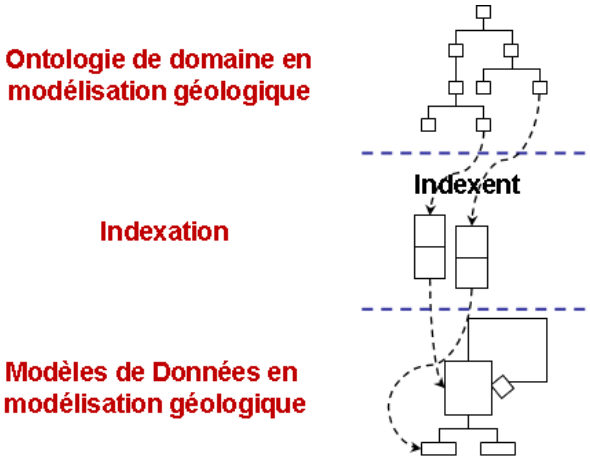


FIG. 6.1 – Ontologie de domaine, données et indexation.

gique en particulier génère un volume important de données conformes à des modèles particuliers et qui permettent les interprétations, le développement de scénarii et enfin les prises de décision.

Les modèles géologiques sont les représentations des objets géologiques et permettent l'identification et la caractérisation de réservoirs potentiels. Ce sont des représentations 2D, 3D ou 4D de données ou d'interprétations de données concernant des ressources souterraines ou se trouvant sous les fonds marins. Ces modèles sont développés par des experts en sciences géologiques qui ont des connaissances métiers approfondies de ces domaines. Leur objectif final est la construction d'un modèle de réservoir utilisé pour des simulations d'écoulement de fluides.

Parmi ces modèles, nous citons les *GRID DATA (.gr2d)* qui permettent de définir des surfaces ou des plans par des grilles régulières ou irrégulières de points et les *WELLS (.well)* qui peuvent contenir des informations sur des trajectoires de puits ainsi que des informations sur les propriétés de roches et les propriétés pétrophysiques des puits. Ces modèles de données peuvent être représentés dans des formats propriétaires ou standards XML définis par des XML-Schema.

Par exemple, nous montrons dans le tableau 6.1 la description du modèle de données *XYZ-File (.xyz)* décrit dans [Beicip-Franlab, 2000]. Ce modèle de données en XYZFile dont la version est 4.3 est un modèle qui permet la caractérisation de coordonnées de points dans l'espace. Il est utilisé pour représenter la topologie d'objets géologiques.

Pour pouvoir décrire la sémantique des services et des workflows informatiques conçus en modélisation géologique, nous avons principalement besoin de caractériser globalement les modèles données géologiques et de les indexer sémantiquement. Le tableau 6.2 montre la caractérisation de quelques modèles de données en modélisation géologique permettant leur référencement et leur indexation. Par exemple, le modèle de données *SEG-Y* est un modèle de données qui permet la représentation de cubes sismiques. Sa description syntaxique se trouve à l'adresse */ifp.fr/SEG-Y.txt*.

Nom du champs		Type	Description
FILE_DESCRIPTION		XYZ-V4.3	Modèle et version de description de fichier.
APPLICATION		String	Description de l'outil avec lequel le fichier a été généré.
SURVEY_NAME		String	Nom de l'étude.
SURFACE_NAME		String	Nom de l'objet géologique.
INTERFACE_ID		Integer	ID de l'objet géologique.
SURFACE_TYPE		String	Décrit le type de surface géologique (Horizon, Faille, etc.)
REPRESENTATION_CODE		String	Donne le format de la donnée (ASCII ou BINARY).
X_VALUES Y_VALUES Z_VALUES	CHANNEL1 CHANNEL2 CHANNEL3	Unit (String) Unit (String) Unit (String)	Indique les dimensions
X1 X2 ... Xn	Y1 Y2 ... Yn	Z1 Z2 ... Zn	Liste les coordonnées de points.

TAB. 6.1 – Modèle en *XYZFile (.xyz)*.

nom	url	description
SEG-Y	/ifp.fr/SEG-Y.txt	ce modèle de données permet la représentation de cubes sismiques, c'est-à-dire d'images sismique 2D ou 3D d'un prospect donnée
XYZFile	/ifp.fr/XYZ.txt	ce modèle de données permet la description de certain objets géologiques comme des réflecteurs, des horizons ou des failles etc.
...	...	...

TAB. 6.2 – Fragment de caractérisation de modèles de données en modélisation géologique.

2.2 Ontologies de domaine en modélisation géologique

La caractérisation des réservoirs de gaz et de pétrole est basée sur l'expertise des professionnels de diverses disciplines reliées aux sciences de la géologie. Pour que ces experts puissent échanger entre eux au travers d'outils de modélisation informatisés, il est nécessaire qu'ils s'accordent sur une représentation commune de la connaissance concernant les objets géologiques modélisés. Dans [Perrin et al., 2008], une ontologie globale, l'ontologie de la géologie de base (*basic geology ontology*) est proposée. Cette ontologie se rapporte aux objets géologiques définis dans les modèles géologiques. L'ontologie de la géologie de base est centrée sur le concept de *GeologicalObject* qui peut être très diversifié (une unité sédimentaire, un récif, une faille, sont autant de *GeologicalObjects*).

Des ontologies plus spécifiques ont été développées comme l'ontologie des temps géologiques ou l'ontologie de datation géologique décrites dans [Mastella, 2010]. De plus, les professionnels engagés dans la modélisation géologique souhaitent avoir accès à la totalité de la connaissance attachée aux objets qu'ils modélisent (unités et limites stratigraphiques, failles individuelles et réseaux de failles, etc.). C'est pour répondre à ce besoin, que des ontologies relatives à diverses disciplines des géosciences ont été développées pour caractériser les concepts qui émergent dans ces disciplines. Ces ontologies *locales* représentent les définitions des concepts attachés aux services et au workflows spécifiques développés le long de la chaine de modélisation géologique, tels que l'interprétation sismique ou la description des puits.

Dans la figure 6.2, nous montrons un fragment de l'ontologie de la sismique (*seismic ontology*) qui représente la partie amont de la modélisation géologique. Cette ontologie qui a été définie dans [Mastella, 2010] en rapport avec les travaux de [Verney, 2009] et dans le cadre du projet *e-Wok Hub* [e Wok Hub Consortium, 2008] [16] qui traite du stockage de CO_2.

Dans cette ontologie, le concept ontologique racine est représenté dans la figure 6.2 par *GeoConcept*. A partir de ce concept, sont dérivés les concepts suivants : *Well* et *SeismicSurface*. Un *SeismicCube* est composé d'un ensemble de *SeismicSurface*. Du concept de *SeismicSurface* sont dérivés les concepts : *ComponentSurface* et *InterpretedSurface*. Un *ComponentSurface* peut être un *Reflector*, un *HorizonGap* ou bien un *ChannelPartCandidate*. De même, une *InterpretedSurface* peut être un *Horizon*, une *Fault* ou un *Channel*. Enfin, une *Fault* peut être plus particulièrement une *DipFault*.

2.3 Indexation des modèles données de modélisation géologique

Dans notre approche, nous proposons d'indexer les modèles de données utilisés dans le domaine de la modélisation géologique par des concepts d'ontologies de domaine. Cette in-

[16] Projet ANR impliquant les partenaires suivants : LISI/ENSMA, l'IFP, le BRGM (Bureau de Recherches Géologiques et Minières), l'INRIA (Institut National de Recherche en Informatique et en Automatique), EADS (European Aeronautic Defense and Space), l'École des Mines de Paris et le CRITT informatique.

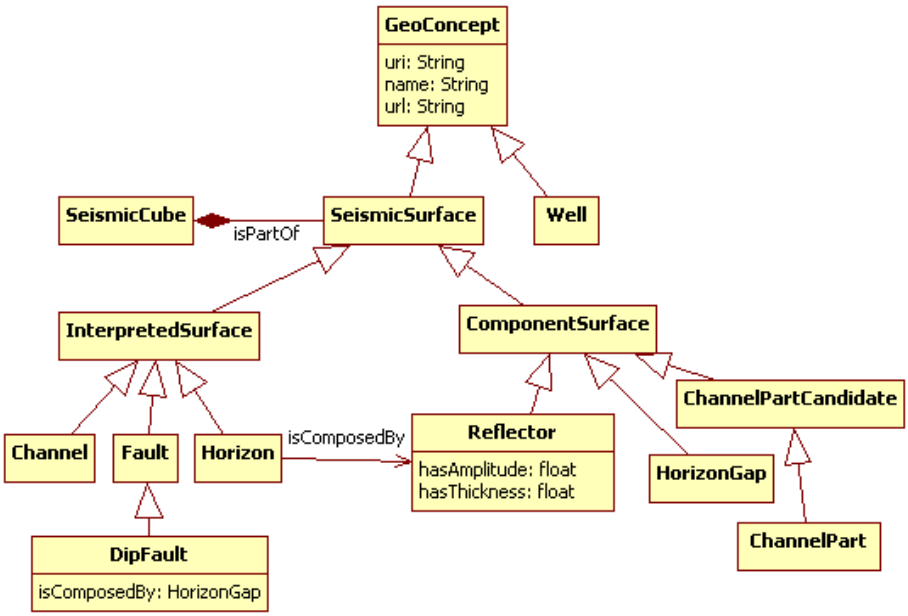


FIG. 6.2 – Fragment de l'ontologie du domaine d'interprétation sismique.

dexation peut être faite de plusieurs façons. Nous avons choisi de la réaliser en instanciant les concepts ontologiques. Ceci nous permet de bénéficier de la subsomption : les modèles qui sont indexés par un concept d'une ontologie de la modélisation géologique seront également indexés par les concepts subsumants. La figure 6.3 montre un exemple d'indexation d'un modèle de données par un concept de l'ontologie de la sismique en le rendant instance de ce concept. Ainsi, dans cette figure, le modèle de données *XYZFile* qui permet de représenter des horizons est indexé par le concept ontologique *Horizon* en instanciant ce dernier.

Ayant présenté la gestion de la partie statique, c'est-à-dire la partie des données de la modélisation géologique, nous nous intéressons à présent à la partie dynamique, c'est-à-dire aux services et aux workflows informatiques en modélisation géologique.

3 Gestion des services en modélisation géologique

Dans cette section, nous présentons la partie qui correspond aux services et aux workflows de la modélisation géologique ainsi qu'à leur définition sémantique à travers les ontologies de services. A l'image de la gestion des données, nous adoptons la même démarche pour la gestion des services et workflows informatiques. La figure 6.4 illustre les trois parties de cette gestion en modélisation géologique selon notre approche. Dans cette figure, les services et workflows informatiques en modélisation géologique (voir section 3.1) sont caractérisés sémantiquement par des concepts d'ontologie de services (voir section 3.2) à travers des indexations sémantiques

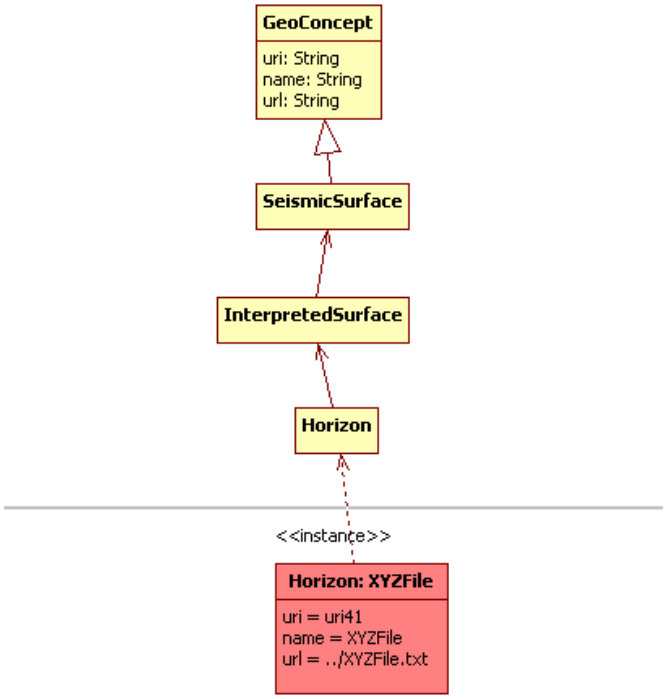


FIG. 6.3 – Indexation d'un modèle de données par un concept de l'ontologie de la sismique.

(voir section 3.3).

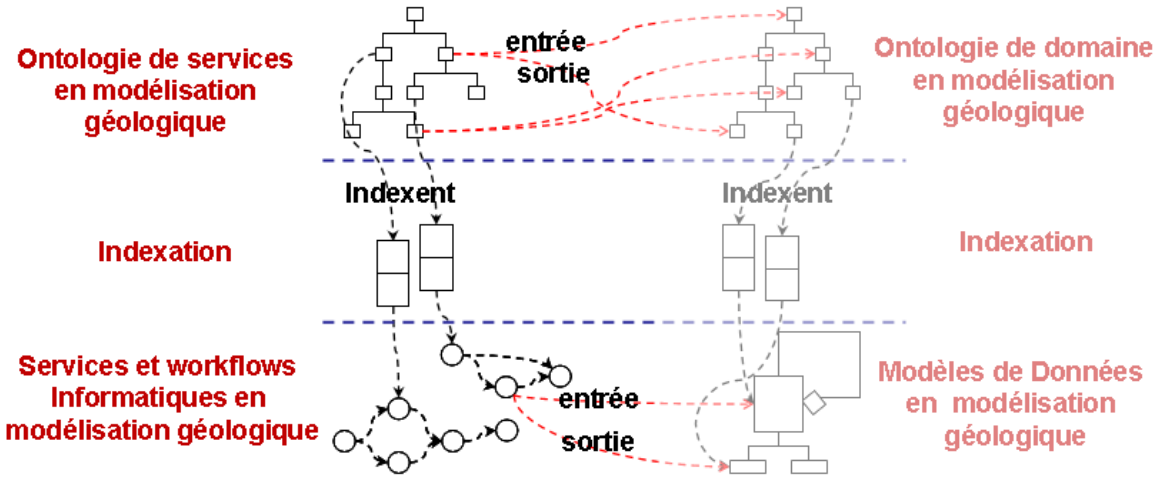


FIG. 6.4 – Ontologie de services, services, workflows informatiques et indexation.

3.1 Services et workflows en modélisation géologique

Les services et les workflows informatiques conçus pour la modélisation géologique ont pour objectif de transformer des données brutes pour permettre la représentation informatique de sous-sols et particulièrement de réservoirs pétroliers. Les fonctions et les processus qui définissent ces transformations sont implémentés pour créer des services et des workflows informatiques sans toutefois décrire de méthodologie explicite. Les services et les workflows développés en modélisation géologique et particulièrement dans sa partie amont, c'est-à-dire l'interprétation sismique, permettent à partir de données sismiques et de données aux puits, de reconstruire un modèle du sous-sol géologiquement cohérent qui est à la base de la construction d'un modèle structural. Nous avons présenté dans le chapitre 3 certaines fonctions et certains processus qu'implémentent ces services et workflows informatiques. Par exemple, le *picking* est une opération qui vise à pointer un objet d'intérêt sur une section 2D d'une image sismique et le *tracking* (en 3D) est un algorithme dont le but est d'étendre la sélection faite lors du *picking* sur l'ensemble de l'image sismique [Dorn, 1998].

La description des services et des workflows informatiques de la modélisation géologique se fait dans des langages de descriptions standards ou propriétaires. Dans notre approche, nous avons identifié un ensemble de caractéristiques fonctionnelles associées aux services et aux workflows informatiques. Parmi ces caractéristiques, il y a les paramètres d'entrée et de sortie définis dans des documents conformes aux modèles de données décrits dans la section 2.1. D'autres paramètres tels que des services informatiques prérequis sont utiles à l'exécution et à la gestion informatique de ces services et workflows informatiques. Le tableau 6.3 montre la représentation de certaines propriétés fonctionnelles sur quelques exemples de services et de workflows informatiques en modélisation géologique. Dans leurs descriptions, des paramètres décrivant leurs emplacements, leurs langages de description, leurs entrées et sorties identifiées par des URI qui correspondent à des modèles de données propriétaires ou à des standards en XML-Schema sont décrits.

Par exemple, le workflow informatique *RMark* réalise une extraction de réflecteurs. Il est décrit dans le langage de description de workflow *BPEL*, il prend en entrée le modèle de données *SEG-Y* et rend en sortie le modèle de données *WITSML*. La description de ce workflow est disponible à l'adresse */ifp.fr/RMark.bpel*. Pour que ce workflow informatique soit correctement exécuté, il faut l'utiliser sur des fichiers SEG-Y qui correspondent à des cubes d'image d'au moins une dizaine de kilomètre cube.

3.2 Ontologies de services en modélisation géologique

Afin de caractériser les workflows et les services de modélisation géologique, nous avons conçu des ontologies de services de modélisation géologique, en particulier pour les services et les workflows d'interprétation sismique [Belaid et al., 2009a]. Nous avons d'abord énuméré les services et les workflows de l'interprétation sismique. Ensuite, nous avons extrait la sémantique

nom	descr.	url	modèle de données en entrées	modèle de données en sorties	pré-requis	fonction
RMark	BPEL	/ifp.fr /RMark.bpel	SEG-Y format	WITSML format	le fichier SEG-Y en entrée doit correspondre à un cube d'au moins 10 km^3	ce workflow informatique est une compositions de services informatiques et qui réalise une extraction de réflecteurs (objets géologiques) représentés dans un modèle de données WITSML à partir d'un cube sismique représenté dans un autre modèle de données SEG-Y
getRef	WSDL	/ifp.fr /getRef.wsdl	SEG-Y format	XYZFile format	le fichier SEG-Y en entrée doit correspondre à un cube d'au moins 10 km^3	ce service informatique permet de réaliser une extraction de réflecteurs représentés dans un modèle de données XYZFile à partir d'un cube sismique représenté dans un modèle de données SEG-Y
HorExtWf	XPDL	/ifp.fr /HorExtWf.xpdl	XYZFile format	XYZFile format	Pas de prérequis	ce workflow informatique réalise une extraction d'horizons représentés dans un modèle de données XYZFile à partir de réflecteurs représentés dans un même modèle de données XYZFile
...	...	...	...	...	...	...

TAB. 6.3 – Quelques services et workflows informatiques de modélisation géologique.

des services, des workflows et des données qu'ils manipulent. Cela nous a permis d'énumérer les concepts de services en interprétation sismique, de leur rattacher des propriétés, de les représenter en hiérarchies et en compositions de services. Enfin, nous avons obtenu une ontologie pour les services d'interprétation sismique. La figure 6.5 montre un fragment de l'ontologie qui caractérise les services d'interprétation sismique.

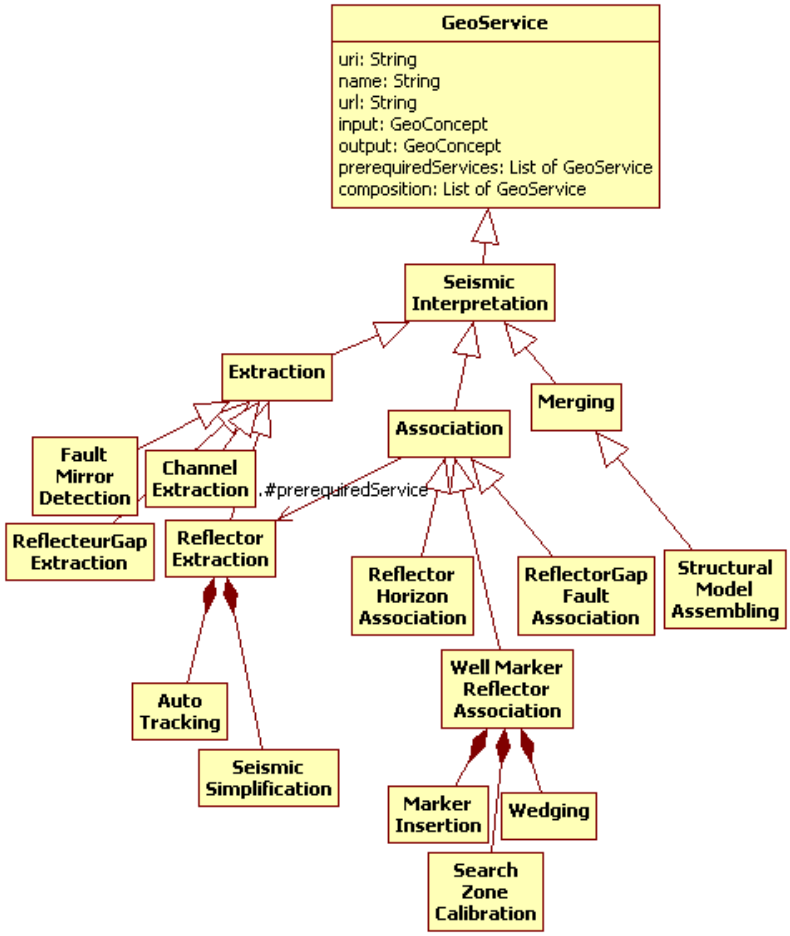


FIG. 6.5 – Fragment de l'ontologies des services pour l'interprétation sismique.

Cette ontologie a comme concept de base *GeoService* et plus particulièrement le concept *SeismicInterpretation*. A partir de ce concept, deux concepts sont issus. Le concept *Extraction* qui caractérise les services pour les extractions d'éléments géologiques. *Association* est le concept qui désigne les services qui associent deux ou plusieurs éléments géologiques relativement les uns par rapport aux autres. Enfin, le concept *Merging* fusionne des éléments géologiques. Le concept Extraction peut être spécialisé en d'autres concepts tels que *ReflectorExtraction*, *ReflectorGapExtraction*, *FaultMirrorDetection* et *ChannelExtraction*. *ReflectorExtraction* par exemple, caractérise les services dont l'objectif est d'extraire des réflecteurs. Il est composé des deux services sémantiques *AutoTracking* et *SeismicSimplification*. Le concept

Association peut être spécialisé en *ReflectorHorizonAssociation*, *ReflectorGapFaultAssociation* et *WellMarkerReflectorAssociation* etc.

Les concepts de cette ontologie de services ont des propriétés qui font référence aux concepts des ontologies de la modélisation géologique dont l'ontologie de l'interprétation sismique précédemment évoquée. Par exemple, le concept *ReflectorExtraction* a comme entrée un *SeismicCube* et comme sortie des *Reflector*. La figure 6.6 illustre quelques services sémantiques de l'ontologie de services en interprétation sismique (voir Figure 6.5) référençant des concepts de l'ontologie de domaine de la sismique (voir Figure 6.2). Par exemple, le service sémantique *Extraction* a comme entrée le concept *SeismicCube* et comme sortie le concept *SeismicSurface* et le service sémantique *ReflectorExtraction* a comme entrée le concept *SeismicCube* et comme sortie le concept *Reflector*. Notons que la covariance (voir chapitre 4 section 3.5) est vérifiée car le fait que le service sémantique *Extraction* subsume *ReflectorExtraction* implique bien que les entrées et sorties de *Extraction* subsument respectivement celles de *ReflectorExtraction*.

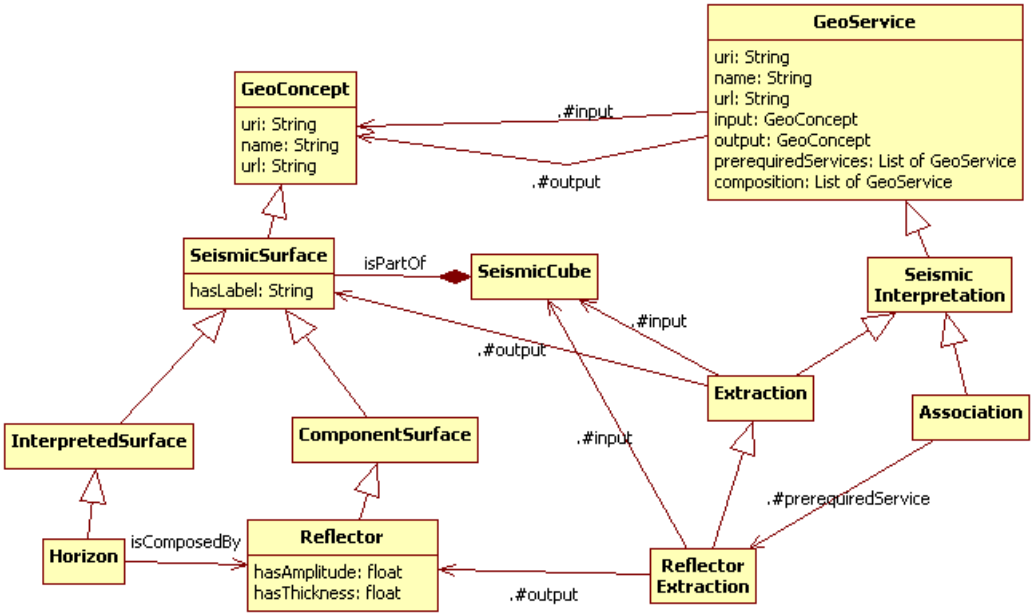


FIG. 6.6 – Exemple de référencement de concepts d'ontologies de domaine par des services sémantiques.

3.3 Indexation des services et workflows informatiques de modélisation géologique

L'objectif de la proposition que nous avons faite est de permettre l'indexation des services et des workflows informatiques en vue de leur réutilisation et de la construction de workflows informatiques plus complexe. A l'image de l'indexation des données, nous avons choisi de

réaliser l'indexation des services et des workflows informatiques en instanciant les services sémantiques. Ceci nous permet de bénéficier de la subsomption : les services et workflows indexés par un service sémantique seront également indexés par les services sémantiques subsumants. La figure 6.7 montre un exemple d'indexation d'un service Web par un service sémantique en l'instanciant. Ainsi, dans cette figure, le service Web *refGet* décrit en WSDL est indexé par le service sémantique *ReflectorExtraction* qui est un concept de l'ontologies des services de l'interprétation sismique. Ce service Web prend en entrée un modèle de fichier de type *SEG-Y* et rend en sortie un modèle de fichier de type *XYZFile*.

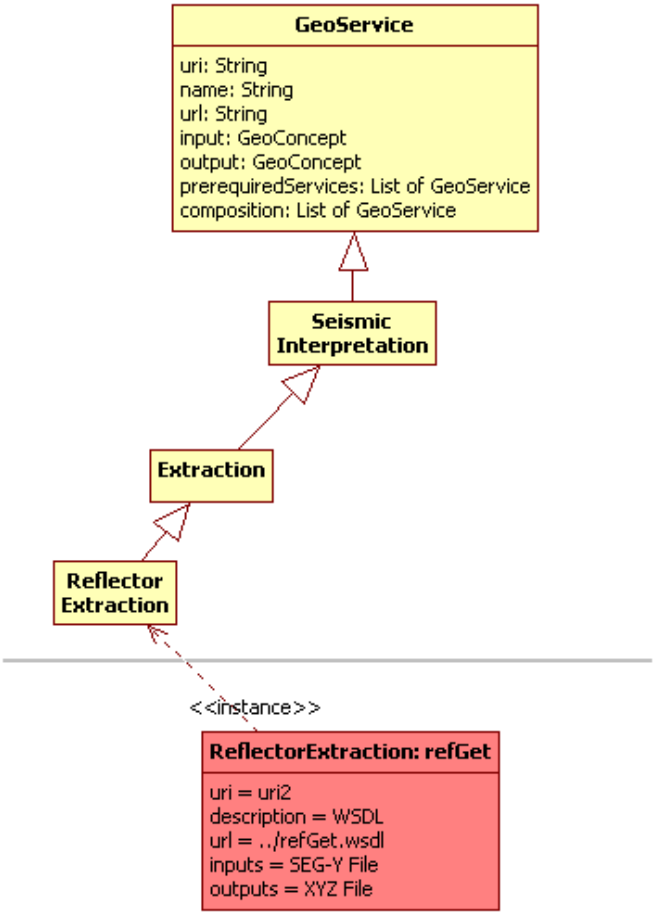


FIG. 6.7 – Indexation d'un service informatique de modélisation géologique par un service sémantique.

Dans cette section, nous avons présenté l'indexation des données et des services en modélisation géologique. Cette indexation sémantique permet la réutilisation des services et workflows informatiques en facilitant leur découverte, leur compréhension ou leur adaptativité. Nous nous basons sur ces descriptions pour proposer, dans la prochaine section, la construction de nouveaux workflows informatiques.

4 Construction de workflows en modélisation géologique

Dans cette partie, nous expliquons la construction de nouveaux workflows informatiques à partir du travail d'indexation et de description sémantique réalisé. Pour permettre cela, nous proposons de procéder d'abord à la composition de services sémantiques pour la construction de *workflows sémantiques*. Ces dernier permettent d'ailleurs d'enrichir la description ontologique des services en modélisation géologique. Nous montrons enfin comment cela peut permettre la construction voire la génération de workflows informatiques.

4.1 Processus de construction d'un workflow sémantique pour l'interprétation sismique

L'objectif à travers la construction préalable d'un workflow sémantique est de permettre aux experts de la modélisation géologique de construire des workflows de manière plus naturelle. Ainsi, la composition de services sémantiques en workflows sémantiques est très proche de la manière avec laquelle les experts en modélisation géologique décrivent les processus. La figure 6.8 montre une représentation candidate pour un processus en interprétation sismique tel qu'il est décrit par des experts en modélisation géologique (partie supérieure, voir chapitre 3 section 4.1). Elle montre également comment il peut être décrit comme un workflow sémantique résultant d'une composition de services sémantiques (partie inférieure).

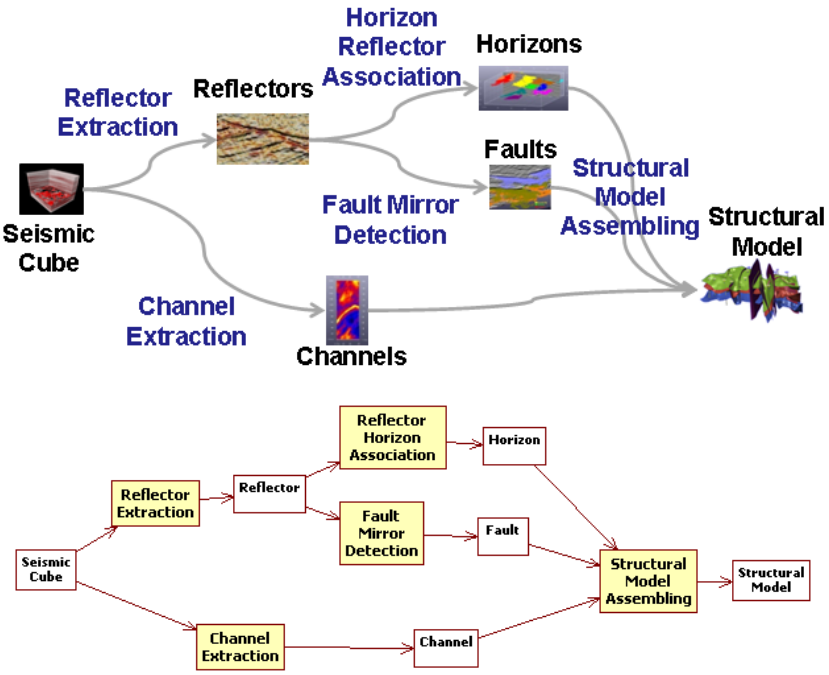


FIG. 6.8 – Workflow sémantique potentiel simplifié d'une interprétation sismique.

Cette description en workflow sémantique est conforme au modèle représenté sur la figure 4.12 (chapitre 4). Nous présentons ainsi sur la figure 6.9 le workflow sémantique en tant qu'instance de ce modèle (partie gauche) et sa description sous forme infixée correspondante (partie droite).

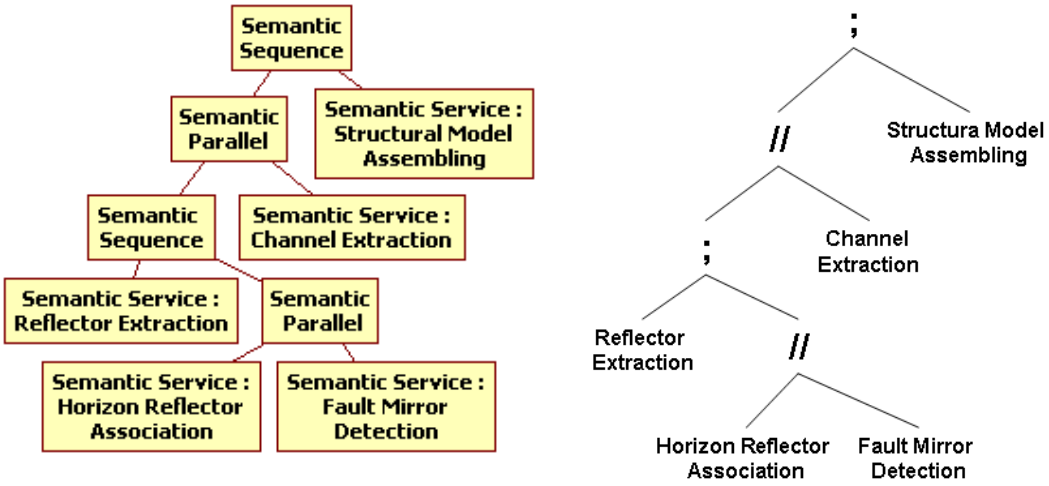


FIG. 6.9 – Description d'un workflow sémantique en tant qu'instance de son modèle et sous forme infixée.

4.2 Composition sémantique en interprétation sismique

Les services sémantiques communiquent entre eux à travers leurs entrées et sorties sémantiques. Sur la figure 6.10, nous montrons la condition qui permet l'interopérabilité au niveau sémantique. Ainsi, si deux services sémantiques *SS1* et *SS2* sont composés en séquence et que *SS2* a besoin du résultat rendu par *SS1* alors le concept en sortie du service sémantique *SS1* doit être subsumé par le concept en entrée du service sémantique *SS2*.

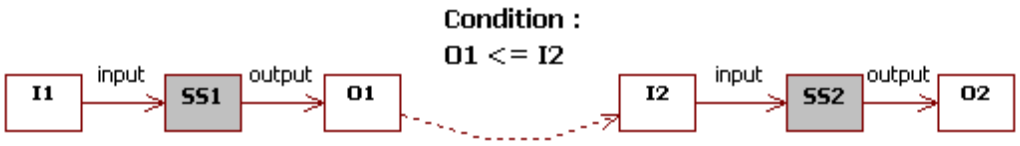


FIG. 6.10 – Exemple d'interopérabilité au niveau sémantique.

D'autres vérifications sur la composition peuvent être faites sur les propriétés fonctionnelles telles que la précédence d'un service sémantique par un autre. Enfin, certaines propriétés fonctionnelles sont critiques dans la validation de la composition telles que la propriété qui renseigne sur les services sémantiques de contrôle de qualité d'un service sémantique donné.

4.3 Mise à jour de l'ontologies des services de l'interprétation sismique par addition de services sémantiques composés

Le workflow sémantique est conceptuellement nouveau. Ainsi, il correspond à un nouveau concept de service sémantique. Le workflow que nous avons présenté sur la figure 6.8 réalise une construction de modèle structural. Ainsi, un nouveau concept de service sémantique émerge et qu'on dénomme *StructuralModelBuilding*. L'ontologie des services dans le domaine d'interprétation sismique peut être enrichie avec le nouveau concept. La figure 6.11 représente l'ontologie exposée dans la figure 6.5 enrichie par le service sémantique *StructuralModelBuilding*.

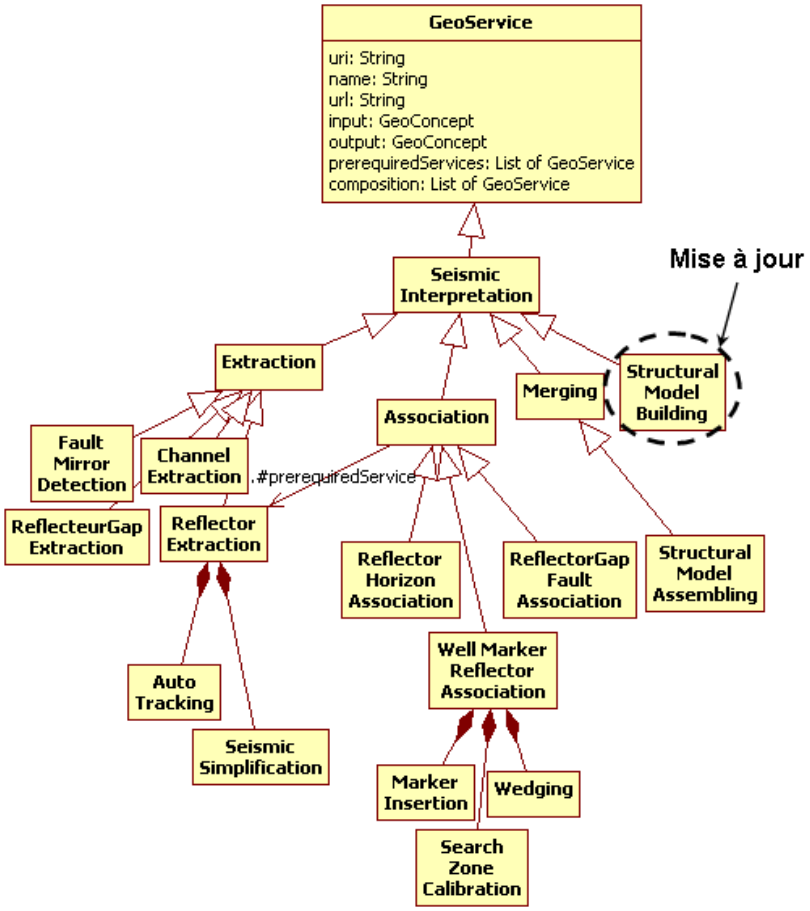


FIG. 6.11 – Fragment de l'ontologie des services pour l'interprétation sismique enrichie du service sémantique *StructuralModelBuilding*.

Quand un workflow sémantique est ajouté en tant que service sémantique, ses propriétés fonctionnelles doivent être calculées. Par exemple, l'entrée sémantique pour le workflow sémantique *StructuralModelBuilding* est le concept *SeismicCube*, et sa sortie sémantique est un *StructuralModel* et ce workflow n'a pas de service sémantique pré-requis.

4.4 Construction d'un workflow informatique pour l'interprétation sismique

A partir du workflow sémantique exposé sur la figure 6.9, il est possible d'obtenir un workflow informatique. Cela est réalisé en sélectionnant pour chaque service sémantique un service informatique particulier parmi ceux qu'il indexe. Dans notre cas, nous avons sélectionné pour le service *ReflectorExtraction* le Web service décrit en WSDL *getRef.wsdl*. Pour le service sémantique *ChannelExtraction*, nous avons sélectionné le workflow informatique décrit en BPEL *ChenauxWF.bpel*, etc. Ainsi, après avoir effectué la sélection pour tous les services sémantiques, nous obtenons le workflow informatique exposé dans la figure 6.12.

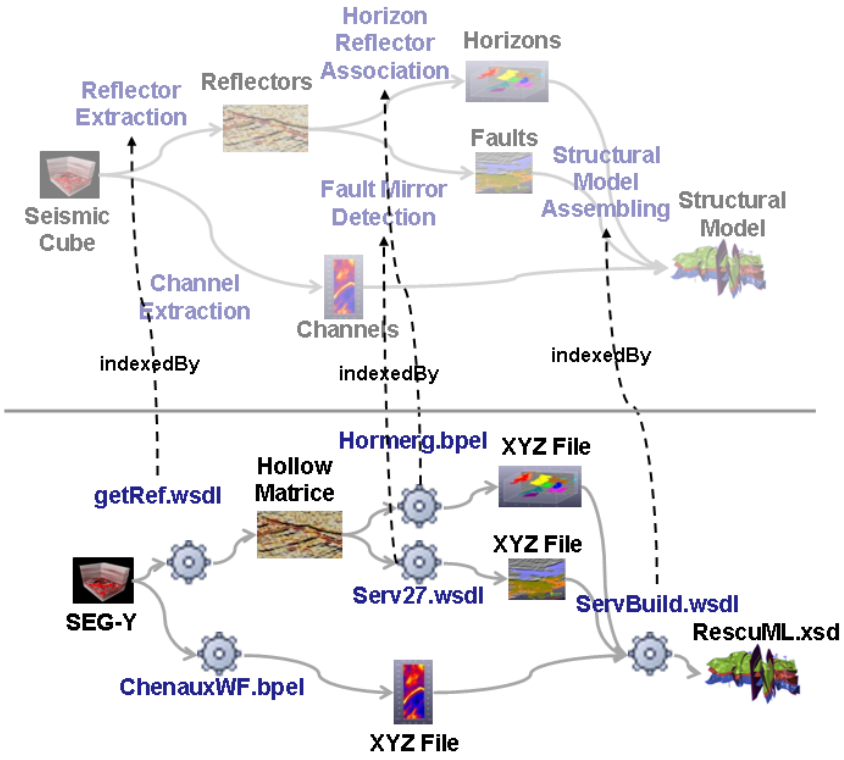


FIG. 6.12 – Workflow informatique d'interprétation sismique dérivé de la sélection de services et de workflows informatiques pour chaque service sémantique.

Dans cette figure, nous commençons à partir d'une instance de modèle de données en SEG-Y qui représente une image 3D. Ensuite, le Web service *getRef.wsdl* est exécuté pour obtenir une instance de modèle de données *HollowMatrix*. Ensuite, le workflow *Hormerg.bpel* ainsi que le service Web *Serv27.wsdl* sont exécutés en parallèle afin d'obtenir deux instances de modèle de données en XYZFile qui représentent respectivement des horizons et des failles. En parallèle à toutes ces exécutions, le workflow *ChenauxWF.bpel* est exécuté pour obtenir une instance de modèle de données en XYZFile qui représente les chenaux. Enfin, les trois instances de modèle de données en XYZFile qui représentent les trois objets géologiques sont combinés pour obtenir

une instance de modèle de données en RescuML qui représente le modèle structural.

Bien que les services sémantiques forment une composition en un workflow sémantique valide, il peut ne pas en être de même pour le workflow informatique obtenu. En effet, si dans l'exemple précédent, le workflow informatique *ChenauxWF.bpel* avait un modèle de données en *IJKFile* comme sortie par exemple, il ne serait pas directement composable avec le service informatique suivant qui prend en entrée une instance de modèle de données en *XYZFile*. Dans ce cas, un service informatique dit d'*adaptation* qui permettrait de convertir une instance de modèle de données en *XYZFile* en une instance de modèle de données en *IJKFile* est recherché dans la base de données (par exemple le service *Convert12.wsdl*). Il est alors automatiquement ajouté au workflow informatique global comme cela est montré dans la Figure 6.13. Si en revanche un tel service n'existe pas, le service composé ne peut être construit sans conversion manuelle.

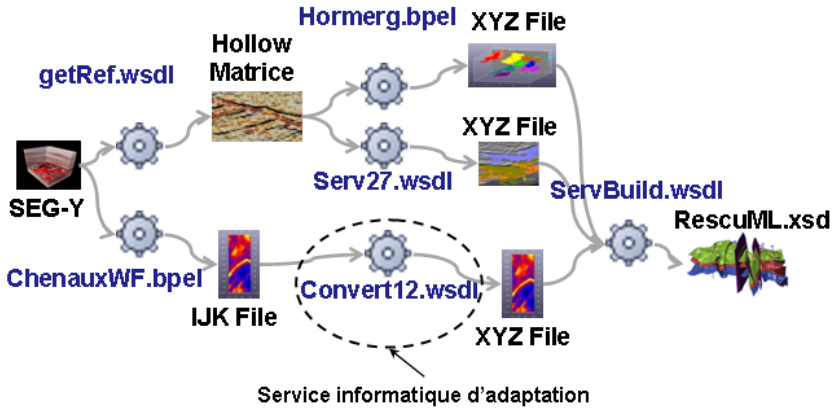


FIG. 6.13 – Autre workflow informatique d'interprétation sismique dérivé de la sélection de services et de workflows informatiques pour chaque service sémantique.

5 Conclusion

Dans ce chapitre, nous avons montré l'application de notre proposition sur le domaine de la modélisation géologique. Nous avons ainsi explicité la sémantique des modèles de données utilisés en modélisation géologique, des services et des workflows informatiques qui les manipulent en créant des ontologies de domaine et de services. Ceci nous a permis à cette étape l'indexation sémantique de ces modèles, services et workflows a travers leur caractérisation ontologique. Une telle indexation permet leur réutilisation en facilitant entre autre leur découverte. Nous avons présenté les ontologies de modélisation géologique et avons montré que les concepts de ces ontologies indexaient sémantiquement les modèles de données. L'indexation et la caractérisation sémantique permettent également la conception de nouveaux workflows informatiques en modélisation géologique. Nous avons présenté un exemple de construction de

workflow sémantique proche de la conception de processus de fonction décrits par les experts du domaine. A partir de ce workflow sémantique, il est possible de construire un workflow informatique en associant un service ou un workflow informatique à chaque service sémantique composant le workflow sémantique global.

Dans le prochain chapitre, nous présentons l'utilisation d'OntoDB en tant qu'entrepôt sémantique pour stocker les modèles de données, les services et les workflows informatiques en modélisation géologique.

Chapitre 7

Implémentation de notre approche sur l'interprétation sismique

Sommaire

1 Introduction 155

2 Persistance des données de modélisation géologique sur OntoDB . . . 155

2.1 Création des concepts liés à la modélisation géologique 155

2.2 Indexation des modèles de données liés à la modélisation géologique 156

3 Implémentation des services nécessaires à la modélisation géologique sur OntoDB 157

3.1 Création d'ontologies de services de modélisation géologique . . 158

3.2 Indexation de services et de workflows informatiques en modélisation géologique 158

4 Exploitation de l'implémentation en modélisation géologique sur OntoDB 160

4.1 Recherche de services à travers leurs attributs sémantiques 161

4.2 Recherche de services sémantiques à travers des services et workflows informatiques indexés 162

4.3 Recherche générale de services ou de workflows informatiques en modélisation géologique 162

4.4 Recherche de services ou de workflows informatiques de modélisation géologique avec restriction sur propriétés 163

4.5 Autres requêtes 165

5 Conclusion 165

Résumé. Nous montrons dans ce chapitre comment nous avons utilisé et adapté la BDBO OntoDB pour rendre persistants les modèles de données, les services et

les workflows de la modélisation géologique manipulés. Nous verrons que cette implémentation permet l'écriture de requêtes sémantiques qui facilitent la gestion de services et de workflows par les acteurs de la modélisation géologique.

1 Introduction

Dans le chapitre précédent, nous avons appliqué notre proposition au domaine de la modélisation géologique. Dans ce domaine, les données manipulées sont volumineuses. Aussi, le problème de passage à l'échelle est important. Pour répondre à ce problème, nous proposons de rendre persistantes les informations utiles à notre approche au sein d'une BD. Notre approche reposant sur l'utilisation d'ontologies, nous avons choisi d'utiliser une BD spécialisée pour le stockage d'ontologies appelée BDBO. Plus précisément, nous avons choisi OntoDB développée et disponible au sein de notre laboratoire. Si cette BDBO permet le stockage des ontologies de domaine conçues en modélisation géologique, elle ne permet pas initialement le stockage de services, de workflows, des modèles de données manipulés ainsi que des indexations sémantiques nécessaires. Aussi, nous montrons dans ce chapitre comment nous avons utilisé et adapté OntoDB pour implanter notre approche dans un environnement persistant. Ce travail de persistance en base permet de concevoir des requêtes OntoQL qui exploitent ces informations sur OntoDB pour aider les acteurs de la modélisation géologique dans leur gestion des modèles, des services et des workflows qu'ils utilisent.

2 Persistance des données de modélisation géologique sur OntoDB

Nous montrons dans cette section comment les données nécessaires au processus de modélisation géologique ainsi que les concepts sémantiques qui en définissent le sens sont matérialisées. Nous avons montré dans la section 2 du chapitre 5 (voir figure 6.2), comment nous spécialisons la classe standard *#DomainOntologyConcept* pour permettre la création de concepts dans une ontologie de domaine. Dans cette partie, nous montrons comment nous créons dans OntoDB les concepts nécessaires à la modélisation géologique par des instructions OntoQL. Nous montrons aussi comment nous indexons sémantiquement dans OntoDB les modèles de données créés pour représenter les données liées à la modélisation géologique.

2.1 Création des concepts liés à la modélisation géologique

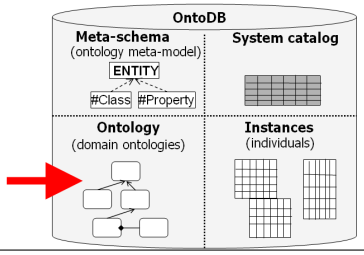
Pour représenter les ontologies de domaine décrivant la modélisation géologique sur OntoDB, nous créons les concepts qui les composent à travers des instructions OntoQL.

Considérons l'ontologie de géologie définie dans la section 2.2 du chapitre 6. Nous montrons dans la figure 7.1 la création du concept *Reflector* de cette ontologie. Il est créé avec un certain nombre de propriétés qui caractérisent des réflecteurs et les propriétés qui caractérisent les modèles de données qui les représentent :

– la propriété *URI* permet de renseigner de manière unique une instance de ce concept ;

- la propriété *name* renseigne le nom du format de description du modèle de donnée ajouté ;
- la propriété *URL* permet de représenter le lien vers la description du modèle de données ajouté. Ces trois premières propriétés permettent de caractériser les modèles de données qui seront instances du concept ;
- la propriété *hasAmplitude* représente la longueur d'un réflecteur ;
- la propriété *hasThickness* représente la profondeur d'un réflecteur. Ces deux dernière propriétés permettent de caractériser des instances de réflecteurs.

Requête d'enrichissement des ontologies de modélisation géologique.




```
CREATE #DomainOntologyConcept GeoConcept (
 PROPERTIES (
    URI STRING,
    name STRING,
    URL STRING))
...
CREATE #DomainOntologyConcept Reflector UNDER
ComponentSurface (
 PROPERTIES (
    hasAmplitude float,
    hasThickness float))
...
CREATE EXTENT OF Reflector
(URI, name, URL, hasAmplitude, hasThickness)
```

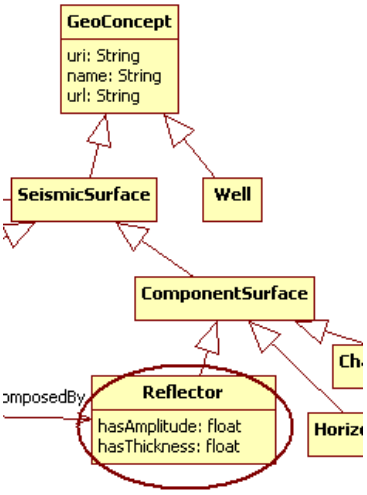


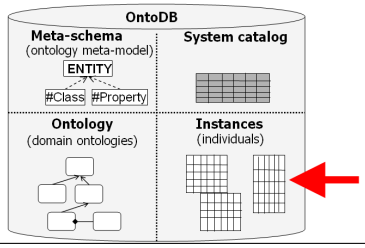
FIG. 7.1 – Création du concept *Reflector* dans l'ontologie de la sismique.

2.2 Indexation des modèles de données liés à la modélisation géologique

L'indexation sémantique d'un modèle de données se fait en instanciant un concept de l'ontologie de modélisation géologique. Par exemple, *XYZFile* est un modèle de données utilisé dans la modélisation géologique. Il permet de représenter un nuage de points, c'est-à-dire un ensemble de points par leurs coordonnées. Ce modèle de données peut représenter des horizons. Ainsi, il est potentiellement indexable par le concept *Horizon* de l'ontologie de modélisation géologique. Nous représentons l'indexation de ce modèle par son insertion en tant qu'instance du concept *Horizon* sur la figure 7.2. Sur cette figure, le modèle de données *XYZFile* possède

l'URI *uri41* et la description de son schéma se trouve à l'adresse physique */ifp.fr/xyzfile.txt*.

Requête d'indexation d'un modèle de données.




```
INSERT INTO Horizon (
    URI, name, URL)
 VALUES (
   'uri41', 'XYZFile', '/ifp.fr/XYZFile.txt')
```

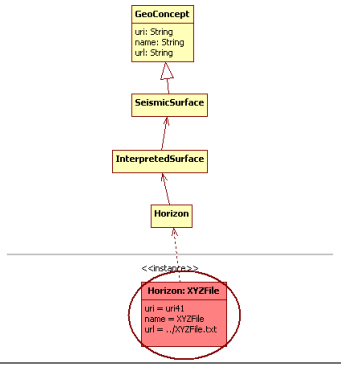


FIG. 7.2 – Instanciation du concept ontologique *Horizon*.

Ayant présenté dans cette section la représentation de la partie statique : les données de la modélisation géologique, nous nous intéressons dans la prochaine section à la représentation de la partie dynamique : les services et les workflows informatiques en modélisation géologique.

3 Implémentation des services nécessaires à la modélisation géologique sur OntoDB

Dans cette partie, nous montrons comment nous utilisons OntoDB au travers d'instructions OntoQL pour stocker les services et workflows informatiques de modélisation géologique ainsi que les concepts d'ontologies de services, c'est-à-dire les services sémantiques qui en définissent le sens. Nous avons déjà défini dans la section 3.1 du chapitre 5 l'enrichissement du modèle classique d'ontologies pour permettre la caractérisation de services sémantiques par les attributs qui leur sont propres. Dans ce qui suit, nous montrons comment nous créons les concepts d'ontologies de services de modélisation géologique par des création de classes de type *#SemanticService* sur OntoDB. Nous montrons également comment nous décrivons et indexons les services et les workflows informatiques par instanciation de services sémantiques sur OntoDB. De plus, les travaux effectués sur les performances d'OntoDB, présentés dans [Dehainsala et al., 2007], assurent le passage à l'échelle.

3.1 Création d'ontologies de services de modélisation géologique

Pour représenter les ontologies de services en modélisation géologique sur OntoDB, nous créons les services sémantiques qui les composent à travers des instructions OntoQL. Pour créer des services sémantiques de modélisation géologique, nous créons des classes de type *#SemanticService* dans OntoDB. Considérons l'ontologie de services définie dans la section 3.2 du chapitre 6 (voir figure 6.5). Nous montrons dans la figure 7.3 la création du service sémantique *ReflectorExtraction* de cette ontologie.

Dans cette figure, ce service sémantique a comme entrée le concept sémantique *SeismicCube* de l'ontologie de domaine de modélisation géologique et en sortie le concept sémantique *Reflector* conformément à la figure 6.6 du chapitre 6. Une instance potentielle d'un service sémantique est tout service ou workflow informatique capable de remplir une fonction décrite par ce même service sémantique. Le service sémantique *ReflectorExtraction* définit des propriétés qui caractérisent les services et les workflows qui réalisent des extractions de réflecteurs :

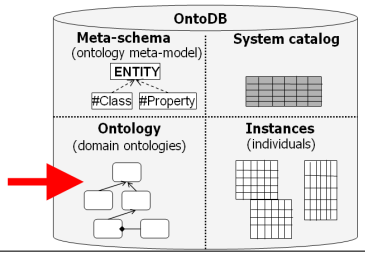
- la propriété *URI* permet de renseigner de manière unique une instance de ce service sémantique ;
- la propriété *description* renseigne le nom du service ou du workflow informatique ajouté ;
- la propriété *URL* permet de représenter le lien vers la description du service ou du workflow informatique ajouté ;
- la propriété *inputsModels* permet de représenter les modèles de données référencés en entrée dans la description du service ou du workflow informatique ;
- la propriété *outputsModels* permet de représenter les modèles de données référencés en sortie dans la description du service ou du workflow informatique.

D'autres propriétés peuvent être créées telles que le concepteur du service ou du workflow (*creator*), les droits d'utilisation (*rights*) ou la date de création (*date*).

3.2 Indexation de services et de workflows informatiques en modélisation géologique

L'indexation sémantique des services et des workflows informatiques de modélisation géologique se fait en instanciant un service sémantique de l'ontologie de services associée. Nous illustrons dans la figure 7.4 un exemple d'indexation de service Web *refGet* par le service sémantique *ReflectorExtraction*. Les informations utiles à son exécution et à sa gestion informatique sont disponibles dans les valeurs des propriétés définies. Ainsi, ce service Web *getRef* est défini par l'URI *uri2*. Il est décrit dans le langage de description de service Web *WSDL*. Sa description se trouve physiquement à l'adresse *http://.../refGet.wsdl*. La description du modèle de données en entrée de ce service informatique se trouve à l'adresse physique */ifp.fr/SEG-Y.txt*. La description du modèle de données en sortie de ce service informatique se trouve à l'adresse physique */ifp.fr/XYZFile.txt*.

Requête d'enrichissement de l'ontologie des services en modélisation géologique.




```
CREATE #SemanticService GeoService (
 DESCRIPTOR (
    #input = ARRAY(
      SELECT doc.#oid
      FROM #DomainOntologyConcept doc
      WHERE doc.#name='GeoConcept')
    #output = ARRAY(
      SELECT doc.#oid
      FROM #DomainOntologyConcept doc
      WHERE doc.#name='GeoConcept')
 PROPERTIES (
    URI STRING,
    name STRING,
    description STRING,
    URL STRING,
    inputsModels REF (GeoConcept) ARRAY,
    outputsModels REF (GeoConcept) ARRAY))
...
CREATE #SemanticService ReflectorExtraction
UNDER Extraction(
 DESCRIPTOR (
    #input = ARRAY(
      SELECT doc.#oid
      FROM #DomainOntologyConcept doc
      WHERE doc.#name='SeismicCube')
    #output = ARRAY(
      SELECT doc.#oid
      FROM #DomainOntologyConcept doc
      WHERE doc.#name='Reflector'))
...
CREATE EXTENT OF ReflectorExtraction
(URI, name, description, URL, inputsModels,
outputsModels)
```

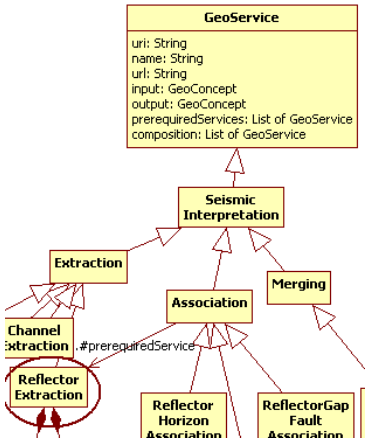


FIG. 7.3 – Création du service sémantique *ReflectorExtraction*.

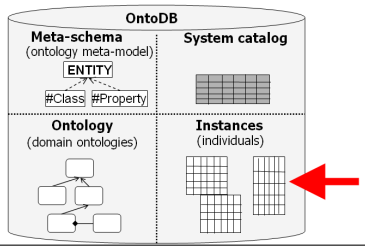
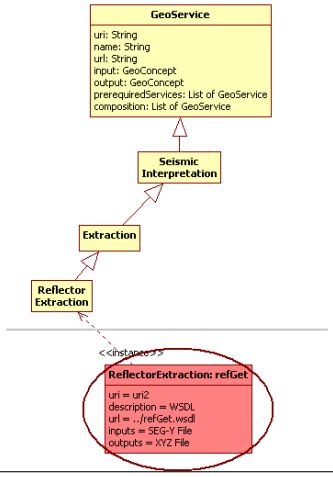
Requête d'indexation d'un service Web en modélisation géologique.	
INSERT INTO ReflectorExtraction (URI, name, description, URL, inputsModels, outputsModels) **VALUES** ('uri2', 'getRef', 'WSDL', 'http://.../refGet.wsdl', (**SELECT** oid **FROM** GeoConcept **WHERE** URL = '/ifp.fr/SEG-Y.txt'), (**SELECT** oid **FROM** GeoConcept **WHERE** URL = '/ifp.fr/XYZFile.txt'))	

FIG. 7.4 – Instantiation d'un service Web par le service sémantique *ReflectorExtraction*.

Les sections précédentes ont permis la persistance des modèles de données, des services et des workflows informatiques ainsi que des descriptions ontologiques qui en définissent le sens au sein d'un entrepôt sémantique : OntoDB. Dans la section suivante, nous mettons en évidence l'apport en gestion de services et de workflows en modélisation géologique et sa traduction en requêtes OntoQL.

4 Exploitation de l'implémentation en modélisation géologique sur OntoDB

Les géologues, les géophysiciens et autres spécialistes de la modélisation géologique utilisent des systèmes informatiques en ayant une connaissance *a priori* des modèles de données, des services et des workflows informatiques qu'ils manipulent. Cependant, vu qu'il ne connaissent pas tous les modèles et services et que leurs connaissances techniques sont limitées, cela cause une grande perte de temps et d'efficacité. Des besoins en gestion des services et des workflows conçus en modélisation géologiques se font ainsi sentir parmi lesquels :

– rechercher un service ou un workflow informatique sémantiquement et non par son nom ou sa description syntaxique ;
– rechercher un service ou un workflow informatique à partir des concepts qu'il manipulent

en entrée et en sortie ;
- trouver un service ou un workflow informatique remplissant la même fonction qu'un autre pour le remplacer ;
- retrouver la fonction remplie par un service ou un workflow donné, etc.

Notre approche permet de stocker au sein de la même BDBO :
- les descriptions syntaxiques des modèles de données issus de la modélisation géologique ;
- les descriptions syntaxiques des services et des workflows informatiques utilisés pour traiter les données en modélisation géologique ;
- les ontologies de domaine et de services en modélisation géologique ;
- les indexations sémantiques qui lient les différentes descriptions syntaxiques aux descriptions ontologiques (sémantique).

Ce stockage permet ainsi une gestion plus naturelle et plus adaptée des modèles, des services et workflows traduisibles en requêtes OntoQL posées sur OntoDB. Il est à noter que ces requêtes sont toutefois complexes et qu'elles sont à cacher aux acteurs de la modélisation géologique. Cette complexité réside dans le fait qu'elles interrogent la base de données à base ontologique à différents niveaux (méta-modèle ou modèle d'ontologies), différentes piles (pile de données ou pile de service) et différentes représentations conceptuelles (modèles et ontologies). Dans cette section, nous présentons un ensemble de requêtes OntoQL pertinentes pour la gestion de services et workflows en modélisation géologique.

4.1 Recherche de services à travers leurs attributs sémantiques

Notre approche permet la recherche des services sémantiques, et particulièrement pour la modélisation géologique, à travers leurs attributs fonctionnels sémantiques. Cela revient à retrouver une fonction ou un processus composé de fonctions suivant les objets qu'il manipule ou ce qui le caractérise sémantiquement de manière plus générale. Ainsi, des services sémantiques décrits dans une ontologie de services de modélisation géologique peuvent être retrouvés suivant les concepts ontologiques qu'ils prennent en entrée. La recherche est permise également sur des services sémantiques qui peuvent potentiellement précéder ou succéder à l'exécution d'un service sémantique donné.

Requête. Retrouver les services sémantiques de modélisation géologique qui prennent en entrée le concept ontologique *Reflector*.

```
SELECT SS.#name
FROM #SemanticService SS, #DomainOntologyConcept DOC
WHERE DOC.#name = 'Reflector'
 AND DOC.#oid IN SS.#inputs
```

Explication. Rechercher les noms (`SS.#name`) des services sémantiques (`SS`) qui ont comme entrée sémantique (`DOC. #oid IN SS. #inputs`) le concept ontologique *Reflector* (`DOC.`

`#name = 'Invoice'`).

Requête. Retrouver les services sémantiques qui sont les successeurs potentiels du service sémantique *ReflectorExtraction*.

```
SELECT SS2.#name
FROM #SemanticService SS1, #SemanticService SS2
WHERE SS1.#name = 'ReflectorExtraction'
 AND SS1.#oid IN SS2.#prerequiredServices
```

Explication. Rechercher les noms (`SS2.#name`) des services sémantiques (`SS2`) qui succèdent (`SS1.#oid IN SS2.#prerequiredServices`) au service sémantique *ReflectorExtraction* (`SS1.#name = 'ReflectorExtraction'`).

4.2 Recherche de services sémantiques à travers des services et workflows informatiques indexés

L'indexation d'un service ou d'un workflow informatique de modélisation géologique par des services sémantiques permet de savoir quelle fonction il réalise (soit parfaitement soit une fonction subsumante).

Requête. Retrouver le service sémantique indexant le service ou le workflow informatique de modélisation géologique dont le nom est *RefExtr*.

```
SELECT SS.#name
FROM #SemanticService SS, GeoService GS
WHERE TYPEOF(GS).#oid = SS.#oid
 AND GS.name LIKE 'RefExtr'
```

Explication. GeoService est le service sémantique racine de modélisation géologique, qui subsume tous les autres services sémantiques de modélisation géologique. `TYPEOF(GS)` indique qu'on recherche le concept ontologique duquel le service informatique `GS` est instance. Ainsi, il s'agit de rechercher le ou les noms (`SS.#name`) du ou des services sémantiques (`SS`) qui indexent (`TYPEOF(GS).#oid = SS.#oid`) le service ou workflow informatique dont le nom est *RefExtr* (`GS.name LIKE 'RefExtr'`).

4.3 Recherche générale de services ou de workflows informatiques en modélisation géologique

La recherche sémantique de services ou de workflows informatiques utilisés pour la modélisation géologique est rendue possible par notre approche. En effet, les services et les workflows

informatiques de modélisation géologique sont indexés par des services sémantiques en instanciant ces derniers. Rechercher sémantiquement des services ou des workflows informatiques revient donc à rechercher un service sémantique particulier dont la fonction nous intéresse. Ensuite, il s'agit de chercher dans ses instances les services et les workflows informatiques de modélisation géologique qui réalisent la fonction cible.

Il est à noter que les instances d'un service sémantique donné sont non seulement ses instances propres, mais également les instances des services sémantiques qui le subsument. Ainsi, il est possible de rechercher des services ou des workflows informatiques soit par les fonctions exactes qu'ils réalisent, soit par des fonctions plus spécialisées.

Requête. Retrouver tous les services ou workflows informatiques qui réalisent des extractions de réflecteurs.

```
SELECT RES.uri, RES.url
FROM ReflectorExtraction RES
```

Explication. Rechercher les identifiants (`RES.uri`) ainsi que les chemins (`RES.url`) des services ou des workflows informatiques qui sont instances du service sémantique *ReflectorExtraction* (`FROM ReflectorExtraction RES`)

Requête. Retrouver tous les services ou workflows informatiques qui réalisent *exactement* des extractions de réflecteurs.

```
SELECT RES.uri, RES.url
FROM #SemanticService SS, ReflectorExtraction RES
WHERE TYPEOF(RES).#oid = SS.#oid
 AND SS.#name = 'ReflectorExtraction'
```

Explication. Rechercher les identifiants (`RES.uri`) ainsi que les chemins (`RES.url`) des services ou des workflows informatiques, instances du service sémantique *ReflectorExtraction* (`FROM ReflectorExtraction RES`) et qui ne sont instances que de ce service sémantique (`TYPEOF(RES).#oid = SS.#oid AND SS. #name = 'Reflector Extraction'`).

4.4 Recherche de services ou de workflows informatiques de modélisation géologique avec restriction sur propriétés

L'objectif de cette recherche est de pouvoir retrouver des services ou des workflows informatiques réalisant une fonction donnée et manipulant des données conformes à des concepts donnés. Il est ainsi également possible de restreindre une recherche sur les attributs des services sémantiques. Cette recherche est réalisée en vérifiant ce que les services sémantiques recherchés ont comme attributs sémantiques (entrée sémantique, sortie sémantique, service sémantique pré-requis, etc.).

Requête. Retrouver tous les services ou les workflows informatiques qui réalisent une extraction de réflecteurs ou une fonction spécialisée et qui ont comme entrée le concept ontologique *SeismicCube*.

```
SELECT CI.uri, CI.url
FROM #SemanticService SS, #DomainOntologyConcept DOC, SS AS CI,
 ReflectorExtraction RES
WHERE DOC.#name = 'SeismicCube'
 AND DOC.#oid IN SS.#inputs
 AND RES.uri = CI.uri
```

Explication. L'interrogation conjointe des ontologies et des instances des concepts qui les composent est possible. Le constructeur **AS** permet en effet de calculer les instances de classes déterminées dynamiquement. Ainsi, la requête retrouve les services sémantiques `#Semantic Service SS` qui ont comme entrée le concept ontologique *SeismicCube* (`DOC.#name LIKE 'SeismicCube' AND DOC.#oid IN SS.#inputs`). Ensuite, toutes les instances des services sémantiques obtenus sont parcourues (`SS AS CI`). Enfin, seuls les services et workflows informatiques qui réalisent une extraction de réflecteur (`ReflectorExtraction RES`) sont gardés (`RES.uri = CI.uri`).

Un autre objectif est de pouvoir rechercher des services ou des workflows informatiques réalisant une fonction donnée et manipulant des modèles de données particuliers. Il est ainsi également possible de restreindre une recherche sur les propriétés des services ou des workflows informatiques. Cette recherche est réalisée en recherchant les instances d'un service sémantique particulier qui vérifient une ou plusieurs contraintes sur les propriétés.

Requête. Retrouver tous les services ou workflows informatiques qui réalisent une extraction de réflecteurs et qui produisent un modèle de données en sortie du type *XYZFile*.

```
SELECT RES.uri, RES.url
FROM ReflectorExtraction RES, GeoConcept GC
WHERE GC.oid IN RES.outputsModels
 AND GC.name = 'XYZFile'
```

Explication. Cette requête retrouve tous les identifiants des services et des workflows informatiques qui réalisent des extractions de réflecteurs `RES.uri` ainsi que leurs chemins `RES.url` et dont le modèle de données en sortie (`GC.oid IN RES.outputsModels`) est *XYZFile* (`GC.name = 'XYZFile'`).

4.5 Autres requêtes

D'autres requêtes sémantiques existent parmi lesquelles l'adaptativité que nous avons définie dans la section 3.5 du chapitre 4 et qui est indépendante du domaine d'application. L'adaptativité est le fait de retrouver les services ou les workflows informatiques qui remplissent des fonctions similaires, c'est-à-dire des services ou des workflows informatiques qui sont indexés par le même service sémantique. Dans notre approche, ceci s'interprète comme des services ou des workflows informatiques qui sont indexés par le même service sémantique ou un service sémantique subsumé.

Requête. Retrouver la fonction réalisée par le service ou workflow informatique dont l'uri est *uri15* et les services ou les workflows informatiques qui remplissent une fonction équivalente.

```
SELECT SS.#name, GS1.uri, GS1.url
FROM GeoService GS1, GeoService GS2, #SemanticService SS
WHERE (TYPEOF(GS1).#oid = TYPEOF(GS2).#oid)
 AND (GS2.uri = 'uri15')
 AND NOT (GS1.oid = GS2.oid)
 AND (SS.#oid = TYPEOF(GS2).#oid)
```

Explication. Cette requête retrouve le service sémantique `SS.#name` correspondant au service informatique dont l'URI est *uri15* (`SS.#oid = TYPEOF(GS2).#oid`). Elle retrouve également tous les identifiants des services informatiques `GS1.uri` ainsi que les liens physiques vers leurs descriptions `GS1.url` qui sont indexés par le même service sémantique (`TYPEOF(GS1).#oid = TYPEOF(GS2).#oid`) que le service ou workflow informatique dont l'uri est *uri15* (`GS2.uri = 'uri15'`) et qui sont bien évidemment différents de ce même service (**NOT** `(GS1.oid = GS2.oid)`).

5 Conclusion

Dans ce chapitre, nous avons montré l'implémentation de notre proposition appliquée au domaine de la modélisation géologique. Plus concrètement, nous avons montré comment créer les concepts des ontologies dans ce domaine. Nous avons indexé les modèles de données utilisés dans la modélisation géologique par les concepts des ontologies de domaine. Nous avons également créé dans OntoDB des ontologies de services au travers de services sémantiques. Enfin, nous avons indexé les services et workflows informatiques de la modélisation géologique par les services sémantiques. La BDBO résultante est considérée comme un entrepôt sémantique contenant toutes les descriptions des services et des workflows informatiques, des modèles de données de la modélisation géologique, et ainsi que des ontologies, c'est-à-dire les descriptions sémantiques qui en définissent le sens.

Notre objectif est d'exploiter cet entrepôt sémantique pour supporter des requêtes complexes. Nous avons montré des exemples de requêtes écrites en OntoQL qui permettent de questionner la base de données à base ontologique OntoDB. Toutefois, les experts en géologie pétrolière, les géologues, les géophysiciens, etc. ne sont pas des experts en bases de données. Il est donc difficile pour eux de poser de telles requêtes directement sur OntoDB. Dans le prochain chapitre, nous présentons le Geological Workflow Editor (GWE) qui représente le prototype logiciel que nous avons développé et qui intègre les implémentations que nous avons montrées. Le GWE permet à travers une interface utilisateur de cacher toutes les requêtes de type base de données et d'offrir une interface graphique qui permet la recherche de services sémantiques, de services et de workflows informatiques de modélisation géologique, de construire de nouveaux workflows, etc.

Chapitre 8

Geological Workflow Editor (GWE)

Sommaire

1 Introduction 169
2 Présentation générale du GWE 169
2.1 GWE sur OntoDB 169
2.2 Profil des utilisateurs 169
3 Réutilisation de services et de workflows existants 171
3.1 Gestion d'ontologies de modélisation géologique 173
3.1.1 Gestion d'ontologies de domaine 173
3.1.2 Gestion d'ontologies de services 174
3.2 Indexation de services et de workflows informatiques 177
3.3 Exploitation des services et des workflows représentés dans le GWE 178
3.4 Exécution de workflows informatiques 178
4 Création de nouveaux workflows 179
4.1 Construction de workflows sémantiques 181
4.2 Validation de workflows sémantiques 182
4.3 Enrichissement de l'ontologie des services 184
4.4 Association des services et workflows informatiques aux services sémantiques 184
4.5 Construction du workflow informatique 185
4.6 Indexation du workflow informatique obtenu 186
5 Conclusion 187

Résumé. Dans le chapitre précédent, nous avons montré les requêtes sémantiques, permises par notre approche, permettant de faciliter la gestion de services et de

workflows informatiques. Ces requêtes ne pouvant être écrites que par des informaticiens, nous avons développé un outil nommé *Geological Workflow Editor (GWE)*. Nous présentons cet outil dans ce chapitre en montrant comment il permet de mettre en œuvre et d'exploiter notre approche dans une interface conviviale.

1 Introduction

Dans les chapitres précédents, nous avons présenté notre approche basée sur l'indexation de services et de workflows informatiques à travers des caractérisations ontologiques. Nous avons également présenté notre approche de construction de workflows sémantiques et de génération de workflows informatiques. Nous l'avons implémentée en utilisant la structure de base de données à base ontologique OntoDB. Ceci nous a permis de valider cette approche. Cependant, cette implémentation ne peut pas être utilisée directement par des non-informaticiens. Or, le public ciblé par nos travaux sont les géologues et autres acteurs de la modélisation géologique. Ainsi, l'objectif que nous nous sommes fixé est de pouvoir développer une application logicielle qui exploite la capacité de description de notre approche de manière transparente aux utilisateurs en l'adoptant à notre domaine d'application : la modélisation géologique. Ainsi, le Geological Workflows Editor (GWE) représente un prototype logiciel capable d'assister les différents acteurs et experts en géologie tels que des géologues et des géophysiciens dans leurs tâches de modélisation géologique.

Ce chapitre est organisé comme suit. Dans la section 2, nous donnons une présentation générale du GWE. Nous présentons ensuite dans la section 3 la partie du GWE dans laquelle se fait la création des ontologies de domaine et de services et l'indexation des services et des workflows informatiques par leurs descriptions ontologiques. Dans la section 4, nous exposons la partie du GWE dans laquelle se fait la création des workflows sémantiques. Nous finissons par une conclusion sur les apports du GWE pour la modélisation géologique.

2 Présentation générale du GWE

2.1 GWE sur OntoDB

Le GWE est un logiciel que nous avons développé sous la plateforme *Eclipse* et connecté à la base de données à base ontologique OntoDB. La figure 8.1 illustre le lien entre le GWE et OntoDB. Dans cette figure, les utilisateurs accèdent via une interface graphique au GWE. Le GWE est connecté à OntoDB à travers les API OntoJDBC et OntoAPI présentées dans le chapitre 1. Ainsi, les utilisateurs qui ne sont pas familiarisés avec les langages de requêtes de bases de données peuvent effectuer leur gestion de services et de workflows informatiques sur OntoDB via le GWE de manière transparente. OntoDB est, quant à elle, utilisée comme un entrepôt pour stocker les modèles de données, les services et les workflows informatiques ainsi que les ontologies de domaine et de services qui en définissent le sens.

2.2 Profil des utilisateurs

Dans l'application que nous proposons, nous avons identifié quatre profils différents :

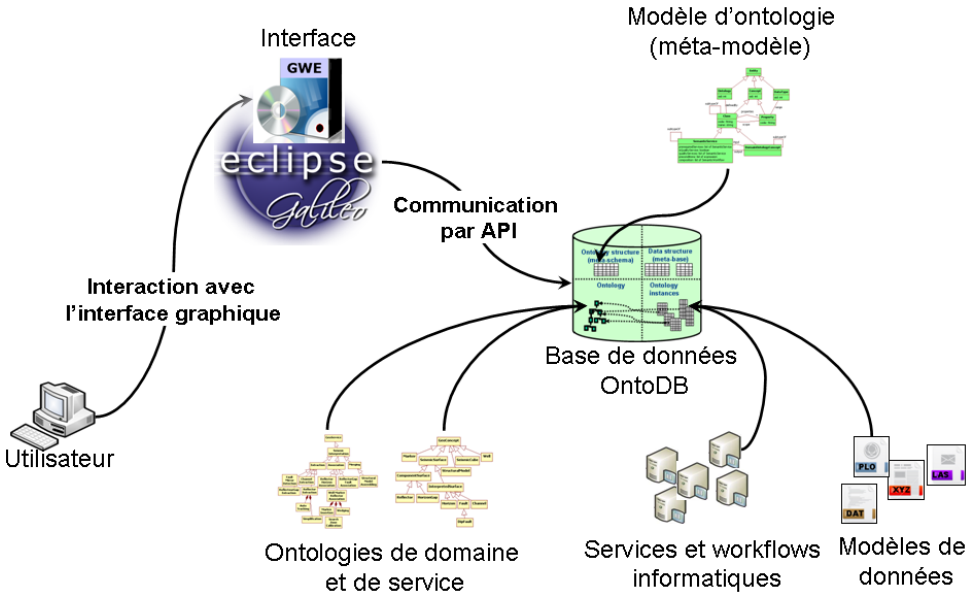


FIG. 8.1 – Connexion du GWE à OntoDB.

- *expert du domaine* : il définit les ontologies de domaine en créant les concepts ontologiques et en leur attribuant des propriétés. Il crée également les ontologies de services en leur associant les services sémantiques qui les composent ;
- *expert technique* : il indexe les modèles de données par les concepts des ontologies de domaine. Il indexe les services et les workflows informatiques par les services sémantiques ;
- *manager de workflows sémantiques* : il définit des workflows sémantiques issus de compositions de services sémantiques. À partir de cette description, le workflow sémantique devient conceptuellement lui-même un service sémantique et l'expert du domaine peut enrichir l'ontologie de services si le concept n'existe pas déjà. L'expert technique peut ensuite associer à chaque service sémantique qui compose le workflow sémantique un service ou un workflow informatique particulier et ainsi obtenir un workfow informatique qui sera à son tour indexé ;
- *utilisateur final* : il choisit un service ou un workflow sémantique. Ensuite, il sélectionne un service ou un workflow informatique qui est indexé par le service sémantique choisi ou bien déduit à partir du workflow sémantique choisi. Enfin, il peut exécuter le service ou le workflow informatique.

Ces profils utilisateurs identifiés ne sont pas pris en compte dans la version actuelle du GWE. Nous nous sommes en effet actuellement focalisés sur la validation de notre approche qui concerne l'indexation et la recherche de services. Dans la prochaine section, nous détaillons la partie du GWE pour l'indexation sémantique des services et des workflows informatiques.

3 Réutilisation de services et de workflows existants

Les services et logiciels de traitement de données de modélisation géologique sont créés pour des besoins particuliers. Les workflows informatiques qui sont obtenus en composant des services sont construits sans qu'une méthodologie particulière soit mise en œuvre. Ainsi, que ce soit pour les services ou bien pour les workflows informatiques, aucune ressource à l'exception de leurs noms, ne peut expliciter les fonctions qu'ils réalisent. Toutefois, ces services et ces workflows informatiques sont appelés à être réutilisés plusieurs fois. De plus, d'autres utilisateurs ont des tâches similaires et réutiliseraient des services et workflows informatiques existants à condition d'avoir une connaissance précise sur leurs fonctions. Grâce au GWE, il devient possible d'indexer sémantiquement des services et des workflows informatiques existants. Cette indexation est rendue possible en extrayant la sémantique des services et workflows de la modélisation géologique. La production de cette sémantique doit être réalisée par des experts en modélisation géologique en créant des ontologies de services, dont les concepts sont des services sémantiques permettant de caractériser sémantiquement en hiérarchies les services et workflows informatiques de la modélisation géologique. De cette manière, les recherches des services et workflows informatiques sont facilitées et passent par une recherche sur une couche sémantique et ensuite un filtrage sur les services et workflows informatiques indexés. La figure 8.2 résume les étapes sur le GWE qui permettent la réutilisation des services et des workflows dans le GWE.

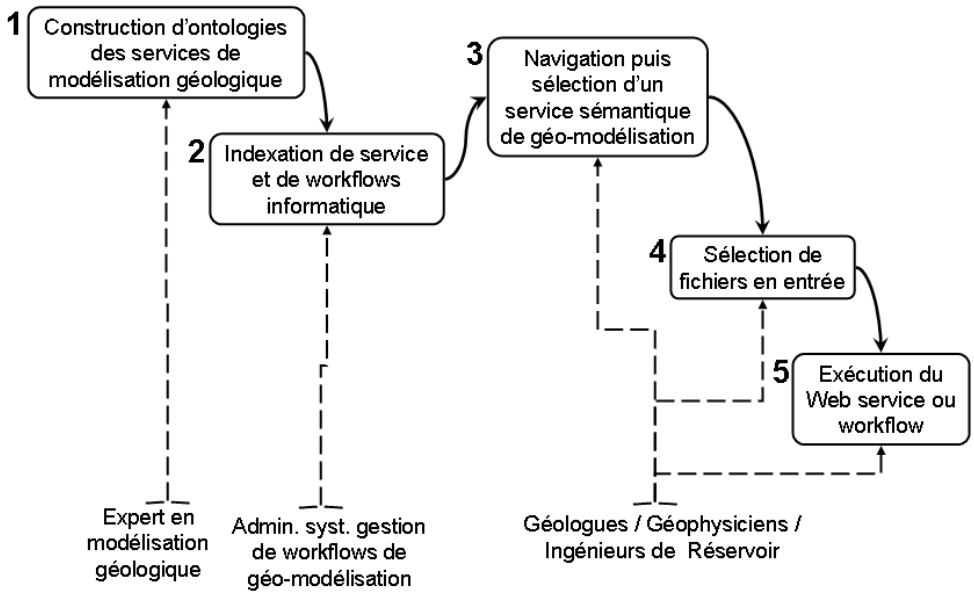


FIG. 8.2 – Réutilisation de services et de workflows informatiques.

Sur cette figure, nous voyons également les profils affectés aux types de tâches. Ainsi, l'étape (1) de la figure 8.2 consiste à construire la ou les ontologies de services, c'est-à-dire à extraire la sémantique des services et des workflows informatiques. Cette étape ne peut être

réalisée que par des experts dans le domaine de la modélisation géologique. Les ontologies obtenues peuvent évidemment évoluer. C'est pourquoi cette étape ne doit pas être vérrouillée pour passer à la suivante. Elle est au contraire amenée à être effectuée continuellement. Ensuite, l'étape (2) consiste à indexer les services et les workflows du domaine de la modélisation géologique. Cette indexation est réalisée par un administrateur de services informatiques. L'indexation est également faite continuellement. A chaque fois que des services ou des workflows informatiques sont créés ou construits, ils peuvent être indexés par des services sémantiques qui sont les concepts des ontologies de services précédemment créés. Ensuite, vient l'étape (3) qui représente l'exploitation de l'implémentation de l'approche à proprement parler. Les utilisateurs finaux, c'est-à-dire les géologues, les géophysiciens ou les ingénieurs de réservoir peuvent exploiter l'entrepôt sémantique pour réaliser des tâches de modélisation géologique. Ainsi, il peuvent rechercher dans les ontologies de services les services sémantiques qui représentent les tâches abstraites qu'ils souhaitent réaliser. Une fois cette recherche effectuée, et à chaque fois, un service sémantique est sélectionné. Ces utilisateurs peuvent ainsi choisir un service ou un workflow informatique parmi ceux indexés par le service sémantique. Un utilisateur final peut alors sélectionner des données (étape (4)) et lancer le service ou le workflow informatique sélectionné avec ces mêmes données en tant qu'entrées initiales (étape (5)). Les développements actuels du GWE couvrent les premières étapes qui correspondent à ce scénario. La cinquième étape, moins prioritaire pour la validation de nos travaux, est en cours de développement.

Nous montrons à présent les différentes zones du GWE participant à la réalisation du scénario de réutilisation de services et de workflows. Sur la figure 8.3, nous montrons une vue des zones d'édition et de visualisation du GWE pour ce premier scénario :

- dans la zone (A) de la figure, un espace est réservé pour la création de nouveaux concepts d'ontologies de domaine et de services afin de permettre l'évolution de ces ontologies ;
- dans la zone (B), un espace est réservé pour visualiser des représentations de modèles de données, de services et de workflows informatiques indexés par les concepts des ontologies de services ;
- dans la zone (C), un espace est réservé pour les éditeurs de création de concepts d'ontologies ou la réalisation d'indexation de modèles, de services ou de workflows informatiques ;
- dans la zone (D), un espace est réservé pour représenter les valeurs des propriétés des différentes représentations de description syntaxique.

Dans cette section, nous décrivons en détail les étapes de ce scénario. Nous commençons par expliquer comment la création et la gestion des ontologies de services sont effectuées. Nous montrons ensuite comment peut être réalisée l'indexation des services ou des workflows informatiques de la modélisation géologique. Enfin, nous expliquons comment se fait la recherche sur les services et les workflows indexés.

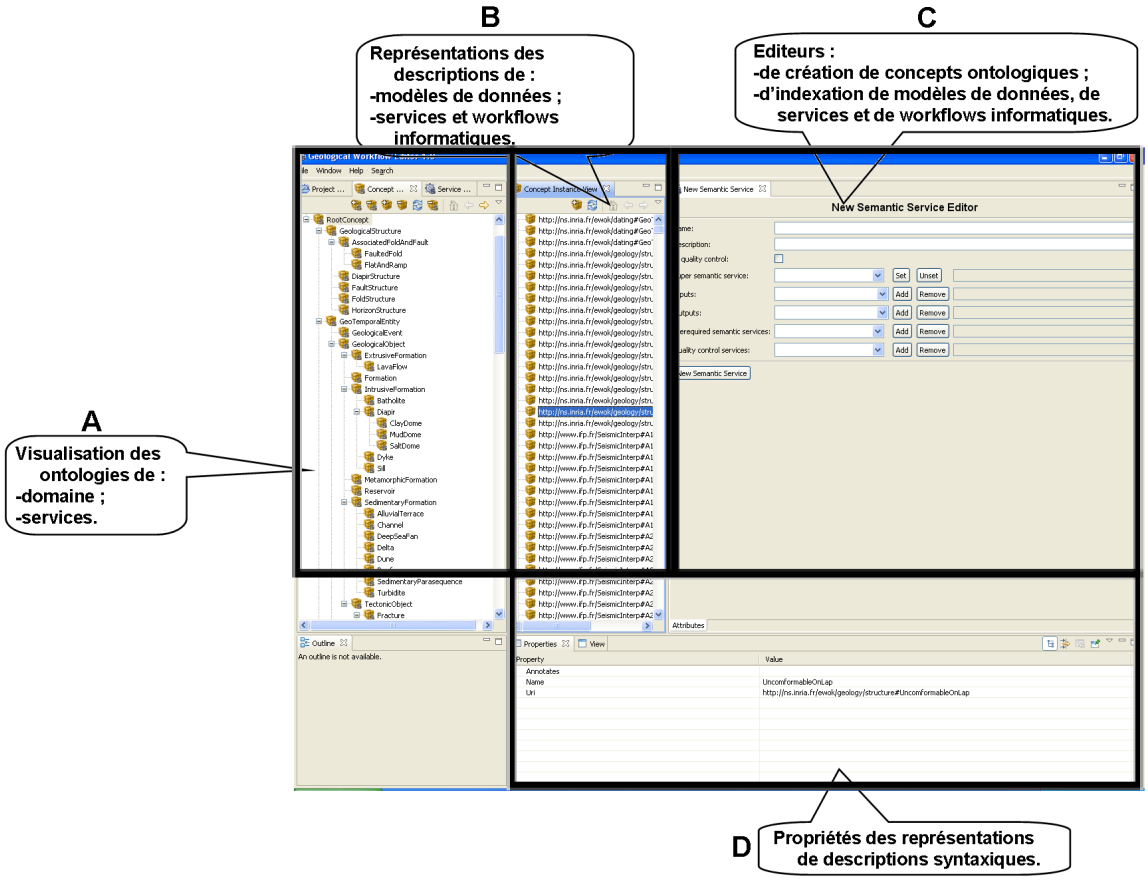


FIG. 8.3 – Vue du GWE pour le scénario de réutilisation de services et de workflows.

3.1 Gestion d'ontologies de modélisation géologique

La caractérisation en hiérarchie des services ainsi que des workflows informatiques en modélisation géologique permet la construction d'ontologies de services dans ce domaine. Un exemple d'une telle ontologie est donnée sur la figure 6.5 du chapitre 6. Dans le GWE, il est possible de créer des ontologies de domaine ainsi que des ontologies de services.

3.1.1 Gestion d'ontologies de domaine

Le GWE est un logiciel dans lequel les modèles de données ainsi que les descriptions ontologiques qui en définissent le sens peuvent être représentés. Ainsi, il est possible de créer des ontologies de domaine. Sur la figure 8.4, nous montrons l'interface du GWE pour la visualisation des ontologies de domaine et leur gestion.

Dans la partie gauche de la figure, les ontologies de domaines sont représentées en arbre. De nombreuses actions peuvent être effectuées sur ces ontologies parmi lesquelles :

– création de nouveaux concepts d'ontologies de domaine ;

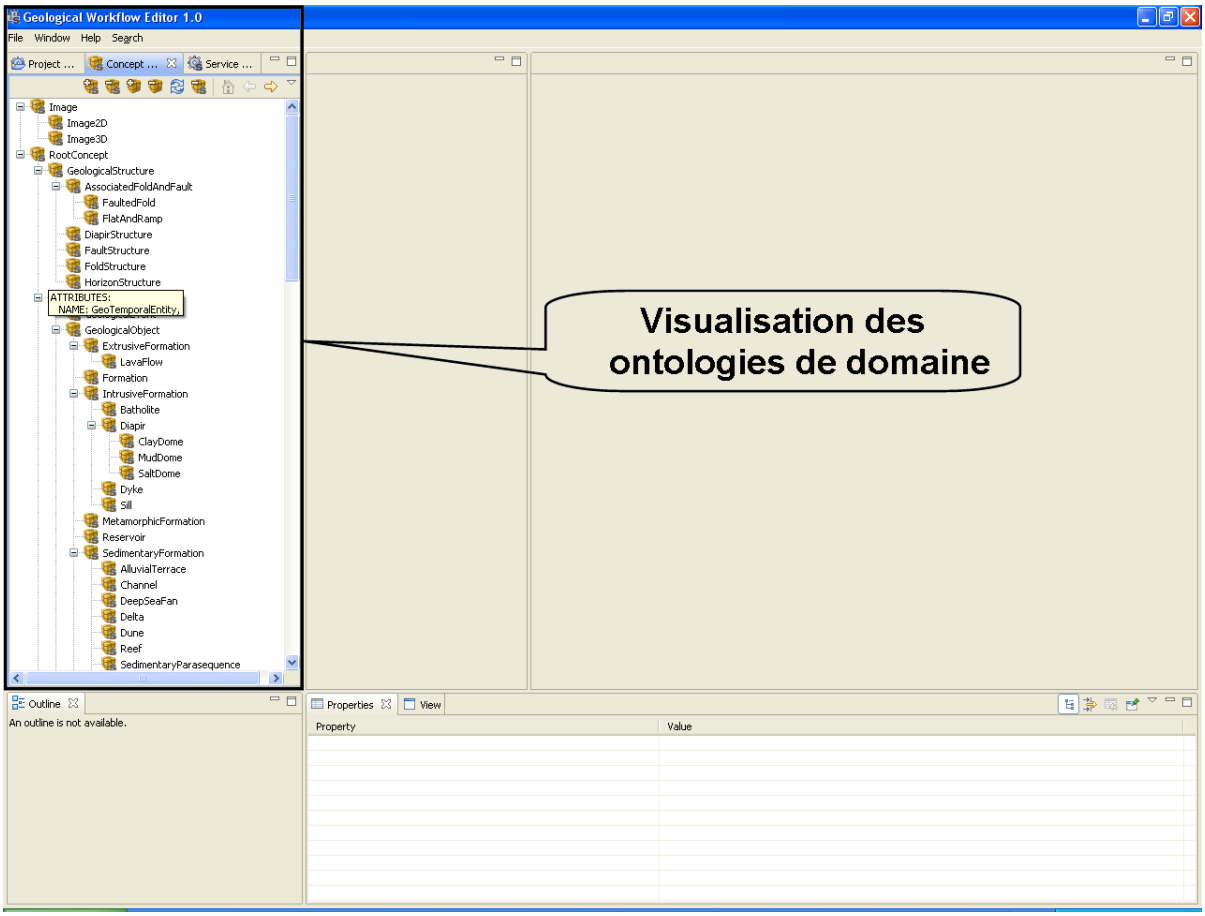


FIG. 8.4 – Vue des ontologies de domaine sur le GWE.

– suppression de concepts existants d'ontologies de domaine ;
– visualisation des attributs des concepts ontologiques.

3.1.2 Gestion d'ontologies de services

Dans le GWE, il est possible d'indexer des services et des workflows informatiques par des services sémantiques. Le GWE permet de créer des ontologies de services composées par ces services sémantiques. Sur la figure 8.5, nous montrons l'interface du GWE qui permet la visualisation et la gestion des ontologies de services.

Dans cette figure, nous montrons qu'il est possible de visualiser la valeur des attributs d'un service sémantique. Par exemple, le service sémantique *Construction* a comme entrée le concept ontologique *SeismicSurface*, etc.

La création d'ontologies de domaine se fait en créant de nouveaux concepts de cette ontologie. De même, la création d'ontologies de services se fait en créant des services sémantiques qui sont les concepts de ces ontologies. Il est possible d'enrichir des ontologies de services,

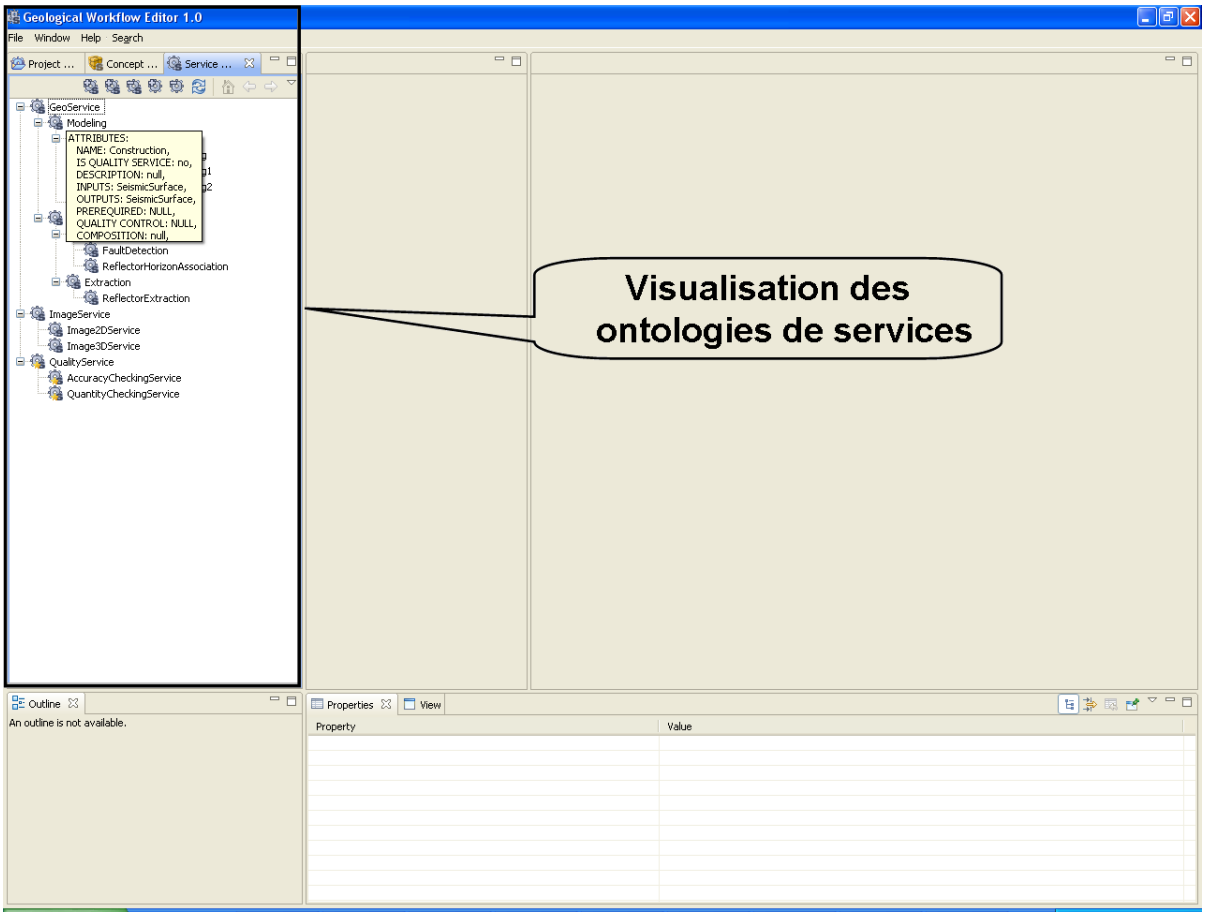


FIG. 8.5 – Vue des ontologies de services sur le GWE.

c'est-à-dire de les faire évoluer en rajoutant de nouveaux services sémantiques. Sur la figure 8.6, nous montrons l'interface du GWE pour la création d'un nouveau service sémantique.

Sur cette figure, il est possible de voir la fenêtre de création de service sémantique. Un certain nombre d'attributs doivent être valués pour pouvoir enrichir l'ontologie des services avec un nouveau service sémantique. Les attributs sont les suivants :

- *name* : le nom du service sémantique. Exemple : "ReflectorExtraction" ;
- *description* : la description du service sémantique. Exemple : "Ce service sémantique a pour but de caractériser et d'indexer les services et workflows informatiques qui réalisent une extraction de réflecteurs" ;
- *isQualityControl* : indique le fait que ce service sémantique est un service sémantique de contrôle de qualité ou un service sémantique classique. Exemple : "Non". Par contre, d'autres services sémantiques peuvent être des services de contrôle de qualité, c'est-à-dire des services sémantiques dont l'objectif est d'indexer les services informatiques qui vérifient la qualité des données produites par d'autres services et workflows informatiques ;
- *superSemanticService* : le service sémantique qui subsume ce service, c'est-à-dire le service sémantique qui remplit une fonction plus générale que ce service sémantique.

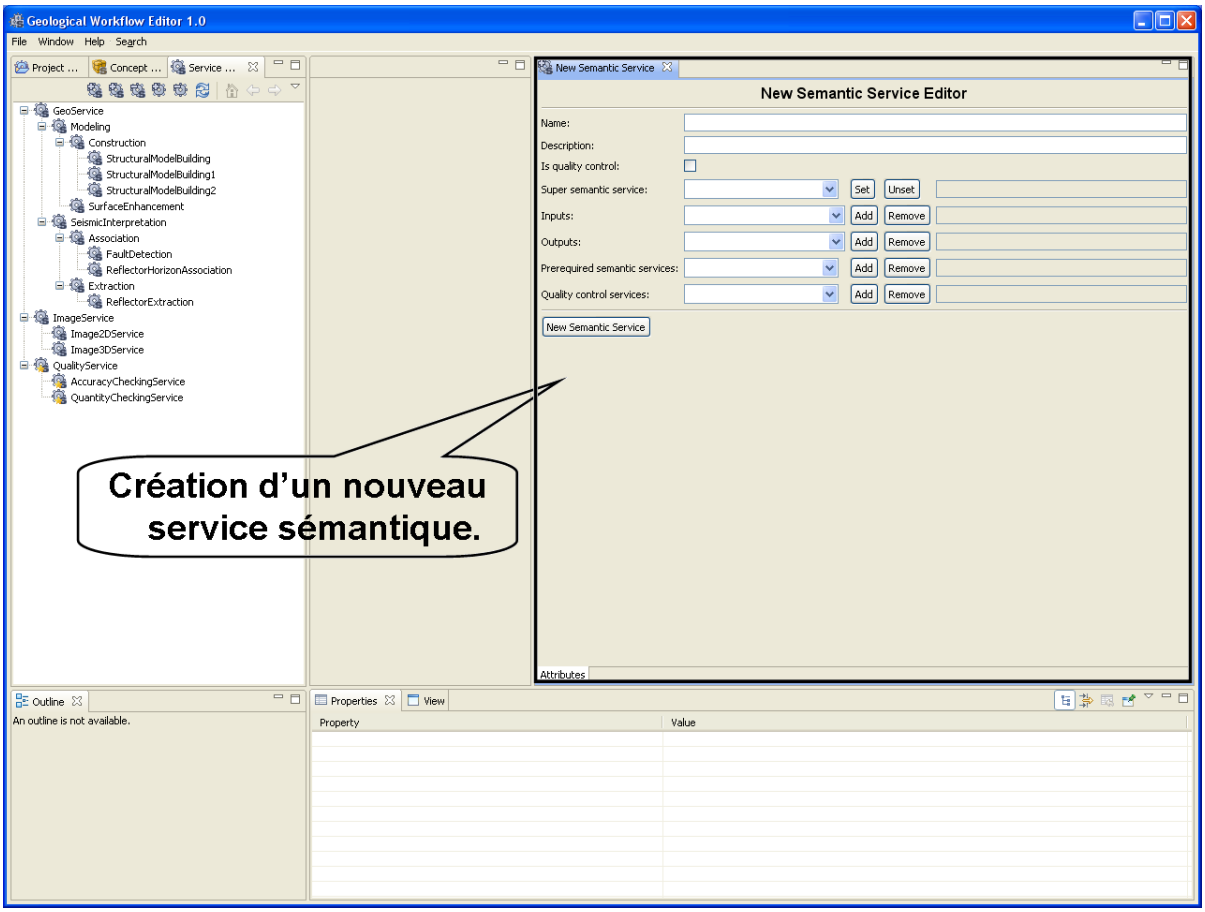


FIG. 8.6 – Vue de l'interface de création de services sémantiques.

Exemple : "Extraction" ;

- *inputs* : les concepts ontologiques que ce service sémantique prend en entrée. Exemple : "SeismicCube" ;
- *outputs* : les concepts ontologiques que ce service sémantique rend en sortie. Exemple : "Reflector" ;
- *prerequiredSemanticServices* : quels sont les services sémantiques qui doivent être effectués avant le service sémantique en question. Exemple : "Null" pour dire qu'aucun service sémantique ne doit obligatoirement s'effectuer préalablement ;
- *qualityControlServices* : les services sémantiques de contrôle de qualité qui doivent être effectués juste avant le service sémantique en question. Exemple : "Null" pour dire qu'aucun service sémantique de contrôle de qualité ne doit obligatoirement s'effectuer directement avant.

Enfin, le GWE permet également de supprimer des services sémantiques et de les modifier pour faire évoluer les ontologies de services.

3.2 Indexation de services et de workflows informatiques

Une fois les concepts des ontologies de domaine et de services créés, il est possible d'indexer les services et les workflows informatiques. Cette indexation se fait en instanciant les concepts des ontologies de services, c'est-à-dire des services sémantiques. Dans la figure 8.7, nous montrons comment il est possible d'indexer dans le GWE un service ou un workflow informatique de la modélisation géologique en le rendant instance d'un service sémantique.

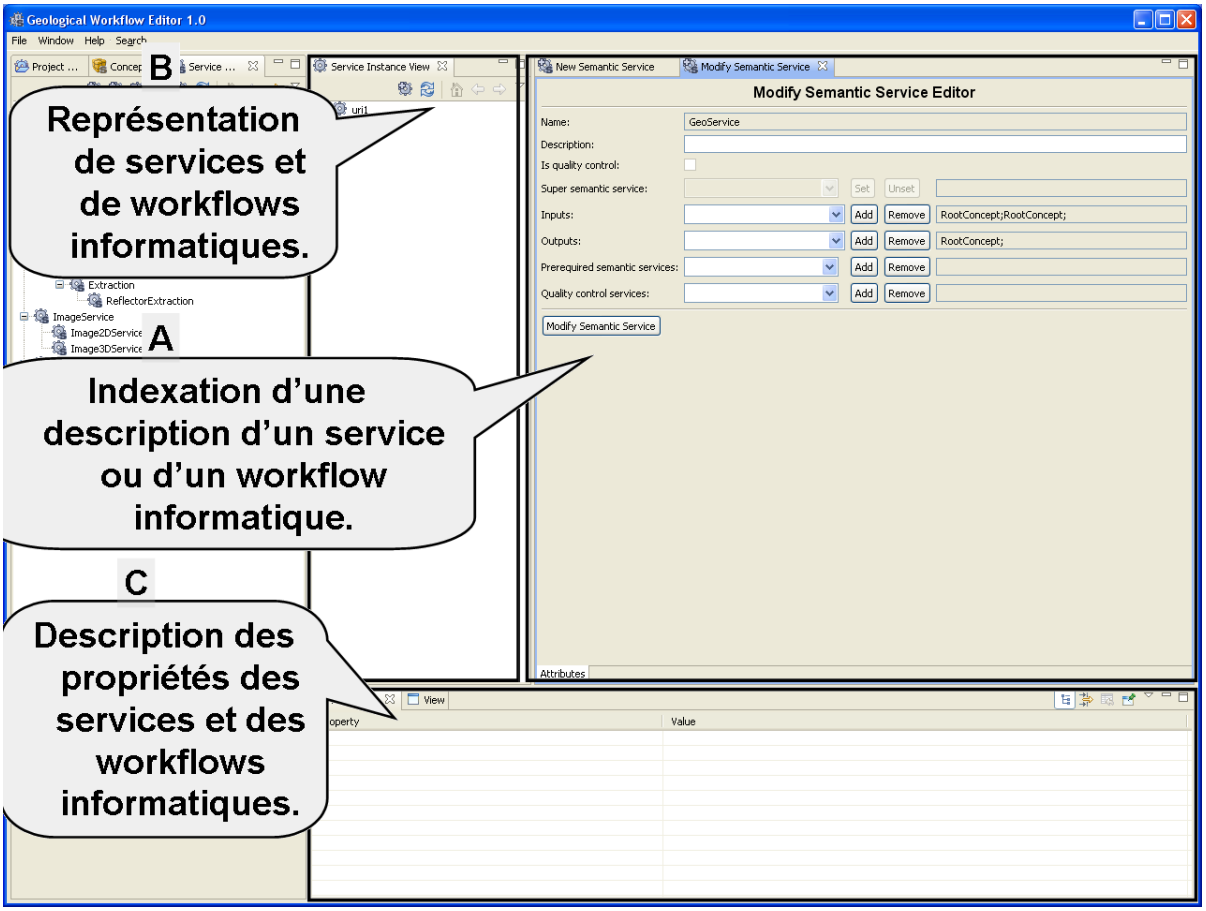


FIG. 8.7 – Vue de l'interface d'indexation de services et de workflows informatiques.

Sur la partie (A) de la figure, il est possible de voir un éditeur d'instanciation de services sémantiques. Cet éditeur peut s'ouvrir si un service sémantique est sélectionné. Les propriétés qui définissent le service ou le workflow informatique à indexer doivent être valuées. Ces propriétés sont les suivantes :

- *URI* : l'URI du service ou du workflow informatique à indexer sémantiquement. Exemple : "uri4" ;
- *name* : le nom du service ou du workflow informatique à indexer. Exemple : "getRef" ;
- *URL* : l'URL, c'est-à-dire le chemin physique où se trouve la description du service ou du workflow informatique à indexer. Exemple : "serviceinfos/getRef.wsdl" ;

- *descriptionLanguage* : le langage utilisé pour la description du service Web ou du workflow informatique à indexer, et utilisé pour décrire le service ou le workflow. Exemple : "WSDL" ;
- *inputs* : les modèles de données qui sont en entrée dans la description du service ou du workflow informatique à indexer. Exemple : "SEG-Y" ;
- *outputs* : les modèles de données qui sont en sortie dans la description du service ou du workflow informatique à indexer. Exemple : "XYZFile" ;
- *isComplex* : le fait que l'indexation concerne un service Web ou bien un workflow informatique. Exemple : "No" car dans ce cas, il s'agit d'un service et non pas d'un workflow informatique ;
- *composition* : n'est valuée que dans le cas où c'est un workflow informatique et non pas un service informatique qui est indexé. Exemple : "Null" car pas de composition.

Sur les parties (B) et (C), nous avons respectivement d'une part la représentation des descriptions des services et des workflows informatiques, et d'autre part la représentation de leurs propriétés.

3.3 Exploitation des services et des workflows représentés dans le GWE

Il est possible d'effectuer une recherche sur tous les services et workflows informatiques en filtrant sur les valeurs de leurs propriétés. Par exemple, il est possible de rechercher les services informatiques qui produisent des modèles de données particuliers.

Ce que nous rendons possible à travers le GWE est la recherche de services et de workflows informatiques de manière sémantique. Ainsi, il est possible d'effectuer une recherche sur les ontologies de services en navigant dans les arborescences de celles-ci. Par exemple, on peut rechercher un service d'interprétation sismique et par spécialisation, un service d'interprétation sismique qui extrait des éléments, etc. Et cela jusqu'à arriver au service sémantique cible, par exemple, un service sémantique d'extraction d'horizons. En sélectionnant le service sémantique cible, les services et les workflows informatiques indexés par ce service sémantique sont identifiés.

Chacun de ces services ou workflows informatiques ont des propriétés fonctionnelles telles que les modèles de données en entrée, en sortie, la localisation (URL) et des propriétés non-fonctionnelles liées à la qualité de service (disponibilité, performance, etc.). Il est ainsi possible de choisir parmi les différents services et workflows informatiques en fonction de ces propriétés.

3.4 Exécution de workflows informatiques

L'exécution n'est pas encore intégrée dans le GWE mais fait partie des perspectives. L'objectif est de permettre l'exécution des workflows informatiques obtenus. Une fois ces workflows informatiques indexés, il est possible de les sélectionner en vue de les exécuter. Dans ce cas,

il faut choisir une ou plusieurs instances de concepts ontologiques correspondant à des objets géologiques. Ces instances annotent des instances de modèles de données. Elles doivent être conformes aux modèles de données en entrée du workflow informatique sélectionné. Ainsi, il est possible d'exécuter le workflow informatique dans un moteur de workflow.

L'une des solutions à adopter consiste à connecter le GWE à plusieurs moteurs de workflows. Lors de la sélection d'un service ou d'un workflow informatique, il sera exécuté sur le bon moteur.

Ayant présenté dans cette section le scénario de réutilisation de services et de workflows informatiques, nous nous intéressons maintenant à un deuxième scénario du GWE basé sur le premier. Ainsi, l'indexation des services et workflows par des caractérisations ontologiques permet de créer de nouveaux workflows en se basant sur le concept de workflow sémantique. C'est le scénario que nous décrivons dans la prochaine section.

4 Création de nouveaux workflows

La création de nouveaux workflows informatiques nécessite une connaissance technique des services et des sous-workflows à composer d'une part, et une connaissance explicite des fonctions précises qu'ils remplissent d'autre part. Grâce à l'indexation sémantique des services et workflows informatiques de la modélisation géologique, il est possible de faire abstraction de telles informations lors de la création de nouveaux workflows informatiques. Concrètement, il s'agit de créer d'abord des workflows sémantiques en composant des services sémantiques. Cette création est effectuée par des experts en gestion de workflows sémantiques. Il appartient ensuite à ces experts et aux administrateurs techniques d'associer à chaque service sémantique qui compose ces workflows sémantiques, des services ou des workflows informatiques. Cette association se fait en choisissant pour chaque service sémantique qui compose le workflow sémantique, un seul service ou workflow informatique. Des workflows informatiques sont ainsi obtenus et associés au workflow sémantique correspondant et qui représentent eux-même des services sémantiques. La figure 8.8 résume les étapes sur le GWE de création des nouveaux workflows informatiques à partir du travail effectué sur l'indexation sémantique des services et des workflows informatiques.

Dans cette figure, nous pouvons également voir le profil qui est affecté à un type de tâche donné. L'étape (1) de la figure 8.8 correspond à la construction d'un workflow sémantique. Cette construction s'effectue en composant des services sémantiques à partir des ontologies de services. Cette tâche est effectuée par un expert en gestion de workflows sémantiques. Le workflow sémantique ainsi construit est soumis à une validation partielle correspondant à l'étape (2). Ce nouveau workflow sémantique permet d'enrichir l'ontologie des services dont les services sémantiques qui ont été composés font partie (étape (3)). L'étape suivante (étape (4)) consiste à sélectionner un service ou un workflow informatique pour chaque service sémantique qui compose le workflow sémantique globale. L'étape (5) permet la construction d'un workflow

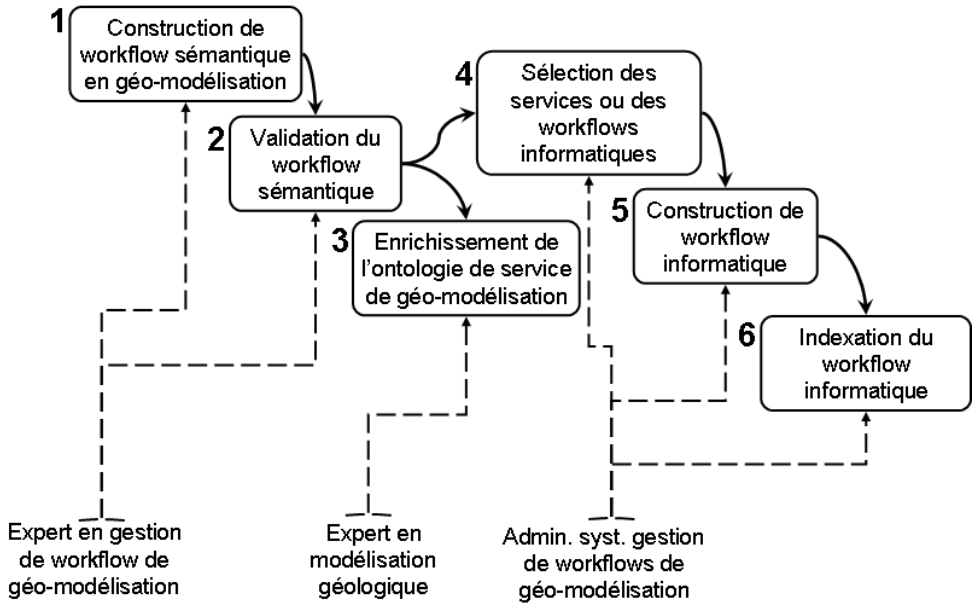


FIG. 8.8 – Création de nouveaux workflows informatiques.

informatique. Enfin, l'étape (6) consiste à indexer le workflow informatique ainsi construit. Par la suite, les utilisateurs géologues, géophysiciens et ingénieurs de réservoirs peuvent rechercher le workflow informatique de la même manière que dans le précédent scénario.

Nous montrons à présent les différentes zones du GWE participant à la réalisation du scénario de création de nouveaux workflows. Nous montrons dans la figure 8.9 une vue des zones d'édition et de visualisation du GWE relatives à ce scénario :

- dans la zone (A) de la figure, un espace est réservé pour la visualisation des ontologies de services ;
- dans la zone (B), un espace est réservé pour visualiser les services et les workflows informatiques indexés par les concepts des ontologies de services ;
- dans la zone (C), un espace est réservé pour un éditeur graphique de construction de workflows sémantiques ;
- dans la zone (D), un espace est réservé pour la visualisation en arbre des workflows sémantiques construits ;
- dans la zone (E), un espace est réservé pour la visualisation des propriétés des workflows sémantiques construits.

Dans cette section, nous décrivons en détail les étapes de ce scénario. Nous commençons par expliquer la construction d'un workflow sémantique sur le GWE et sa validation, puis nous montrons comment un workflow sémantique permet l'enrichissement d'ontologies de services. Nous expliquons ensuite la sélection des services ou des workflows informatiques pour chaque service sémantique composant le workflow sémantique global. Enfin, nous décrivons la construction de workflows informatiques.

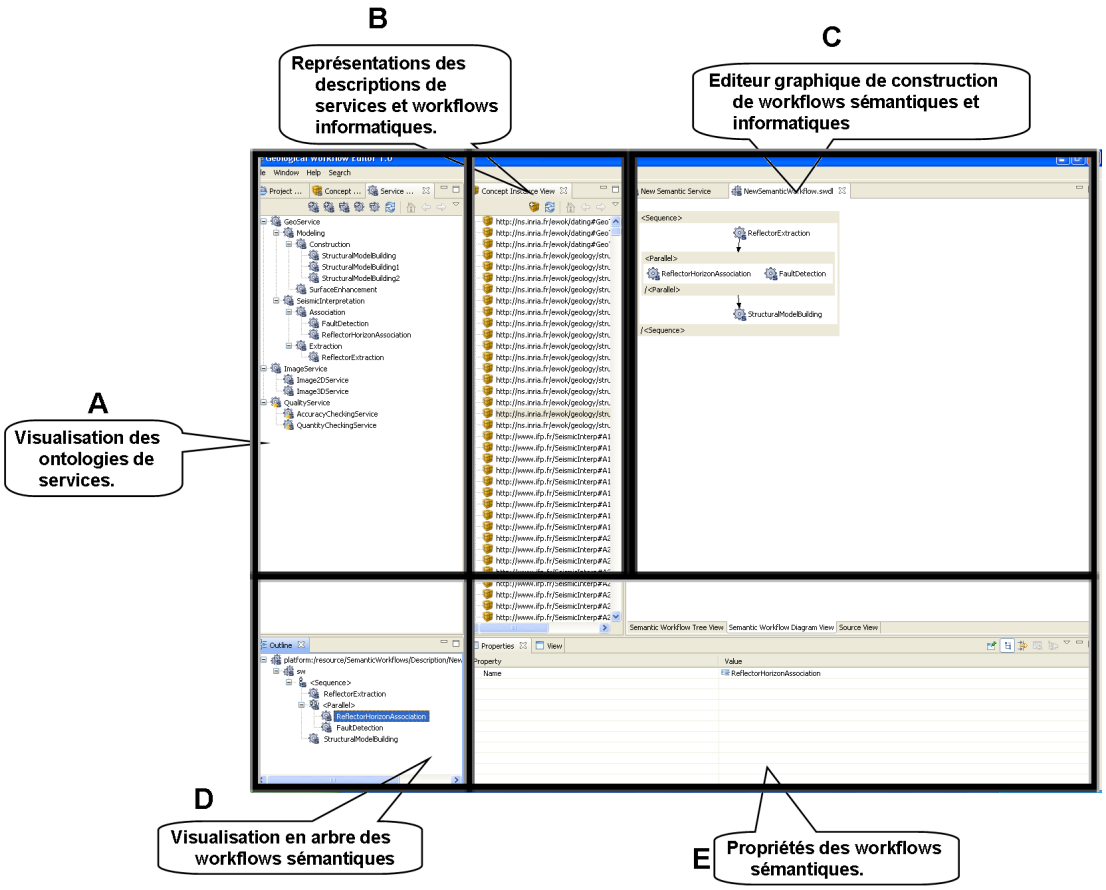


FIG. 8.9 – Vue du GWE pour le scénario de création de nouveaux workflows informatiques.

4.1 Construction de workflows sémantiques

A travers le GWE, il est possible de construire des workflows sémantiques en composant des services sémantiques d'ontologies de services. Dans la figure 8.10, nous montrons l'interface du GWE pour visualiser les workflows sémantiques conçus.

Concrètement, il s'agit de sélectionner des services sémantiques à partir de la vue des ontologies de services (partie (C) de la figure 8.10). Les services sémantiques sélectionnés doivent être composés suivant un modèle de description de composition de services sémantiques décrit dans la section 4.1 du chapitre 4. Le GWE propose plusieurs représentations visuelles du workflow sémantique :

- représentation graphique (figure 8.10 partie (A)) ;
- vue en arborescence (figure 8.10 partie (B)) ;
- représentation en XML (figure 8.11).

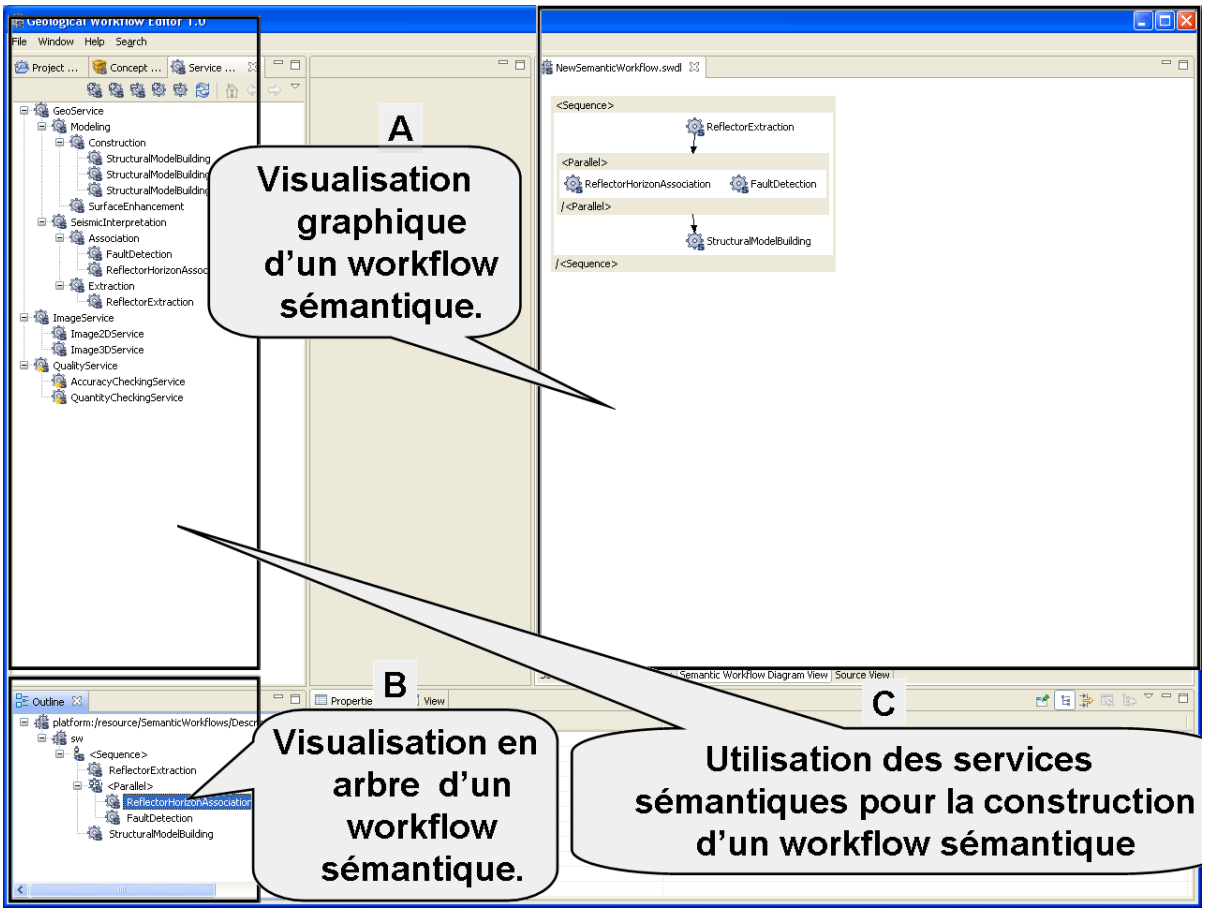


FIG. 8.10 – Vue de l'interface de création de workflows sémantiques.

4.2 Validation de workflows sémantiques

La validation de workflows sémantiques peut être réalisée sur certaines propriétés fonctionnelles au niveau sémantique parmi lesquelles :

- *services sémantiques prérequis*. Il est possible de valider partiellement la précédence requise entre les services sémantiques qui composent un workflow sémantique particulier. Ainsi, si dans le workflow sémantique, un service sémantique particulier SS_i requiert un autre service sémantique SS_j, ce dernier doit se trouver avant lui séquentiellement dans le workflow sémantique. S'il ne se trouve pas avant, le worflow sémantique global aura comme service sémantique prérequis le service sémantique SS_j ;
- *services sémantiques de contrôle de qualité prérequis*. Dans le cas où un service sémantique SS_i nécessite qu'un service sémantique de contrôle de qualité $SSCQ_j$ le précède, et que ce dernier ne le précède pas directement dans le workflow sémantique global, le GWE signale une erreur. En effet, pour le cas de la modélisation géologique, on considère que les services sémantiques de contrôle de qualité doivent précéder directement un service sémantique donné.

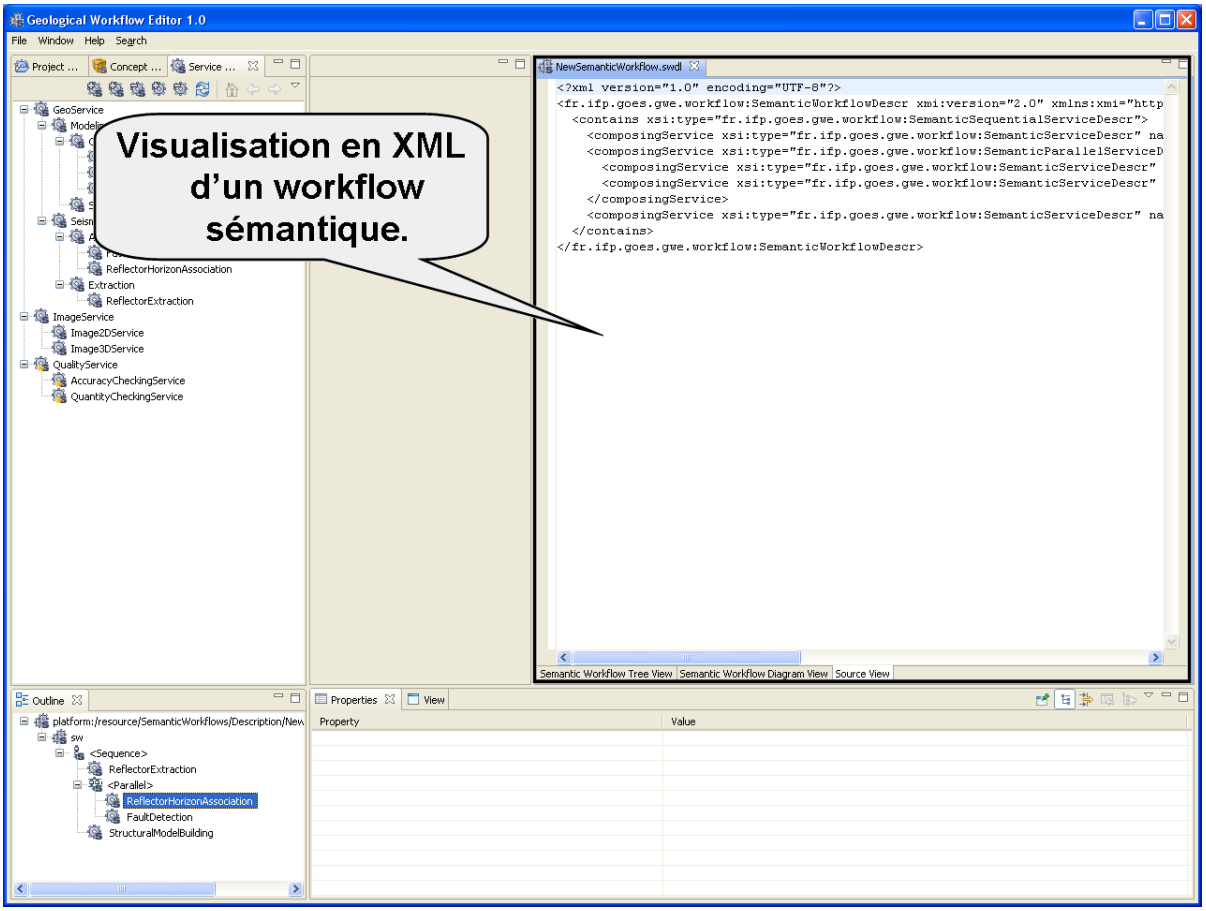


FIG. 8.11 – Visualisation du workflow sémantique en XML.

– *concepts en entrée et en sortie*. En ce qui concerne les concepts en entrée et en sortie d'un service sémantique donné, le GWE signale les services sémantiques dont les concepts en entrée n'ont pas été associés (mappés) à des concepts en sortie. Ce n'est donc qu'un avertissement (warning) vu qu'il n'est pas possible de "décider" si c'est une erreur ou bien une impossibilité d'association, ou encore une entrée du workflow sémantique global. Dans tous les cas, le GWE calcule les concepts en entrée et en sortie du workflow sémantique global.

Nous donnons un exemple du premier type de validation. Nous représentons pour cela dans la figure 8.12 le processus présenté dans le chapitre 3 (partie gauche), similaire à une description de workflow sémantique, et sa vue graphique *en arbre* obtenue dans le GWE (partie droite).

Ainsi, dans la description du service sémantique *HorizonReflectorAssociation*, l'un des services sémantiques prérequis à ce service sémantique est *ReflectorExtraction*. Toutefois, dans le cas hypothétique où le service sémantique *ReflectorExtraction* ne précédait pas (directement ou indirectement) le service *HorizonReflectorAssociation*, le GWE se contente de le signaler par un avertissement et de calculer la liste de services sémantiques prérequis dont ferait partie *ReflectorExtraction*.

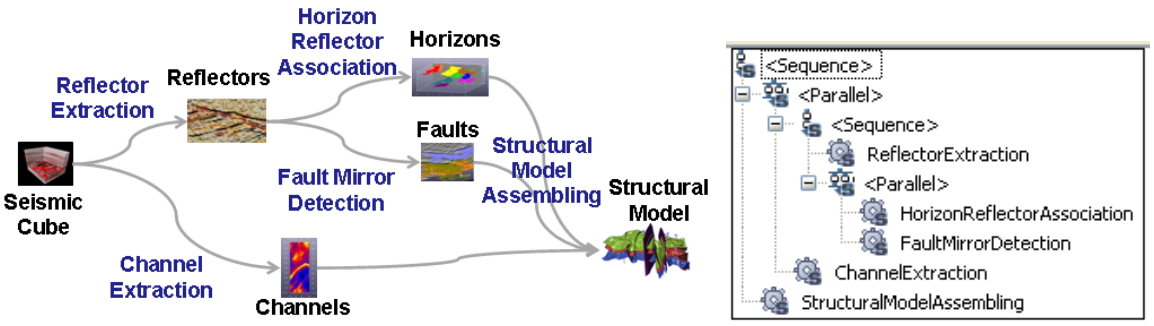


FIG. 8.12 – Workflow sémantique d'interprétation sismique.

4.3 Enrichissement de l'ontologie des services

Une fois que le workflow sémantique a été créé par l'expert en gestion de workflows de modélisation géologique, il représente lui-même un nouveau concept sémantique et peut être considéré à son tour comme un service sémantique. Le GWE permet ainsi d'enrichir une ontologie de services en ajoutant le workflow sémantique construit en tant que service sémantique en respectant la relation de subsomption définie sur les services sémantiques. Cet ajout se fait en calculant les attributs du potentiel nouveau service sémantique à partir des services sémantiques qui composent le workflow sémantique et de leurs attributs. Par exemple, l'entrée sémantique pour le workflow sémantique *StructuralModelBuilding* est le concept *SeismicCube*, et sa sortie sémantique est un *StructuralModel* et il n'a pas de service sémantique pré-requis. L'interface du GWE pour l'ajout d'un service sémantique est montrée dans la figure 8.6.

Dans la section 4.3 du chapitre 6, nous avons montré comment le workflow sémantique *StructuralModelBuilding* représenté sur la figure 6.8 permet d'enrichir l'ontologie de services représentée sur la figure 6.5. Nous montrons sur la figure 8.13 l'éditeur du GWE pour l'ajout d'un service sémantique (partie gauche) et l'ontologie de services ainsi mise à jour correspondant à la figure 6.11 du chapitre 6 (partie droite).

4.4 Association des services et workflows informatiques aux services sémantiques

Lorsque la construction du workflow sémantique est effectuée par l'expert en gestion de workflows, l'administrateur de workflows et de services informatiques prend le relais. Il peut choisir d'associer à chaque service sémantique qui compose le workflow sémantique un service ou bien un workflow informatique indexé par le même service sémantique. Il est possible lorsqu'un service ou un workflow informatique est associé à un service sémantique qu'il ne soit pas possible de le prendre en compte à cause d'une indisponibilité relevant de sa qualité de service. Dans ce cas, il est possible de le substituer par un autre service ou workflow informatique dit remplaçant. Cela peut être défini par l'expert en gestion de workflow ou bien par un calcul

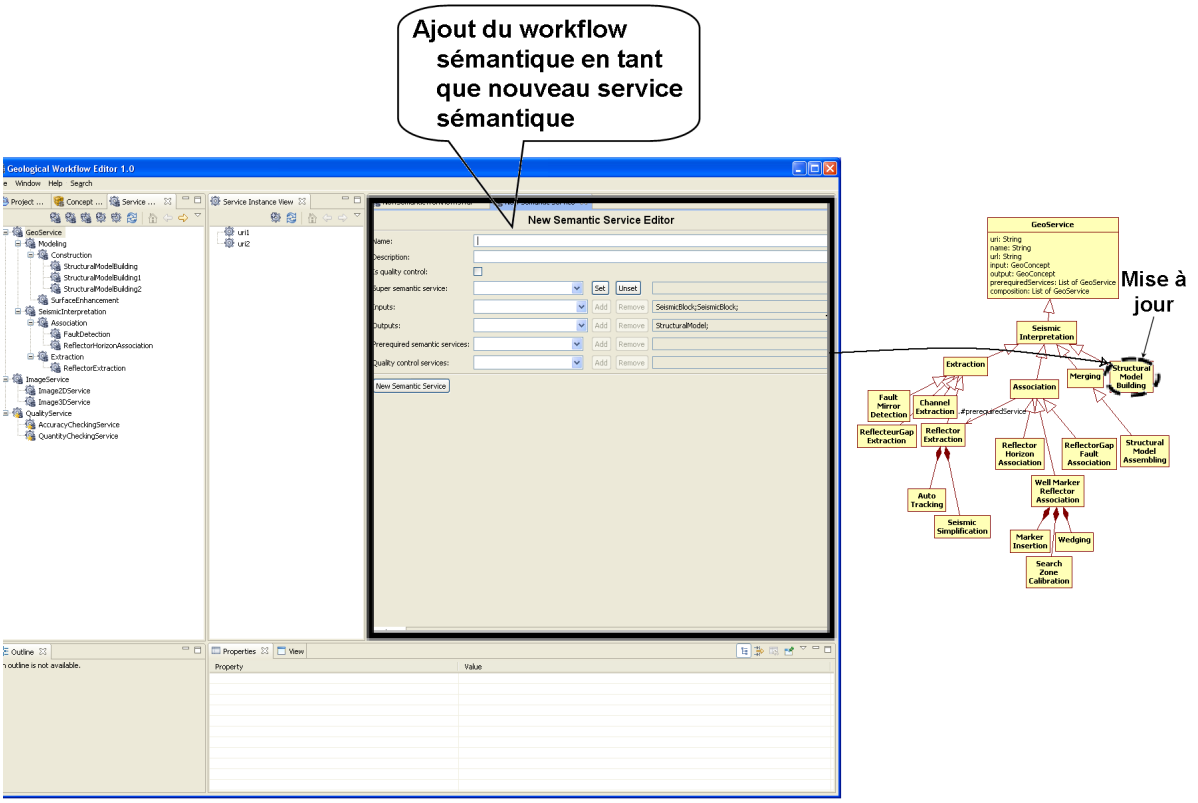


FIG. 8.13 – Enrichissement de l'ontologie des services.

automatique.

Sur la figure 8.14, nous montrons comment il est possible d'associer dans le GWE des services et des workflows informatiques, pour chaque service sémantique qui compose le workflow sémantique construit. Ainsi, après sélection d'un service sémantique (partie (B) de la figure) parmi ceux composant le workflow sémantique global, il faut lui associer un service ou un workflow informatique (partie (A)).

4.5 Construction du workflow informatique

Lorsque tous les services sémantiques qui composent le workflow sémantique sont associés à des services ou workflows informatiques, il est possible de construire un workflow informatique dans un langage de description standard tel que BPEL. Cette partie n'est pas encore supportée dans la version actuelle du GWE, elle est en cours de développement. De plus, il est possible de "générer" sous certaines conditions un workflow informatique dont la formalisation fait partie des perspective. L'un des travaux concernant la génération de workflows

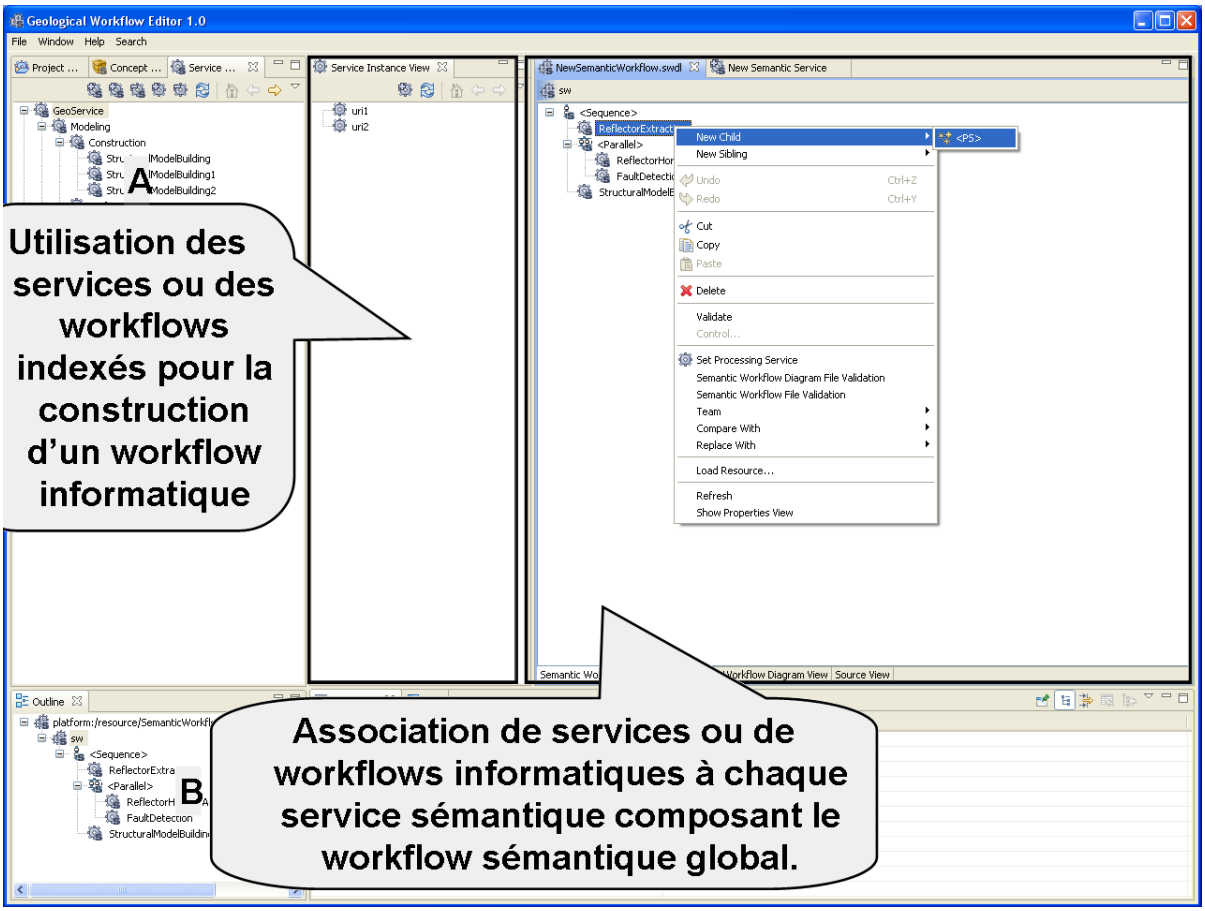


FIG. 8.14 – Association de services informatiques aux services sémantiques dans un workflow sémantique.

informatiques est l'adaptation de modèles de données. En effet, bien que parfois la composition de workflow sémantique soit valide, elle pose des problèmes au niveau informatique. A titre d'exemple, deux services sémantiques peuvent être composés pour donner un workflow sémantique plus complexe (la sortie sémantique du premier service correspond à l'entrée sémantique du deuxième). En revanche, les services ou workflows informatiques associés aux deux services sémantiques peuvent ne pas être composables au niveau informatique car à un même concept ontologique peut correspondre plusieurs modèles de données. Dans la section 4.4 du chapitre 6, ce cas de figure est bien illustré. Dans ce cas, un service informatique dit d'adaptation entre modèles de données est recherché et inséré en cas d'existence dans la séquence pour permettre l'interopérabilité.

4.6 Indexation du workflow informatique obtenu

Une fois que le workflow informatique a été déduit du workflow sémantique conçu, nous disposons d'un nouveau workflow informatique qui peut à son tour être indexé et/ou exécuté. Ce

workflow informatique peut être indexé par le nouveau service sémantique qui a permis d'enrichir l'ontologie de services. Ses propriétés sont calculées à travers les services informatiques qui le composent. Ainsi lors de l'indexation, c'est-à-dire l'insertion du workflow informatique en tant qu'instance d'un service sémantique, les modèles de données en entrées/sorties du workflow informatique, la description du workflow informatique, sa localisation, son créateur, etc. sont les valeurs des propriétés du service sémantique indexant ce workflow informatique.

5 Conclusion

Dans les chapitres précédents, nous avons appliqué notre approche sur la modélisation géologique. Ce chapitre nous a permis de montrer le GWE, prototype développé pour supporter différents scénarii de modélisation géologique. Deux scénarii de gestion sémantique de services et de workflows informatiques en modélisation géologique ont été présentés :

- réutilisation de services et de workflows informatiques par indexation sémantique ;
- création de nouveaux workflows informatiques.

Nous avons développé le GWE pour qu'il réponde à ces besoins en offrant une interface graphique rendant son utilisation facile aux géologues, aux géophysiciens et autres acteurs de la modélisation géologique. Nous avons présenté tout au long de ce chapitre, les différentes fonctionnalités qui ont été développées pour la réalisation de ces objectifs de manière détaillée.

Ainsi, bien que le GWE soit au stade de prototypage, il est déjà utilisable pour représenter des ontologies de domaine et de services. Il permet l'indexation des services et des workflows de modélisation géologique. De plus, il est utilisable pour construire des workflows sémantiques et enrichir les descriptions ontologiques. Nous permettons l'association de services et workflows informatiques aux services sémantiques composant les workflows sémantiques. Comme perspectives, il s'agit de permettre la génération de workflows informatiques dans des descriptions standards tels que BPEL. Il s'agit également de connecter le GWE à un moteur de workflow ou bien d'en construire un suivant les descriptions de workflows informatiques construits.

Le GWE est également un outil générique dans le sens où son application au domaine de la modélisation géologique n'implique pas qu'il soit lié à ce domaine. Il peut être utilisé dans n'importe quel domaine scientifique où des ontologies de services peuvent être conçues. Cette indépendance est permise par la généricité de l'approche que nous avons proposée.

Conclusion générale et perspectives

Conclusion générale

Les services informatiques et les services Web en particulier permettent d'assister les ingénieurs dans la réalisation de tâches telles que la modélisation, le traitement ou le stockage d'informations. Leur composition donne naissance à des workflows réalisant des fonctions plus complexes permettant la systématisation voire l'automatisation de ces tâches. Les fonctions réalisées par ces services et ces workflows informatiques ne sont toutefois pas explicitement décrites ce qui rend leur découverte et leur réutilisation dépendantes de connaissances techniques. Cette absence de sémantique explicite rend les ingénieurs dépendants des développeurs de services et de workflows. Dans le domaine de la modélisation géologique, l'interprétation, l'analyse et le stockage de ces données sont réalisées au travers de logiciels, de services informatiques et de workflows. Ces derniers sont cependant conçus sans méthodologie formelle et leurs descriptions syntaxiques sont dépourvues d'informations sur les concepts représentés par les données manipulées ou les fonctions réalisées.

Pour pallier à ces limitations, de récentes approches d'enrichissement sémantique des descriptions de services Web et de workflows ont été proposées. Elles permettent d'expliciter le sens des fonctions réalisées par les services et les workflows pour faciliter les aspects liés à leur gestion tels que l'interopérabilité, la découverte, l'adaptativité, etc. Ces approches présentent en revanche des insuffisances. Leurs mécanismes d'enrichissement sémantique procèdent soit par annotation sémantique soit par extension des langages de descriptions syntaxiques. Elles ne séparent pas la partie syntaxique utile à l'exécution de la partie sémantique utile aux aspects de gestion. Ainsi, les services et les workflows surchargés ne peuvent plus être utilisés sans que de nouveaux outils soient développés. De plus, aucune classification hiérarchique ou relationnelle selon la sémantique des services et workflows informatiques n'est fournie rendant la recherche ou l'adaptativité restreinte.

Nos contributions portent sur trois points :

1. nous avons présenté une approche ayant pour objectif une meilleure gestion des services

et des workflows. Elle traite les insuffisances mises en évidence ;

2. nous avons outillé cette approche avec le GWE (Geological Workflow Editor) ;
3. nous l'avons mise en œuvre dans le domaine de la modélisation géologique.

Ainsi, les différentes contributions de nos travaux peuvent être détaillées comme suit.

Gestion de services et de workflows

Indexation sémantique au travers de caractérisations ontologiques

Nous avons proposé une approche pour permettre la gestion sémantique de services et de workflows. Notre approche permet l'explicitation de la sémantique des services et des workflows informatiques dans un domaine cible indépendamment des services et des workflows informatiques eux-mêmes. Cette explicitation se fait à travers la création d'ontologies de services permettent de caractériser hiérarchiquement les concepts sémantiques de services et de workflows. La hiérarchie de services utilise une relation de subsomption basée sur la covariance. Notre approche permet également l'indexation ou l'annotation sémantique des workflows et des services informatiques existants par leurs descriptions sémantiques, c'est-à-dire par les concepts des ontologies de services créées, sans modification de leur description syntaxique. Nous avons également procédé à l'indexation sémantique des modèles de données auxquels les services et les workflows informatiques font référence dans leurs descriptions de manière analogue. Ce procédé permet de lier de manière formelle les modèles aux concepts ontologiques qui en décrivent le sens.

En indexant les services et les workflows informatiques déjà conçus et utilisés, leur découverte et leur réutilisation en sont ainsi facilitées. Cette approche facilite également la construction de nouveaux workflows à partir de workflows et de services informatiques préalablement indexés en construisant préalablement des workflows sémantiques.

Ontologies pour la représentation de services sémantiques

Nous avons enrichi les modèles *classiques* d'ontologies pour permettre la représentation d'ontologies de services et particulièrement des services sémantiques qui les composent. En effet, les services sémantiques sont des concepts ontologiques nécessitant une modélisation particulière : ils ont des entrées, des sorties, des services sémantiques pré-requis et d'autre attributs qui référencent des concepts d'ontologies de domaine et d'autres services sémantiques. Une relation de subsomption fondée sur la covariance a été mise en place.

Nous avons pu ainsi appliquer cette approche sur la BDBO de type 3 OntoDB qui permet l'extension de son méta-schéma, c'est-à-dire de son modèle d'ontologie. Ceci nous a permis de représenter au sein de la même BDBO les ontologies de domaine et de services dont les concepts nous ont servi à indexer les services, les workflows et les données qu'ils manipulent.

Extension du langage de requête de BDBO OntoQL

OntoQL permet d'effectuer des requêtes sur la BDBO OntoDB. Nous avons proposé une extension du langage OntoQL pour supporter le modèle d'ontologie de services proposé avec des concepts aux attributs de types complexes. Nous avons ensuite implémenté cette extension du langage OntoQL. En particulier, nous avons implémenté des concepts liés à la gestion de références entre concepts ontologiques et à la gestion de listes de références ontologiques.

Conception et implémentation d'un prototype offrant une meilleure gestion de services et de workflows

Nous avons développé un prototype supportant la gestion de workflows basé sur l'approche que nous avons proposée. Le GWE est générique et nécessite peu d'efforts pour être adapté à un domaine quelconque de l'ingénierie où des ontologies de services et de domaine peuvent être conçues. En effet, le GWE est implémenté suivant le modèle d'ontologie que nous avons étendu pour permettre la représentation de services sémantiques. Ainsi, il permet la création d'ontologies de domaine et de services en instanciant le modèle d'ontologies. Ensuite, il est possible d'indexer les modèles de données, les services et les workflows informatiques. Le GWE stocke les services et les workflows informatiques, leur caractérisation ontologique et les indexations sémantiques dans une BDBO OntoDB. Enfin, il est possible d'exploiter ces indexations sémantiques pour faciliter la découverte, l'interopérabilité ou l'adaptativité de ces services et de ces workflows informatiques.

Modélisation géologique

Ontologie de services en modélisation géologique

Nous avons contribué à la création d'une ontologie de services qui caractérise les fonctions d'analyse, de calcul, d'interprétation, etc. dans le domaine de la modélisation géologique. Nous avons ainsi mis en évidence les différents logiciels et programmes informatiques utilisés dans l'étape la plus en amont de la modélisation géologique, à savoir l'interprétation sismique. Ceci nous a permis d'extraire les informations sémantiques de ces logiciels, de ces programmes et de leurs compositions. Nous avons ainsi pu créer des services sémantiques, donner leurs caractéristiques et les mettre en relations les uns par rapport aux autres ce qui a donné naissance à l'ontologie des services en interprétation sismique. C'est cette ontologie de services qui sert à indexer les services et les workflows informatiques au travers de services sémantiques.

Implémentation d'un prototype pour la gestion de services et de workflows en modélisation géologique

Nous avons appliqué et adapté le prototype que nous avons développé, c'est-à-dire le GWE, pour qu'il réponde à des besoins spécifiques du domaine de la modélisation géologique. Le GWE permet ainsi aux géologues d'effectuer des recherches sémantiques sur les services et les workflows de la modélisation géologique. Il permet entre autre de remplacer un service ou workflow par un autre qui remplit la même fonction ou une fonction spécialisée. Il permet aussi de créer de nouveaux workflows sémantiques en modélisation géologique en composant sémantiquement les services et les workflows qui sont définis dans ce domaine.

Perspectives

Les travaux présentés dans ce mémoire laissent envisager de nombreuses perspectives. Dans cette section, nous présentons succinctement celles qui nous paraissent être les plus intéressantes. Elles concernent la modélisation informatique des aspects liés à (1) la gestion sémantique des services et des workflows informatiques d'une part, et à (2) l'application à la modélisation géologique d'autre part.

Gestion de services et de workflows

Modélisation des aspects non-fonctionnels liés aux services sémantiques

Dans notre approche, nous utilisons le concept de service sémantique qui nécessite l'extension du modèle d'ontologie. Nous avons ainsi permis la représentation des attributs sémantiques fonctionnels tels que les entrées, les sorties, les services sémantiques prérequis, etc. Nous avons en particulier permis la représentation de l'attribut de contrôle de qualité requis pour les services de la modélisation géologique. Une perspective de ce travail est de prendre en compte les autres aspects non-fonctionnels liés à la qualité de services tant au niveau sémantique qu'au niveau informatique. Ces aspects concernent la disponibilité, les préférences, la sécurité, etc. Leur modélisation permettrait de supporter des outils d'évaluation en vue de guider la sélection de services dans un contexte d'utilisation particulier. Par exemple, lors de la sélection d'un service ou d'un workflow informatique pour chaque service sémantique composant un workflow sémantique global, les valeurs des propriétés non-fonctionnelles peuvent donner des indications guidant vers le choix d'un service par un utilisateur ou bien par un algorithme.

Composition et validation semi-automatique de workflows sémantiques

Dans nos travaux, nous avons supposé que la composition de workflows sémantiques se fait interactivement par l'utilisateur. Dans notre cas, des acteurs de la modélisation géologique composent des services sémantiques pour créer des workflows sémantiques. Une perspective concerne la proposition voire l'automatisation de cette composition. En effet, les services sémantiques ont des caractérisations fonctionnelles et sémantiques. Il est possible que des compositions soient proposées par raisonnement et ainsi être réalisées semi-automatiquement avec l'assistance d'experts de domaine et/ou de raisonneurs. Des objectifs peuvent être stipulés tels que l'obtention d'un concept à partir d'un autre, par exemple un *StructuralModel* à partir d'un *SeismicCube*. Dans ce cas, le raisonneur tente de composer ces services sémantiques de telle manière qu'il construise une composition ayant comme entrée un *StructuralModel* et comme sortie un *SeismicCube* et obtiendrait un ou plusieurs workflows sémantiques.

D'autre part, dans nos travaux, nous ne nous sommes pas intéressés en détail aux aspects liés à la validation des workflows tant au niveau sémantique qu'au niveau informatique. Une autre perspective serait de permettre la validation des workflows sémantiques créés en prenant en compte leurs caractéristiques fonctionnelles et non-fonctionnelles. Des règles de validation doivent néanmoins être établies et intégrées au modèle de description de composition. Par exemple, une règle stipulant que chacune des entrées sémantiques d'un service sont produites en sortie par au moins un des services sémantiques qui le précèdent.

Génération semi-automatique voire automatique de workflows informatiques

Nous avons proposé le concept de worfklow sémantique qui est la composition de services sémantiques. L'objectif au travers des workflows sémantiques est de permettre la construction voire la génération de workflows informatiques en sélectionnant un service ou un workflow informatique pour chaque service sémantique qui compose un workflow sémantique. La génération est une perspective qui nécessite une étude approfondie des critères la rendant possible. En effet, des indéterminismes sont à lever pour permettre la génération de workflows informatiques ou bien des conventions peuvent être posées. Une étude théorique sur les conditions de génération de tels workflows informatiques est une des perspectives de nos travaux. La génération peut être entièrement automatisée pour le cas trivial de la séquence et en supposant que chaque sortie d'un service ou d'un workflow informatique alimente l'entrée du service ou du workflow informatique le succédant dans l'enchaînement. Elle ne peut en revanche qu'être semi-automatique dans le cas où il y a une impossibilité de déterminer si une entrée d'un service informatique est alimentée par un sortie de l'un des services informatiques qui le précèdent ou bien si c'est une entrée absolue du workflow informatique global à construire. Pour remédier à cela, la définition d'un langage de composition de workflows sémantiques au sens du modèle d'ontologie développé peut être envisagée.

Substituabilité des services sémantiques

Dans notre approche, nous nous sommes fixés comme objectif principal la caractérisation sémantique des services et des workflows d'un domaine en construisant des ontologies de services. Pour parvenir à cet objectif, nous nous sommes principalement basés sur la covariance pour établir la relation de subsomption dans les ontologies de services que nous avons définies. Toutefois, d'autres relations, issues de la théorie des types notamment, telles que la contravariance, peuvent être considérées et fait partie de nos perspectives. Un des avantages à considérer d'autres types de relations est d'assurer la substituabilité des services sémantiques dans les compositions au niveau sémantique, c'est-à-dire dans les workflows sémantiques. Ainsi, le choix par exemple qui s'appuie sur la relation de contravariance pour les entrées et de covariance pour les sorties assurerait une telle substituabilité.

Modélisation géologique

Évaluation du GWE

Nous avons développé le GWE sur la base de notre approche appliquée au domaine de la modélisation géologique. Les ontologies créées sont construites par un nombre restreint d'acteurs de la modélisation géologique. De plus, le GWE n'a été utilisé que par quelques acteurs de la modélisation géologique et doit être donc soumis à une évaluation plus poussée. Cette évaluation concerne principalement les aspects techniques tels que les performances et le passage à l'échelle liés au GWE et à la BDBO OntoDB. Nous pensons que les structures utilisées pour les développements permettent la montée en charge vu que nous stockons dans OntoDB uniquement des indexations sémantiques sur des liens vers les descriptions syntaxiques. Enfin, l'évaluation concerne également l'utilisabilité de l'approche par les acteurs de la modélisation géologique. Ainsi, ces derniers ayant une mode de travail bien établi, une certaine adaptation du GWE doit être effectuée principalement sur l'IHM de ce dernier.

Extension du GWE

Le GWE est un prototype logiciel de gestion de workflows. Il permet déjà d'utiliser, après une recherche sémantique, les services et les workflows de la modélisation géologique. Toutefois, de nombreux objectifs ont été découverts au fur et à mesure de son utilisation. Cela laisse entrevoir de nombreuses évolutions parmi lesquelles la génération de workflows informatiques et de leurs exécutions. En effet, le GWE doit ainsi permettre la connexion à un ou plusieurs moteurs de workflows afin de permettre l'exécution des services et des workflows informatiques indexés. D'autre part, certaines technologies voire le développement d'un outil propriétaire pourrait permettre l'exécution de descriptions hétérogènes de services et de workflows. Enfin, le GWE doit permettre la sélection de données à partir d'OntoDB et leur utilisation en tant

qu’instances en entrée des descriptions informatiques des services et des workflows informatiques à exécuter. La sélection des données doit pouvoir se faire soit directement, soit à travers leurs descriptions sémantiques.

Bibliographie

[Abadi et al., 2007] Abadi, D. J., Marcus, A., Madden, S. R., et Hollenbach., K. (2007). Scalable semantic web data management using vertical partitioning. In *In Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB'07), pages 411-422*.

[Ait-Ameur, 2009] Ait-Ameur, Y. (2009). A semantic repository for adaptative services. Int. conf. SHWS'09.

[Ait-Sadoune et Ait-Ameur, 2009] Ait-Sadoune, I. et Ait-Ameur, Y. (2009). A proof based approach for modelling and verifying web services compositions. In Society, I. C., editor, *Proceedings of the 2009 14th IEEE International Conference on Engineering of Complex Computer Systems*, pages 1–10, Potsdam Germany.

[Ait-Sadoune et Ait-Ameur, 2010] Ait-Sadoune, I. et Ait-Ameur, Y. (2010). Stepwise design of bpel web services compositions, an event b refinement based approach. In Heidelberg, S. B. ., editor, *Software Engineering Research, Management and Applications (SERA 2010)*, volume 296/2010 of *Studies in Computational Intelligence*, pages 51–68, Montreal, Canada.

[Alexaki et al., 2001a] Alexaki, S., Christophides, V., Karvounarakis, G., Plexousakis, D., et Tolle, K. (2001a). On storing voluminous rdf descriptions: The case of web portal catalogs. In Proc. ofWebDB'01 (co-located with ACM SIGMOD'01).

[Alexaki et al., 2001b] Alexaki, S., Christophides, V., Karvounarakis, G., Plexousakis, D., et Tolle, K. (2001b). Rdfsuite: Managing voluminous rdf description bases. 2nd Int. Workshop on the Semantic Web.

[Beicip-Franlab, 2000] Beicip-Franlab (2000). Simgrid, reservoir simulator grid building. Technical report, The Reservoir Modeling Line.

[Beicip-Franlab, 2010] Beicip-Franlab (2010). *OpenFlowSuite 2010*.

[Belaid et al., 2010] Belaid, N., Ait-Ameur, Y., Jean, S., et Rainaud, J.-F. (2010). Toward a semantic management of geological modeling workflows. In *International Conference on Knowledge Engineering and Ontology Development (KEOD'10)*.

[Belaid et al., 2009a] Belaid, N., Ait-Ameur, Y., et Rainaud, J.-F. (2009a). A semantic handling of geological modelling workflows. MEDES '09: International ACM Conference on

Management of Emergent Digital EcoSystems, Lyon, France, October 27-30, 2009, 978-1-60558-829-2, ACM, 2009, pp. 83-90.

[Belaid et al., 2009b] Belaid, N., Ait-Ameur, Y., et Rainaud, J.-F. (2009b). A semantic repository for geological modeling workflows,. ICWS'09, IEEE International Conference on Web Services, ICWS 2009, Los Angeles, CA, USA, 6-10 July 2009, IEEE, 2009, pp. 1030-1031.

[Belaid et al., 2009c] Belaid, N., Ait-Ameur, Y., Rainaud, J.-F., et Jean, S. (2009c). Swapt: Semantic workflow architecture for petroleum techniques. WEBIST 2009 - Proceedings of the Fifth International Conference on Web Information Systems and Technologies, Lisbon, Portugal, March 23-26, 2009, 978-989-8111-81-4, INSTICC Press, 2009, pp. 101-104.

[Beneventano et al., 2000] Beneventano, D., Bergamaschi, S., Castano, S., Corni, A., Guidetti, R., Malvezzi, G., Melchiori, M., et Vincini, M. (2000). Information Integration: The MOMIS Project Demonstration. In *Proceedings of 26th International Conference on Very Large Data Bases (VLDB'00)*, pages 611–614. Morgan Kaufmann.

[Benslimane et al., 2007] Benslimane, D., Maamar, Z., Taher, Y., Lahkim, M., Fauvet, M.-C., et Mrissa, M. (2007). Multi-layer and multi-perspective approach for web services compositio. In *Proc. of the 21st International Conference on Advanced Information Networking and Applications (AINA-07), Niagara Falls, Canada, IEEE*.

[Berners-Lee et al., 2001] Berners-Lee, T., Handler, J., et Lassila, O. (2001). The semantic web. *Scientific American*.

[Borst, 1997] Borst, W. N. (1997). *Construction of Engineering Ontologies*. PhD thesis, University of Twente, Enschede.

[Bozsak et al., 2002] Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Stumme, G., Sure, Y., Tane, J., Volz, R., et Zacharias, V. (2002). Kaon - towards a large scale semantic web. In *Proceedings of the 3rd International Conference on E-Commerce and Web Technologies (EC-WEB'02)*, pages 304–313, London, UK. Springer-Verlag.

[Broekstra et al., 2002a] Broekstra, J., Kampman, A., et Harmelen, F. V. (2002a). Sesame: A generic architecture for storing and querying rdf and rdf schema. In Proc. of the First Inter. Semantic Web Conf., pages 54-68.

[Broekstra et al., 2002b] Broekstra, J., Kampman, A., et van Harmelen, F. (2002b). Sesame: A generic architecture for storing and querying rdf and rdf schema. ISWC'02.

[Bruijn et al., 2005] Bruijn, J., Lausen, H., Krummenacher, R., Polleres, A., Kifer, M., et Fensel, D. (2005). The web service modeling language wsml. *WSMO project, [On Line] http://www.wsmo.org/TR/d16/d16.1/v0.2/20050320/*.

[Bussler, 2004] Bussler, C. (2004). Wsmo. *The Eleventh International Conference on Artificial Intelligence: Methodology, Systems*.

[Carroll et al., 2004] Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., et Wilkinson, K. (2004). Jena: Implementing the semantic web recommendations. In *Proceedings*

of the 13th international World Wide Web conference on Alternate track papers & posters (WWW'04), pages 74–83, New York, NY, USA. ACM Press.

[Castagna, 1995] Castagna, G. (1995). Covariance and contravariance: Conflict without a cause. In *ACM Transations on Programming Languages and Systems, vol. 17, no. 3, pp. 431-447*.

[Castagna et al., 1995] Castagna, G., Ghelli, G., et Longo, G. (1995). A calculus for overloaded functions with subtyping. In *Information Computation, vol. 117, no. 1, pp. 115-135*.

[Catuneanu, 2002] Catuneanu, O. (2002). Sequence stratigraphy of clastic systems: concepts, merits, and pitfalls,. *Journal of African Earth Sciences, Vol. 35, Issue 1, 2002, p. 1-43*.

[Chong et al., 2005] Chong, E. I., Das, S., Eadon, G., et Srinivasan., J. (2005). An efficient sql-based rdf querying scheme . In *In Proceedings of the 31st international conference on Very Large Data Bases (VLDB'05), pages 1216-1227*.

[CIMIANO et al., 2004] CIMIANO, P., HANDSCHUH, S., et STAAB, S. (2004). Towards the self-annotating web,. In *in Proceedings of the 13th International World Wide Web Conference (WWW'04), ACM Press, New-York, USA, pp. 462-471*.

[Clark, 1998] Clark, D. (1998). Seismic interpretation in the 1990s, the leading edge, vol. 17, no. 6, p. 793-794.

[Corcho, 2006] Corcho, O. (2006). Ontology based document annotation: trends and open research problems. *International Journal of Metadata, Semantics and Ontologies, Inderscience, pp. 47-57*.

[Cullot et al., 2003] Cullot, N., Parent, C., Spaccapietra, S., et Vangenot, C. (2003). Ontologies: A contribution to the dl/db debate. In *Proceedings of the 1st International Workshop on Semantic Web and Databases (SWDB'03)*, pages 109–129.

[Dean et Schreiber, 2004] Dean, M. et Schreiber, G. (2004). *OWL Web Ontology Language Reference*. World Wide Web Consortium.

[Deelman et al., 2005] Deelman, E., Singh, G., hui Su, M., Blythe, J., Gil, A., Kesselman, C., Mehta, G., Vahi, K., Berriman, G. B., Good, J., Laity, A., Jacob, J. C., et Katz, D. S. (2005). Pegasus: a framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming Journal*, 13:219–237.

[Dehainsala, 2007] Dehainsala, H. (2007). *Explicitation de la sémantique dans les bases de données : Le modèle OntoDB de bases de données à base ontologique*. PhD thesis, LISI/ENSMA et Université de Poitiers.

[Dehainsala et al., 2007] Dehainsala, H., Pierra, G., et Bellatreche, L. (2007). Ontodb: An ontology-based database for data intensive applications. In *in Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA'07), ser. Lecture Notes in Computer Science, vol. 4443. Springer, 2007, pp. 497-508*. DASFAA'07.

[Dill et al., 2003] Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J. A., et Zien, J. Y. (2003). Semtag and seeker:

Bootstrapping the semantic web via automated semantic annotation,. In *in Proceedings of the 12th International World Wide Web Conference (WWW'03), ACM Press, Budapest, Hongrie, pp. 178-186.*

[Dingli et al., 2003] Dingli, A., Ciravegna, F., et Wilks, Y. (2003). Automatic semantic annotation using unsupervised information extraction and integration. In *in Proceedings of the Workshop on Knowledge Markup and Semantic Annotation (KCAP'03), Sanibel, Floride*.

[Dorn, 1998] Dorn, G. A. (1998). Modern 3-d seismic interpretation,. In *The Leading Edge, Vol. 17, No. 9, p. 1262-1272*.

[e Wok Hub Consortium, 2008] e Wok Hub Consortium (2008). Semantic hubs for geological projects. ESWC'08 Workshop, SeMMA'08.

[Emmerich et al., 2008] Emmerich, W., Aoyama, M., et Sventek., J. (2008). The impact of research on the development of middleware technology. *ACM Transactions on Software Engineering and Methodology, 17(4):1-48*.

[Fankam et al., 2008a] Fankam, C., Jean, S., et Pierra, G. (2008a). Numeric reasoning in the semantic web. In *SeMMA*, pages 84–103.

[Fankam et al., 2008b] Fankam, C., Jean, S., Pierra, G., et Bellatreche, L. (2008b). Enrichissement de l'architecture ansi/sparc pour expliciter la sémantique des données : une approche fondée sur les ontologies. In *Actes de la 2ème Conférence francophone sur les Architectures Logicielles (CAL'08), Revue RNTI*.

[Farrell et Lausen, 2007] Farrell, J. et Lausen, H. (2007). Sawsdl. Technical report, W3C Recommendation.

[Fensel et Bussler, 2002] Fensel, D. et Bussler, C. (2002). The web service modeling framework wsmf. *In Electronic Commerce Research and Applications, 1(2), Elsevier, Scinces B. V.*

[Fensel et Musen, 2001] Fensel, D. et Musen, M. (2001). Special issue on semantic web technology. *IEEE Intelligent Systems (IEEE IS), 16(2)*.

[Foucault et Raoult, 2005] Foucault, A. et Raoult, J. F. (2005). *Dictionnaire de géologie,*. Masson, 6ème edition, Paris, France.

[Georgakopoulos et al., 1995] Georgakopoulos, D., Hornick, M. F., et Sheth, A. P. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases, 3(2):119-153*.

[Groshong, 2006] Groshong, R. H. J. (2006). 3-d structural geology second edition, springer,.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

[Haarslev et Möller, 2001] Haarslev, V. et Möller, R. (2001). Description of the RACER System and its Applications. In *Working Notes of the 2001 International Description Logics Workshop (DL'01)*.

[Haller et al., 2005] Haller, A., Cimpian, E., Mocan, A., Oren, E., et Bussler, C. (2005). Wsmx - a semantic service-oriented architecture. In *In Proceedings of the International Conference on Web Service (ICWS 2005*, pages 321–328.

[Handschuh, 2005] Handschuh, S. (2005). *Creating Ontology-based Metadata by Annotation for the Semantic Web*. PhD thesis, Université de Karlsruhe.

[Handschuh et al., 2002] Handschuh, S., Staab, S., et Ciravegna, F. (2002). S-cream - semi-automatic creation of metadata,. In *in Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW'02), LNCS 2473, Springer-Verlag, Madrid, Espagne,.*

[Handschuh et al., 2001] Handschuh, S., Staab, S., et Maedche, A. (2001). Cream - creating relational metadata with a component-based, ontology-driven annotation framework,. In *in Proceedings of the Knowledge Capture Conference (KCAP'01), Banff, Canada, pp. 76-83*.

[Harris et Gibbins, 2003a] Harris, S. et Gibbins, N. (2003a). 3store: Efficient bulk rdf storage. In *Proceedings of the 1st International Workshop on Practical and Scalable Semantic Systems (PPP'03)*, pages 1–15.

[Harris et Gibbins, 2003b] Harris, S. et Gibbins, N. (2003b). 3store: Efficient bulk rdf storage. IIn Proc. of the 1st Intern. Workshop on Practical and Scalable Semantic Systems (PSSS'03).

[Horrocks et al., 2003] Horrocks, I., Patel-Schneider, P. F., et van Harmelen, F. (2003). From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics*, 1(1):7–26.

[Hull et al., 2006] Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M. R., Li, P., et Oinn, T. (2006). Taverna: a tool for building and running workflows of services. In *Nucleic Acids Research, 34(Web Server issue):W729-W732, doi:10.1093/nar/gkl320.*

[IEC61360-4, 1999] IEC61360-4 (1999). Standard data element types with associated classification scheme for electric components - Part 4 : IEC reference collection of standard data element types, component classes and terms. Technical report, International Standards Organization.

[IEEE, 1990] IEEE (1990). Ieee standard glossary of software engineering terminology.

[ISO13584-25, 2004] ISO13584-25 (2004). Industrial automation systems and integration – parts library – part 25: Logical resource: Logical model of supplier library with aggregate values and explicit content. Technical report, International Standards Organization, Genève.

[ISO13584-42, 1998] ISO13584-42 (1998). Industrial automation systems and integration – parts library – part 42: Description methodology: Methodology for structuring parts families. Technical report, International Standards Organization, Genève.

[Jean, 2007] Jean, S. (2007). *OntoQL, un langage d'exploitation des bases de données à base ontologique*. PhD thesis, Ecole Nationale Supérieure de Mécanique et d'Aérotechnique, Laboratoire d'Informatique Scientifique et Industrielle, Ecole Doctorale : Science Pour l'Ingénieur.

[Jean et al., 2006a] Jean, S., Aït-Ameur, Y., et Pierra, G. (2006a). Querying ontology based databases. The OntoQL proposal. In *Software Engineering and Knowledge Engineering (SEKE'06)*, pages 166–171.

[Jean et al., 2006b] Jean, S., Pierra, G., et Ait-Ameur, Y. (2006b). Domain ontologies : A database-oriented analysis. In *Proc. of Web Information Systems and Technologies (WEBIST'2006), Setubal, Portugal, 11-13 April*.

[Jean et al., 2007] Jean, S., Pierra, G., et Ait-Ameur, Y. (2007). An object-oriented based algebra for ontologies and their instances. In *in Proceedings of the 11th East European Conference in Advances in Databases and Information Systems (ADBIS'07), ser. LNCS, Y. E. Ioannidis, B. Novikov, and B. Rachev, Eds., vol. 4690. Springer, 2007, pp. 141-156*.

[Jordan et Evdemon, 2007] Jordan, D. et Evdemon, J. (2007). Ws-bpel 2.0. Technical report, OASIS Standard.

[Kahan et al., 2001] Kahan, J., Koivunen, M. R., Prud'Hommeau, E., et Swick, R. (2001). Annotea: An open rdf infrastructure for shared web annotations,. In *in Proceedings of the 10th International World Wide Web Conference (WWW'01), ACM Press, Hong-Kong, pp. 623-632*.

[Kavantzas et al., 2005] Kavantzas, N., Burdett, D., Ritzinger, G., Fletcher, T., et Lafon, Y. (2005). Web services choreography description language version 1.0. Technical report, W3C Candidate Recommendation.

[Kim et al., 2004] Kim, J., Spraragen, M., et Gil, Y. (2004). An intelligent assistant for interactive workflow composition. In *In Proceedings of the International Conference on Intelligent User Interfaces (IUI2004), Madeira, Portugal*.

[Kritikos et Plexousakis, 2007] Kritikos, K. et Plexousakis, D. (2007). Owl-q for semantic qos-based web service description and discovery. In *SMRR*.

[Lara et al., 2004] Lara, R., Roman, D., Polleres, A., et Fensel, D. (2004). A conceptual comparison of wsmo and owl-s. *In Proceedings of the European Conference on Web Services (ECOWS 2004)*.

[Lee et Neuendorffer, 2000] Lee, E. A. et Neuendorffer, S. (2000). Moml - a modeling markup language in xml - version 0.4. In *Technical report, University of California at Berkeley*.

[Leymann, 2001] Leymann, F. (2001). Web service flow language (wsfl 1.0). *IBM, Specification available at http://www.ibm.com/software/solutions/webservices/pdf/WSFL.pdf*.

[Loridan et Rosell, 2006] Loridan, C. et Rosell, J. A. (2006). *BONITA - Workflow patterns support*. BULL R&D.

[Ludäscher et al., 2006] Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E. A., Tao, J., et Zhao, Y. (2006). Scientific workflow management and the kepler system. *Concurrency and Computation: Practice and Experience, vol. 18, n° 10, p. 1039-1065*.

[Ma et al., 2004] Ma, L., Su, Z., Pan, Y., Zhang, L., et Liu, T. (2004). Rstar: an rdf storage and query system for enterprise resource management. thirteenth ACM international conference on Information and knowledge management: 484 - 491.

[Maamar et al., 2009] Maamar, Z., sheng, Q. Z., Tata, S., Benslimane, D., et Sellami, M. (2009). Towards An Approach to Sustain Web Services High-Availability Using Communities of Web Services. *International Journal of Web Information Systems*, 5(1):32–55.

[Manola et Miller, 2004] Manola, F. et Miller, E. (2004). *RDF Primer*. World Wide Web Consortium. `http://www.w3.org/TR/rdf-primer`.

[Marshall, 1998] Marshall, C. (1998). Toward an ecology of hypertext annotation. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia. ACM Press New York, NY, USA. pp. 40-49*.

[Martin et al., 2004] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., et Sycara, K. (2004). Owl-s: Semantic markup for web services. Technical Report.

[Mastella, 2010] Mastella, L. (2010). *Semantic exploitation of engineering models: application to petroleum reservoir models*. PhD thesis, l'École nationale supérieure des mines de Paris.

[McBride, 2001] McBride, B. (2001). Jena: Implementing the rdf model and syntax specification. *Proceedings of the 2nd International Workshop on the Semantic Web*.

[Michum et al., 1977] Michum, R. M., Vail, P. R., et Sangree, J. B. (1977). Seismic stratigraphy and global changes of sea level, part 6 : Stratigraphic interpretation of seismic reflection patterns in depositional sequences,. *AAPG Memoir 26*.

[Mille, 2005] Mille, D. (2005). *Modèles et outils logiciels pour l'annotation sémantique de documents pédagogiques*. PhD thesis, Département informatique. Grenoble, Université Joseph-Fourier.

[Miller et al., 2004] Miller, J., Verma, K., Rajasekaran, P., Sheth, A., Aggarwal, R., et Sivashanmugam, K. (2004). Wsdl-s. University of Georgia.

[Mrissa, 2007] Mrissa, M. (2007). *Médiation Sémantique Orientée Contexte pour la Composition de Services Web*. PhD thesis, Université Claude Bernard Lyon I - (spécialité informatique).

[OASIS, 2004] OASIS (2004). Universal description, discovery, and integration (uddi). Technical report.

[OMG, 2006] OMG (2006). Meta object facility (mof) core specification, version 2.0. Technical report, OMG Available Specification.

[Orchestra-Team, 2010] Orchestra-Team (2010). *Orchestra User Guide*.

[Osterweil, 1987] Osterweil, L. (1987). Software processes are software too. *In ICSE'87: Proceedings of the 9th international conference on Software Engineering, pages 2-13, Los Alamitos, CA, USA, 1987. IEEE Computer Society Press*.

[Pan et Heflin, 2003] Pan, Z. et Heflin, J. (2003). Dldb: Extending relational databases to support semantic web queries. In *Proceedings of the 1st International Workshop on Practical and Scalable Semantic Systems (PSSS'03)*, pages 109–113.

[Park et al., 2007] Park, M. J., Lee, J. H., Lee, C. H., Lin, J., Serres, O., et Chung, C. W. (2007). An efficient and scalable management of ontology. DASFAA'07.

[Peer, 2005] Peer, J. (2005). Semantic service and markup sesma. In *In Web Service Semantics Workshop (WSS'05) at the fourteenth International World Wide Web Conference (WWW'05)*.

[Peltz, 2003] Peltz, C. (2003). Web services orchestration and choreography. computer journal, vol. 36(no. 10):46-52. In *IEEE Computer Society Press.*

[Perrin, 1998] Perrin, M. (1998). Geological consistency: an opportunity for safe surface assembling and quick model exploration. *In GOCAD ENSG Conference - 3D Modeling of Natural Objects: A Challenge for the 2000's, volume 3, pages 4-5, Nancy, France*.

[Perrin et al., 2008] Perrin, M., P.Durville, Grataloup, S., L.Mastella, Lions, S., O.Morel, et Rainaud, J.-F. (2008). Knowledge issues for automatic identification of co2 storage sites by means of semantic web technology. *In EAGE CO2 Geological Storage Workshop Workshop on CO2 Sequestration, Budapest Hungary, 28-29 September 2008. EarthDoc.*

[Petrini et Sward, 2007] Petrini, J. et Sward, T. R. (2007). Semantic web abridged relational databases. In *In Proceedings of the 18th International Conference on Database and Expert Systems Applications (DEXA'07), pages 455-459,*.

[Pierra, 2003a] Pierra, G. (2003a). Context-explication in conceptual ontologies: The plib approach. In *Proc. of 10th ISPE International Conf. on Concurrent Engineering: Research and Applications (CE'03), pp 243-254,*.

[Pierra, 2003b] Pierra, G. (2003b). Context-explication in conceptual ontologies: The plib approach. In Jardim-Gonçalves, R., Cha, J., et Steiger-Garçao, A., editors, *Proceedings of the 10th ISPE International Conference on Concurrent Engineering (CE'03)*, pages 243–254.

[Pierra, 2007] Pierra, G. (2007). Context Representation in Domain Ontologies and its Use for Semantic Integration of Data. *Journal Of Data Semantics (JODS)*, X:34–43.

[Pinheiro da Silva et al., 2007] Pinheiro da Silva, P., Salayandia, L., et Gates, A. Q. (2007). Wdo-it! a tool for building scientific workflows form ontologies. *University of Texas at El Paso*.

[Popov et al., 2003] Popov, B., Kiryakov, A., Manov, D., Kirilov, A., Ognyanoff, D., et Goranov, M. (2003). Towards semantic web information extraction,. In *in Proceedings of the Human Language Technologies Workshop (ISWC'03), Sanibel, Floride, pp. 1-22*.

[Psyché et al., 2003] Psyché, V., Mendes, O., et Bourdeau, J. (2003). Apport de l'ingénierie ontologique aux environnements de formation à distance. *Revue Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation (STICEF)*, 10.

[Qin et Fahringer, 2008] Qin, J. et Fahringer, T. (2008). A novel domain oriented approach for scientific grid workflow composition. *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (Supercomputing 2008)*.

[Rojas, 2006] Rojas, H.-D. (2006). *Orchestration à Haut Niveau et BPEL*. PhD thesis, Université Joseph Fourier de Grenoble.

[Roman et al., 2005] Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., et Fensel, D. (2005). Wsmo. *IOS Press*.

[Scheidegger et al., 2008] Scheidegger, C. E., Koop, D., Freire, J., et Silva, C. T. (2008). Querying and re-using workflows with vistrail. *In Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 1251-1254*.

[Schenk et Wilson, 1994] Schenk, D. et Wilson, P. (1994). *Information Modelling The EXPRESS Way*. Oxford University Press.

[Schlumberger Oilfield Glossary, 2009] Schlumberger Oilfield Glossary (2009). http://www.glossary.oilfield.slb.com/default.cfm.

[Spyns et al., 2002] Spyns, P., Meersman, R., et Jarrar, M. (2002). Data modelling versus ontology engineering. *Volume 31, New York, NY, USA, pp. 12-17. ACM Press*.

[Stoffel et al., 1997] Stoffel, K., Taylor, M., et Hendler, J. (1997). Efficient management of very large ontologies. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference AAAI'97/IAAI'97*, pages 442–447.

[Taylor et al., 2005] Taylor, I., Shields, M., Wang, I., et Harrison, A. (2005). Visual grid workflow in triana. *Journal of Grid Computing*, 3(3-4):153–169.

[Thatte, 2001] Thatte, S. (2001). Xlang: web services for business process design. *Microsoft, Specification*.

[TIBCO, 2003] TIBCO (2003). *Managing Business Processes with TIBCO BusinessWorks Workflow*.

[Tondello et Siqueira, 2008] Tondello, G. F. et Siqueira, F. (2008). The qos-mo ontology for semantic qos modeling. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 2336–2340, New York, NY, USA. ACM.

[Vergas-Vera et al., 2002] Vergas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., et Ciravegna, F. (2002). Mnm: Ontology driven semi-automatic and automatic support for semantic markup,. In *in Proceedings of the 13th International Conference on Knowledge Engineering and Management (EKAW'02), LNCS 2473, Springer-Verlag, Madrid, Espagne, pp. 379-391*.

[Verney, 2009] Verney, P. (2009). *Interprétation géologique de données sismiques par une méthode supervisée basée sur la vision cognitive*. PhD thesis, École Nationale Supérieure des Mines de Paris.

[W3C, 1999] W3C (1999). Xml path language (xpath) version 1.0. Technical report.

[W3C, 2000] W3C (2000). Simple object access protocol (soap). Technical report.

[W3C, 2001] W3C (2001). Xml schema. Technical report.

[W3C, 2004] W3C (2004). Web service definition language (wsdl 1.1). Technical report.

[Wesk, 2007] Wesk, M. (2007). Business process management: Concepts, languages, architectures. *Springer Verlag*.

[White, 2004] White, S. A. (2004). Introduction to bpmn. *IBM Corporation*.

[Wilkinson et al., 2003] Wilkinson, K., Sayers, C., Kuno, H., et Reynolds, D. (2003). Efficient rdf storage and retrieval in jena2. *HP Laboratories Technical Report HPL-2003-266*, pages 131–150.

[Workflow Management Coalition, 1994] Workflow Management Coalition (1994). Workflow reference model (wfmc-tc-1003). technical report, workflow management coalition.

[Workflow Management Coalition, 2002] Workflow Management Coalition (2002). Workflow process definition interface - xml process definition language (xpdl) (wfmc-tc-1025). technical report, workflow management coalition.

[Workflow Management Coalition, 2008] Workflow Management Coalition (2008). Xpdl (xml process definition language). Workflow Management Coalition Workflow Standard.

[Zhou et al., 2004] Zhou, C., Chia, L.-T., et Lee, B.-S. (2004). Daml-qos ontology for web services. In *ICWS '04: Proceedings of the IEEE International Conference on Web Services*, page 472, Washington, DC, USA. IEEE Computer Society.

Table des figures

1.1 OCC locales sur le domaine des vins. 13
1.2 OCNC intégrée sur le domaine des vins. 14
1.3 Le modèle en oignon. 15
1.4 Modèle d'ontologie noyau. 17
1.5 Annotation linguistique d'un document dans le projet *e-Wok Hub*. 19
1.6 Architecture générale d'une BDBO. 20
1.7 Un exemple d'ontologie et des instances de ses concepts. 21
1.8 Schéma de BDBO de type 1. 22
1.9 Schéma de BDBO de type 2. 23
1.10 Schéma de BDBO de type 3. 23
1.11 Architecture d'OntoDB. 24
1.12 Extrait de l'interface JOBDBC. 29
1.13 Extrait de l'interface JOBDBC. 30

2.1 Scénario d'exécution d'un service Web. 36
2.2 Exemple de document WSDL pour le service *Purchase Order*. 38
2.3 Requête SOAP. 39
2.4 Réponse SOAP. 39
2.5 Entrée de l'annuaire UDDI. 40
2.6 Une chorégraphie de services Web. 41
2.7 Une orchestration de services Web. 42
2.8 Généalogie des normes en description de workflows [Emmerich et al., 2008]. . 43

2.9 La description graphique du processus *Purchase Order*. 45
2.10 Exemple de workflow BPEL. 46
2.11 La description graphique du processus *Approve_Purchase*. 47
2.12 Exemple d'un workflow XPDL. 48
2.13 Services Web sémantiques. 49
2.14 Modèle d'ontologie de services dans OWL-S. 51
2.15 Exemple de description OWL-S pour le workflow *BravoAir_Process*. 51
2.16 Exemple de description OWL-S pour le service atomique *LogIn*. 52
2.17 Architecture de WSMO. 53
2.18 Opération devis de l'assuré en WSDL. 54
2.19 Description de l'opération devis de l'assuré en SAWSDL. 54
2.20 SESMA, WSDL et BPEL. 55
2.21 Description SESMA pour l'annotation sémantique d'une opération WSDL. . . 55

3.1 Exemples de successions d'horizons. 64
3.2 Exemple réel d'un chenal. 65
3.3 Une faille commune. 66
3.4 Vue d'ensemble de la modélisation géologique d'un gisement (le processus *earth modeling*). 67
3.5 Réflexion sismique. 69
3.6 Exemple d'une image sismique. 69
3.7 Un processus d'interprétation sismique représenté par des géologues. 72
3.8 Représentation graphique d'un workflow informatique d'interprétation sismique. 73
3.9 Vue de l'éditeur de workflow de OpenFlow de *IFP Énergies Nouvelles.* 74

4.1 Approche à base ontologique pour la gestion de workflows. 86
4.2 Un méta-modèle de données. 87
4.3 Une ontologie de domaine sur la documentation. 89
4.4 Modèle d'ontologie de domaine. 90
4.5 Méta-modèle d'annotation et la liaison entre un méta-modèle de données et un modèle d'ontologie de domaine. 91
4.6 Indexation sémantique d'un modèle de facture électronique par un concept ontologique. 92

4.7 Méta-modèle de processus défini en XPDL. 94
4.8 Une ontologie de services sur l'achat des pièces. 95
4.9 Extension du modèle d'ontologie pour la prise en compte de services sémantiques. 96
4.10 Indexation sémantique d'un service Web par le service sémantique *Electronic Invoicing Service*. 98
4.11 Modèle complet de l'approche à base ontologique pour la gestion sémantique de workflows. 99
4.12 Modèle de description de workflows sémantiques. 101
4.13 Un workflow sémantique sur un achat de pièces et une double facturation. . . . 102
4.14 Description du workflow sémantique dans le modèle défini. 102

5.1 Requête OntoDB pour la création de concepts ontologiques. 110
5.2 Exemple de création d'un concept d'une ontologie de domaine. 111
5.3 Exemple d'indexation d'un modèle de données par un concept ontologique. . . 112
5.4 Requête OntoDB pour la création de services sémantiques. 113
5.5 Exemple de création de service sémantique. 115
5.6 Exemple d'indexation d'un service informatique par un service sémantique. . . 116
5.7 Extension d'OntoDB pour la représentation d'ontologies de services et entreposage. 117
5.8 Exemple de représentation d'attributs multivalué dans OntoDB. 124
5.9 Fragment de code JAVA pour l'extension du développement d'OntoQL. 125
5.10 Exemple de covariance. 126
5.11 Fragment de code PL/SQL pour la création d'un déclencheur pour la vérification de la covariance. 127
5.12 Exemple de contraintes entre le niveau ontologique et le niveau instance. 128

6.1 Ontologie de domaine, données et indexation. 136
6.2 Fragment de l'ontologie du domaine d'interprétation sismique. 139
6.3 Indexation d'un modèle de données par un concept de l'ontologie de la sismique. 140
6.4 Ontologie de services, services, workflows informatiques et indexation. 140
6.5 Fragment de l'ontologies des services pour l'interprétation sismique. 143
6.6 Exemple de référencement de concepts d'ontologies de domaine par des services sémantiques. 144

6.7 Indexation d'un service informatique de modélisation géologique par un service sémantique. 145
6.8 Workflow sémantique potentiel simplifié d'une interprétation sismique. 146
6.9 Description d'un workflow sémantique en tant qu'instance de son modèle et sous forme infixée. 147
6.10 Exemple d'interopérabilité au niveau sémantique. 147
6.11 Fragment de l'ontologie des services pour l'interprétation sismique enrichie du service sémantique *StructuralModelBuilding*. 148
6.12 Workflow informatique d'interprétation sismique dérivé de la sélection de services et de workflows informatiques pour chaque service sémantique. 149
6.13 Autre workflow informatique d'interprétation sismique dérivé de la sélection de services et de workflows informatiques pour chaque service sémantique. 150

7.1 Création du concept *Reflector* dans l'ontologie de la sismique. 156
7.2 Instanciation du concept ontologique *Horizon*. 157
7.3 Création du service sémantique *ReflectorExtraction*. 159
7.4 Instantiation d'un service Web par le service sémantique *ReflectorExtraction*. . 160

8.1 Connexion du GWE à OntoDB. 170
8.2 Réutilisation de services et de workflows informatiques. 171
8.3 Vue du GWE pour le scénario de réutilisation de services et de workflows. . . . 173
8.4 Vue des ontologies de domaine sur le GWE. 174
8.5 Vue des ontologies de services sur le GWE. 175
8.6 Vue de l'interface de création de services sémantiques. 176
8.7 Vue de l'interface d'indexation de services et de workflows informatiques. . . . 177
8.8 Création de nouveaux workflows informatiques. 180
8.9 Vue du GWE pour le scénario de création de nouveaux workflows informatiques. 181
8.10 Vue de l'interface de création de workflows sémantiques. 182
8.11 Visualisation du workflow sémantique en XML. 183
8.12 Workflow sémantique d'interprétation sismique. 184
8.13 Enrichissement de l'ontologie des services. 185
8.14 Association de services informatiques aux services sémantiques dans un workflow sémantique. 186

Liste des tableaux

6.1 Modèle en *XYZFile (.xyz)*. 137
6.2 Fragment de caractérisation de modèles de données en modélisation géologique. 137
6.3 Quelques services et workflows informatiques de modélisation géologique. . . 142

Glossaire

ANR : Agence Nationale de la Recherche
API : Application Programming Interface

BD : Base de données
BDBO : Base de données à base ontologique
BPEL : Business Process Execution Language
BPMN : Business Process Modeling Notation
BRGM : Bureau de Recherches Géologiques et Minières

CAO : Conception Assistée par Ordinateur
CAT : Composition Analysis Tool
CRITT : Centre Régional d'Innovation et de Transfert de Technologie

EADS : European Aeronautic Defense and Space
ENSMA : École National Supérieure de Mécanique et d'Aérotechnique

IBM : International Business Machines
IFP : Institut Français du Pétrole - Énergies Nouvelles
ISO : International Organization for Standardization
INRIA : International Organization for Standardization

HTTP : Hypertext Transfer Protocol

LISI : Laboratoire d'Informatique Scientifique et Industrielle

MC : Modèle Conceptuel
MoML : Model Markup Language

OASIS : Organization for the Advancement of Structured Information Standards

OC : Ontologie Conceptuelle
OCC : Ontologie Conceptuelle Canonique
OCNC : Ontologie Conceptuelle Non Canonique
OL : Ontologie Linguistique
OWL : Web Ontology Language
OWL-S : Web Ontology Language for Web Services

PLIB : Parts Library - Norme ISO 13584
RDF : Ressource Description Framework

SAWSDL : Semantic Annotation for WSDL
SESMA : SEmantic Service MArkup
SGBD : Système de Gestion de Bases de Données
SMTP : Simple Mail Transfer Protocol

UDDI : Universal Description, Discovery and Integration
UML : Unified Modeling Language
URI : Uniform Resource Identifier
URL : Uniform Resource Locator

W3C : WorldWideWeb Consortium
WDOs : Workflow-Driven Ontologies
WfMC : Workflow Management Coalition
WfMS : Workflow Management System
WS-CDL : Web Services Choreography Description Language
WSDL : Web Service Description Language
WSFL : Web Services Flow Language
WSMO : Web Service Modeling Ontology

XLANG : XML Business Process Language
XML : Extensible Markup Language
XMLS : XML Schema
XPath : XML Path Language
XPDL : XML Process Definition Language

Résumé

Les services informatiques, et particulièrement les services sur le Web, permettent le traitement ou l'échange d'informations. Leur composition ainsi que la construction de workflows à partir de services permettent la réalisation de tâches complexes. Toutefois, seules des informations utiles à leur gestion informatique (stockage, exécution, etc.) sont spécifiées dans les langages de description de services ou de compositions de services tels que WSDL, BPEL ou XPDL. Ces descriptions, essentiellement syntaxiques, ne sont pas suffisamment riches pour permettre une gestion complexe supportant la découverte, l'interopérabilité ou l'adaptativité de services au niveau sémantique. En effet, ces descriptions ne permettent pas de lier explicitement les services informatiques et les compositions qu'elles définissent aux fonctions et aux processus implémentés ainsi qu'aux concepts manipulés. Dans les domaines de l'ingénierie par exemple, l'activité de composition de services informatiques reste traditionnellement à la charge des ingénieurs. L'absence de sémantique explicite rend les services et workflows informatiques difficilement compréhensibles, réutilisables et composables pour les ingénieurs du domaine d'intérêt qui n'ont pas contribué à leur conception et ayant des connaissances techniques et informatiques en général limitées. Ces ingénieurs restent ainsi dépendants des développeurs de logiciels pour l'utilisation ou la composition de services informatiques.

Pour apporter une solution aux problèmes évoqués, de récentes approches ont été proposées. Cependant, leurs mécanismes d'enrichissement sémantique ne séparent pas la partie syntaxique, utile au stockage ou à l'exécution, de la partie sémantique, utile à la découverte ou à la composition. En outre, ces mécanismes ne fournissent aucune classification hiérarchique ou relationnelle selon la sémantique des services informatiques. Pour remédier à ces limitations, nous proposons une approche basée sur la définition d'*ontologies de services* (conceptualisations partagées) et d'indexations sémantiques. Notre proposition s'appuie sur une architecture de persistance centrée autour des bases de données à base ontologique pour stocker et indexer les différents services ainsi que leurs compositions. La mise en œuvre de notre approche consiste en un prototype logiciel implantant les concepts de service et de workflow sémantiques permettant de stocker, de rechercher, de remplacer, de réutiliser les services et les workflows informatiques existants et d'en construire de nouveaux de manière incrémentale. Ces travaux sont validés en étant appliqués au domaine de la modélisation géologique et de l'ingénierie pétrolière et plus particulièrement à l'interprétation sismique.

Mots-clés : Ontologie, Méta-modélisation, Modélisation et base de données à base ontologique, Indexation sémantique, Service sémantique, Workflow sémantique, Modélisation géologique, Interprétation sismique.

Modélisation de services et de workflows sémantiques à base d'ontologies de services et d'indexations. Application à la modélisation géologique.

Présentée par :

Nabil BELAID

Directeurs de Thèse :

Yamine AIT-AMEUR et **Jean-François RAINAUD**

Résumé. Les services informatiques, et particulièrement les services sur le Web, permettent le traitement ou l'échange d'informations. Leur composition ainsi que la construction de workflows à partir de services permettent la réalisation de tâches complexes. Toutefois, seules des informations utiles à leur gestion informatique (stockage, exécution, etc.) sont spécifiées dans les langages de description de services ou de compositions de services tels que WSDL, BPEL ou XPDL. Ces descriptions, essentiellement syntaxiques, ne sont pas suffisamment riches pour permettre une gestion complexe supportant la découverte, l'interopérabilité ou l'adaptativité de services au niveau sémantique. En effet, ces descriptions ne permettent pas de lier explicitement les services informatiques et les compositions qu'elles définissent aux fonctions et aux processus implémentés ainsi qu'aux concepts manipulés. Dans les domaines de l'ingénierie par exemple, l'activité de composition de services informatiques reste traditionnellement à la charge des ingénieurs. L'absence de sémantique explicite rend les services et workflows informatiques difficilement compréhensibles, réutilisables et composables pour les ingénieurs du domaine d'intérêt qui n'ont pas contribué à leur conception et ayant des connaissances techniques et informatiques en général limitées. Ces ingénieurs restent ainsi dépendants des développeurs de logiciels pour l'utilisation ou la composition de services informatiques.
Pour apporter une solution aux problèmes évoqués, de récentes approches ont été proposées. Cependant, leurs mécanismes d'enrichissement sémantique ne séparent pas la partie syntaxique, utile au stockage ou à l'exécution, de la partie sémantique, utile à la découverte ou à la composition. En outre, ces mécanismes ne fournissent aucune classification hiérarchique ou relationnelle selon la sémantique des services informatiques. Pour remédier à ces limitations, nous proposons une approche basée sur la définition d'*ontologies de services* (conceptualisations partagées) et d'indexations sémantiques. Notre proposition s'appuie sur une architecture de persistance centrée autour des bases de données à base ontologique pour stocker et indexer les différents services ainsi que leurs compositions. La mise en œuvre de notre approche consiste en un prototype logiciel implantant les concepts de service et de workflow sémantiques permettant de stocker, de rechercher, de remplacer, de réutiliser les services et les workflows informatiques existants et d'en construire de nouveaux de manière incrémentale. Ces travaux sont validés en étant appliqués au domaine de la modélisation géologique et de l'ingénierie pétrolière et plus particulièrement à l'interprétation sismique.

Mots-clés : Ontologie, Méta-modélisation, Modélisation et base de données à base ontologique, Indexation sémantique, Service sémantique, Workflow sémantique, Modélisation géologique, Interprétation sismique.